Countering Backdoor Attacks in Image Recognition: A Survey and Evaluation of Mitigation Strategies

Anonymous authors
Paper under double-blind review

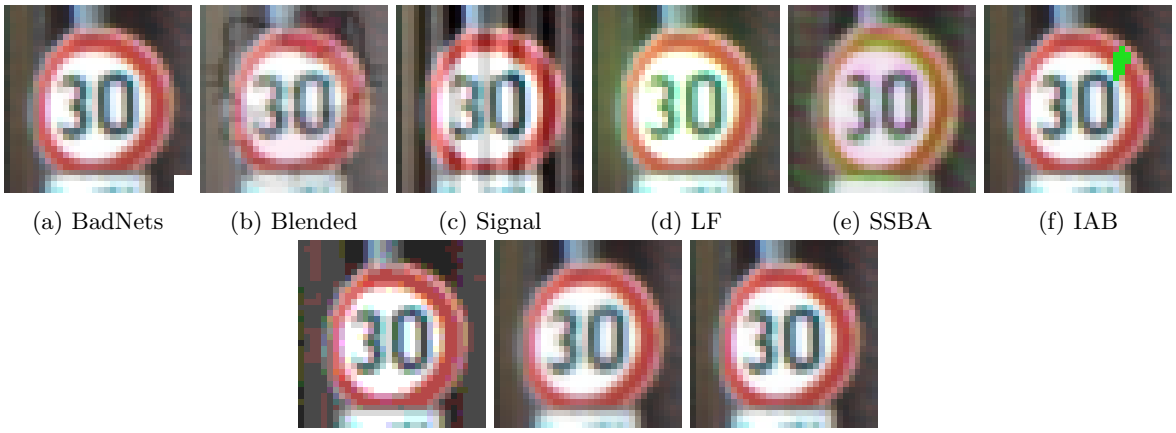
Abstract

The widespread adoption of deep learning across various industries has introduced substantial challenges, particularly in terms of model explainability and security. The inherent complexity of deep learning models, while contributing to their effectiveness, also renders them susceptible to adversarial attacks. Among these, backdoor attacks are especially concerning, as they involve surreptitiously embedding specific triggers within training data, causing the model to exhibit aberrant behavior when presented with input containing the triggers. Such attacks often exploit vulnerabilities in outsourced processes, compromising model integrity without affecting performance on clean (trigger-free) input data. In this paper, we present a review of prominent existing mitigation strategies designed to counter backdoor attacks in image recognition. We provide an in-depth analysis of the theoretical foundations, practical efficacy, and limitations of these approaches. In addition, we conduct an extensive benchmarking of sixteen prominent approaches against eight distinct backdoor attacks, utilizing three datasets, four model architectures, and three poisoning ratios. Our results, derived from 122,236 individual experiments, indicate that while many approaches provide some level of protection, their performance can vary considerably. Furthermore, when compared to two seminal approaches, most newer approaches do not demonstrate substantial improvements in overall performance or consistency across diverse settings. Drawing from these findings, we propose potential directions for developing more effective and generalizable defensive mechanisms in the future.

1 Introduction

In recent years, deep learning has seen remarkable advancements, driving its widespread adoption across diverse industries and academic fields. This rapid integration is evident in sectors such as healthcare, education, automotive, and logistics, where deep learning is increasingly utilized to foster innovation Pouyanfar et al. (2018). A key factor in the success of deep learning is its ability to extract complex patterns from data. While this capability offers significant advantages, it also introduces substantial challenges related to explainability. Despite their strong predictive performance, deep learning models often lack transparency and interpretability, making it difficult to provide clear reasoning for specific predictions.

The black-box nature of deep learning models has been shown to expose them to considerable security vulnerabilities Wang et al. (2019c). In particular, classification models, such as those used in image recognition, have been demonstrated to be vulnerable to manipulation, with multiple instances of adversarial attacks successfully compromising their decision-making processes. For instance, the seminal work of Szegedy et al. (2013) highlights the vulnerability of image classification models to adversarial examples, showing that imperceptible perturbations applied to input images can induce significant misclassifications. The iconic "Panda-Gibbon" image, created using the method proposed in Szegedy et al. (2013) for generating adversarial perturbations, is a striking illustration of the fragility inherent in deep learning models, despite their sophistication. Since then, several other adversarial threats have been identified, affecting a wide range of learning tasks Wang et al. (2019c).



(a) BadNets (b) Blended (c) Signal (d) LF (e) SSBA (f) IAB

(g) BPP (h) WaNet (i) LIRA

Figure 1: Examples of different backdoor triggers used in the literature. Note that while IAB adds a local patch to each image, its position and scale can vary across images.

In real-world scenarios, backdoor attacks represent a significant threat, especially in scenarios where classification outputs drive automated decision-making processes Gu et al. (2017). Backdoor attacks intentionally establish a relationship between a spurious feature within the input space, known as a trigger, and a specific classification outcome. Once this association is established, a compromised model will classify clean images (i.e., unaltered images) and backdoor images (i.e., images containing the trigger) differently. As a result, backdoor attacks undermine the integrity of a model's decision-making by inserting an unwanted behaviour, referred to as a backdoor task, without compromising its ability to correctly perform the original classification task of recognizing clean images Li et al. (2022).

To successfully execute a backdoor attack, an adversary must compromise the victim's training pipeline. As a result, backdoor attacks are particularly dangerous when model training is outsourced to third parties, such as through cloud-based Machine-Learning-as-a-Service (MLaaS) platforms Liu et al. (2019), or when third parties are used to collect training data. The use of third-parties in both cases allows adversaries to manipulate the training data and or training procedure without the victim's knowledge. The widespread use of pre-trained model weights in the deep learning community highlights a realistic vulnerability that adversaries could exploit to facilitate successful backdoor attacks. The recent industry survey Grosse et al. (2023) revealed that 48.1% of participants utilise third-party weights for model training. Moreover, popular machine learning platforms such as Hugging Face allow users to download thousands of off-the-shelf pre-trained models, as well as offer MLaaS, further amplifying the potential risks.

To address the challenges posed by the hard-to-explain nature of current deep learning approaches, machine learning security has emerged as a critical area of research Yuan et al. (2019). Although it does not directly solve the problem of explainability, machine learning security seeks to develop new methods to protect against known threats, ensuring that deep learning can be safely deployed. Within this field, an emerging body of work focuses on developing defensive strategies specifically designed to counter backdoor attacks. These strategies aim to mitigate risks by removing the backdoor task from a model while preserving its ability to classify clean inputs. However, despite significant advancements in this field, the consistency and reliability of current proposals across diverse settings remain uncertain. Many proposed approaches are evaluated using a limited scope of attacks, datasets, model architectures, and data availability conditions. This raises questions about their generalizability and effectiveness in diverse real-world scenarios.

In this work, we critically analyse existing research on backdoor mitigation, distinguishing it from other defensive approaches such as detecting the presence of a backdoor, identifying poisoned training examples and or synthesizing backdoored inputs. Our analysis is centred on works designed for image classification, although backdoor attacks are also relevant in other applications such as natural language processing Sheng et al. (2022) and other computer vision tasks such as object detection Chan et al. (2022) and semantic

Table 1: Comparison of our work with existing related surveys. Note: *# Evaluated* refers to the number of mitigation proposals benchmarked.

Reference	Year	Domain	Defensive Tasks	Experimental Evaluation	# Evaluated
Sheng et al. (2022)	2022	Text	Various	✗	N/A
Cheng et al. (2023)	2023	Text	Various	✗	N/A
Zhao et al. (2024)	2024	Text	Various	✗	N/A
Yan et al. (2023)	2023	Voice	Various	✗	N/A
Wan et al. (2024)	2024	Federated Learning	Various	✗	N/A
Le Roux et al. (2024)	2024	Image	Various	✗	N/A
Li et al. (2022)	2022	Image	Various	✗	N/A
Wu et al. (2022)	2022	Image	Various	✓	4
Ours	2024	Image	Mitigation	✓	16

segmentation Li et al. (2021b). We conduct a comprehensive survey and critical analysis of existing mitigation approaches. Unlike other surveys on this topic, we offer a detailed summary of the prevalent approaches used to address backdoor attacks, along with their main assumptions and limitations. Furthermore, we experimentally evaluate the majority of the discussed works against a diverse range of backdoor attacks, covering eight distinct types. Our evaluations comprising X individual experiments, spanning three different datasets, four model architectures, and three distinct data availability settings. Moreover, we more closely evaluate the performance of the best-performing methods to better understand why they fail in particular cases. Our benchmarking results reveal several key findings that offer valuable insights to inform future research directions.

Existing Surveys and Benchmarks: There exist several surveys on backdoor attack mitigation. We provide a comparative summary of these works in Table 1. Most of these surveys take a broader approach than ours, which often results in a less detailed analysis of the methods currently employed within the image classification domain. For instance, Sheng et al. (2022), Cheng et al. (2023), and Zhao et al. (2024) concentrate on language tasks, while Le Roux et al. (2024) and Yan et al. (2023) focus on face and voice recognition tasks, respectively. Additionally, Wan et al. (2024) specifically surveys mitigation strategies devised for federated learning.

The two works most similar to ours are Li et al. (2022) and Wu et al. (2022). In Li et al. (2022), multiple classification tasks are surveyed, but the analysis of methods specific to image classification is not detailed. While Wu et al. (2022) focuses on image classification, its primary aim is the development of a benchmarking tool. Although Wu et al. (2022) offers an extensive evaluation of various defensive approaches, only five of the nine evaluated methods perform mitigation, with the others employing different defensive strategies (c.f., Table 2 in their original paper). Moreover, the five mitigation approaches assessed by Wu et al. (2022) cannot currently be considered state-of-the-art, particularly in light of more recent results, for example those presented inZhu et al. (2024b) and Wei et al. (2023). While Wu et al. (2022) has included several additional methods since publication, large-scale benchmarking of these works within a consistent setup, to the best of our knowledge, has not been conducted.

2 Background and Preliminaries

In this section, we introduce key foundational and theoretical concepts relevant to our survey and establish a consistent set of notations that will be utilised throughout the paper.

2.1 Notation

Here, we provide a concise overview of the general notation that is used in the subsequent sections. In Table 2, we list the common symbols that are referenced throughout the paper.

Table 2: The list of common symbols.

Symbol	Meaning
θ	Model Parameters
φ	Parameters in θ associated with feature extractor
ω	Parameters in θ associated with the linear classifier
ϕ	Filter matrix of a convolutional layer
ξ	Parameter perturbation applied to θ
δ	Input perturbation applied to x
ϵ	Perturbation budget
λ	Loss Hyperparameter
p	Norm type used to define $\|\cdot\|_p$
$\mathcal{X} \subseteq \mathbb{R}^{h \times w \times c}$	Input space with c channels, w width and h height
$x \in \mathcal{X}$	Clean input image
$\hat{x} \in \mathcal{X}$	Backdoor image
ρ	Trigger Pattern applied to x to produce $\hat{x}$
$\mathcal{M} = \{0, 1, \ldots, m\}$	Label space
$y \in \{0,1\}^m$	Correct label associated with x (One-hot encoded)
$\hat{y} \in \{0,1\}^m$	Target label associated with $\hat{x}$ (One-hot encoded)
$(x, y) \in \mathcal{D}_t$	Training Dataset
$(x, y) \in \mathcal{D}_c$	Clean Dataset
$(\hat{x}, \hat{y}) \in \mathcal{D}_b$	Backdoor Dataset
$(x, y) \in \mathcal{D}_m$	Mitigation Dataset
$(x, y) \in \mathcal{D}_v$	Validation Dataset
$z \in \mathbb{Q}^m$	Logit output of a model
$a \in [0,1]^m$	Softmax output of a model

2.1.1 Cross-Entropy Loss

Let $a = f(x, \theta) \in \mathbb{R}^m$ be the softmax output of a network for input x and parameters θ, and let $y \in \{0,1\}^m$ be the one-hot label. The sample-wise and dataset-wise cross-entropy losses are

$$\mathcal{L}_{\mathrm{CE}}(x, y; \theta) = -\, y^\top \log a, \quad \mathcal{L}_{\mathrm{CE}}(\mathcal{D}; \theta) = \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} \mathcal{L}_{\mathrm{CE}}(x, y; \theta).$$

2.1.2 Model Training

A neural network is a parameterised function $f(\cdot, \theta)$ that is commonly trained by solving

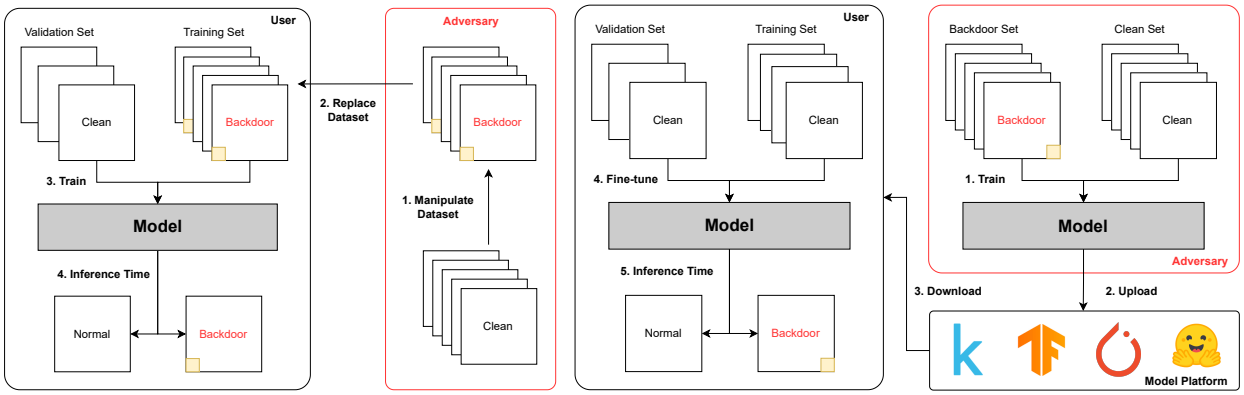
$$\theta^\star = \arg\min_\theta \, \mathcal{L}_{\mathrm{CE}}(\mathcal{D}_{\mathrm{train}}, \theta), \tag{1}$$

typically with some variation of gradient descent. For image classification, f is often a convolutional network that can be factorized into a feature extractor φ and a linear classifier ω:

$$f(x, \theta) = \omega\big(\varphi(x)\big), \quad \theta = \varphi \cup \omega, \; \varphi \cap \omega = \varnothing.$$

2.2 Backdoor Attacks

To execute a backdoor attack, various methods have been proposed in the literature, typically involving the creation of two datasets $\mathcal{D}_c$ (clean) and $\mathcal{D}_b$ (backdoor), which combine to form $\widetilde{\mathcal{D}}_t$, a poisoned variant of the original training dataset $\mathcal{D}_t$. A critical consideration in this process is the poisoning ratio, which represents the proportion of backdoor to clean data (i.e., $\frac{\|\mathcal{D}_b\|}{\|\mathcal{D}_c\|}$), balancing the trade-off between the attack's stealth and effectiveness. The clean dataset $\mathcal{D}_c$ consists of the original unaltered inputs x and their associated labels y. To generate the backdoor dataset $\mathcal{D}_b$, a backdoor function $B(x, \rho) \to \hat{x}, \hat{y}$ is used, where ρ denotes the trigger pattern added to x to produce the backdoored input $\hat{x}$. In targeted backdoor attacks, which are the most extensively studied in the literature, the label $\hat{y}$ is typically assigned a predefined value. Although alternative adversarial objectives, such as the all-target objective described in Zhao et al. (2020), have



(a) Data Poisoning (DP) (b) Outsourced Training (OT)

Figure 2: Threat models considered by existing backdoor attacks.

Table 3: Categorization of state-of-the-art backdoor attacks based on key characteristics and the training procedure employed by the adversary. DP: Data Poisoning, OT: Outsource Training.

Ref	Name	Trigger Characteristics			Threat Model
		Coverage	Consistency	Mode	
Gu et al. (2017)	BadNets	Local	Static	Replacement	DP
Chen et al. (2017)	Blended	Global	Static	Additive	DP
Barni et al. (2019)	Signal	Global	Static	Additive	DP
Zeng et al. (2021)	LF	Global	Static	Additive	DP
Li et al. (2021c)	SSBA	Global	Dynamic	Additive	DP
Nguyen & Tran (2020)	IAB	Local	Dynamic	Replacement	OT
Wang et al. (2022)	BPP	Global	Dynamic	Additive	OT
Nguyen & Tran (2021)	WaNet	Global	Dynamic	Warping	OT
Doan et al. (2021)	LIRA	Global	Dynamic	Additive	OT

been explored, targeted backdoor attacks remain the primary focus within this domain. Using $\widetilde{\mathcal{D}}_t$, θ is optimized to accurately classify both $\mathcal{D}_c$ and $\mathcal{D}_b$. Recent works, such as LIRA Doan et al. (2021)), have introduced specialized training procedures tailored to enhance the effectiveness of backdoor attacks, which, while relevant, fall outside the scope of this work.

Current backdoor attacks are primarily analyzed within two main threat models, as depicted in Figure 2. The first, the *Data Poisoning* threat model, assumes that the adversary's capabilities are limited to modifying the training data, allowing them to replace $\mathcal{D}_t$ with $\widetilde{\mathcal{D}}_t$. The second, more potent adversarial scenario is the *Outsourced Training* threat model, where the adversary has full control over the entire training process. In this setting, the adversary can not only substitute $\mathcal{D}_t$ with $\widetilde{\mathcal{D}}_t$, but also employ an arbitrary non-auditable training procedure. Using these threat models, along with additional characteristics introduced in studies such as Wu et al. (2022), we categorize prominent backdoor attacks relevant to image classification in Table 3. However, this work does not consider attacks that make additional assumptions such as those made in the backdoor reactivation attack proposed by Zhu et al. (2024a). Rather, we focus on a sample of the most popular general-purpose and representative backdoor attacks currently present in the literature to assess the robustness of existing methods to a diverse range of attacks.

2.2.1 Trigger Characteristics

Trigger characteristics pertain to the type of trigger ρ used by the adversary, encompassing coverage, consistency and modification mode, as defined in Wu et al. (2022). Below, we briefly discuss these characteristics with reference to the works listed in Table 3.

Coverage Trigger coverage refers to the extent of modification when ρ is integrated into the x by B Wu et al. (2022). In Figure 1, we provide examples of backdoor images generated by the considered attacks. Coverage is typically classified as global or local. Global coverage implies that ρ affects a significant portion of x. For instance, the Blended attack Chen et al. (2017) alters x by incorporating a trigger image (e.g., a picture of Hello Kitty) with a blending ratio that controls its transparency. In contrast, local triggers modify only a small portion of x. Among the attacks considered, BadNets Gu et al. (2017) and IAB Nguyen & Tran (2020) utilize local triggers. In the case of BadNets, a small $n \times n$ pixel pattern (e.g., a 3×3 white square) is inserted into images at a fixed position (e.g., at the bottom left-hand corner).

Consistency Trigger consistency refers to whether the same ρ is used across $\mathcal{D}_b$. When ρ is fixed, the attack is deemed static. For example, the BadNets attack inserts the same $n \times n$ pixel pattern into the same position in each image. However, the success of works such as Wang et al. (2019b) in identifying static triggers used by BadNets has led to the preference for dynamic triggers in recent works. More specifically, dynamic methods often make ρ dependent on x. For example, IAB generates a unique ρ for each $(x, y) \in \mathcal{D}_c$. Unlike other methods such as SSBA, BPP, and LIRA, which synthesise barely perceptible patterns, IAB generates patterns similar to those used in BadNets. Moreover, the triggers generated by IAB for different images are designed to be non-reusable, meaning the trigger for x does not work for x'.

Mode Trigger modification mode refers to how ρ is applied to x. The two most common modes are additive and replacement. Attacks using the replacement mode substitute parts of x with ρ. For example, BadNets replaces the pixels in the $n \times n$ region of x that overlaps with ρ. in contrast, additive attacks add ρ to x (i.e., $\hat{x} = x + \rho$). The exception to this categorization is the WaNet attack, where the elastic image warping method utilized in this attack cannot be classified as either additive or replacement.

2.3 Backdoor Mitigation

The landscape of defense against backdoor attacks encompasses several sub-problems, addressed by different bodies of work. For instance, some works propose methods that identify backdoor samples within the poisoned dataset $\widetilde{\mathcal{D}}_t$ Gao et al. (2019). Another common task is trigger synthesis, which involves generating the backdoor dataset $\mathcal{D}_b$ from the clean dataset $\mathcal{D}_c$ given the model parameters θ Liu et al. (2019). While these are important avenues of research, we concentrate on backdoor mitigation. Distinct from other defensive strategies, backdoor mitigation seeks to remove backdoor behaviour from the model while preserving its original classification capability Liu et al. (2018).

Although several works address backdoor mitigation, inconsistencies often arise concerning the threat model considered by each work. To ensure a fair comparison, we focus on proposals that adopt a set of assumptions consistent with the strongest adversarial scenario, specifically, the Outsourced Training threat model outlined in section 2.2. In this setting, the defender, who performs the mitigation, has access to θ and a small set of clean mitigation data $\mathcal{D}_m$. Crucially, the defender does not have access to any backdoor data $\mathcal{D}_b$. In addition, our review only considers the most prominate works published in prominent peer-reviewed venues, particularly, CORE[1] ranked A/A* conferences and SJR[2] ranked Q1 journals.

In the following sections, we present a comprehensive analysis of existing backdoor mitigation approaches within the image classification domain. While discussing each work, we attempt to summarise the methodology used by each work, as well as highlight important assumptions. We broadly group works into two main categories, pruning and fine-tuning, while also providing more granular sub-typing.

3 Model Pruning

When backdoor attacks were first introduced for image classification, Gu et al. (2017) hypothesized that models extract two distinct *feature* sets. They posited that certain model components, such as convolutional and or dense layers, become associated either with the main or backdoor task. Consequently, when the

[1] `https://www.core.edu.au/icore-portal`
[2] `https://www.scimagojr.com/journalrank.php`

Table 4: Categorization of the surveyed backdoor mitigation approaches.

Ref	Name	Category	Type
Liu et al. (2018)	FP	Pruning	Metric
Zheng et al. (2022a)	BNP		
Zheng et al. (2022b)	CLP		
Wu & Wang (2021)	ANP		Masking
Chai & Chen (2022)	AWN		
Li et al. (2023)	RNP		
Huang & Bu (2023)	FMP		
Karim et al. (2024)	NFT		
Wang et al. (2023)	MM-BD		Additive
Cheng et al. (2024)	UNIT		
Zhu et al. (2024b)	NPD		
Wang et al. (2019a)	NC*		Synthesis Unlearn
Qiao et al. (2019)	MESA	Fine-Tuning	Synthesis Unlearn
Liu et al. (2022)	BAERASER		
Min et al. (2024)	FST		Traditional
Zhu et al. (2023)	FT-SAM		
Xu et al. (2024b)	BTI-DBF		
Li et al. (2021a)	NAD		Knowledge Distillation
Pang et al. (2023)	BCU		
Zeng et al. (2022)	i-BAU		Adversarial Training
Mu et al. (2023)	PBE		
Wei et al. (2023)	SAU		

*While NC is grouped with MESA and BAERASER, it is indeed a pruning approach.

model is presented with $\hat{x}$, the components contributing to the backdoor task result in its classification as $\hat{y}$ rather than y.

Historically, model pruning has been employed to identify and remove redundant model components, thereby improving inference efficiency He & Xiao (2023). Building upon the hypothesis that model components specialise, several studies have explored the application of model pruning for backdoor mitigation. Pruning-based approaches aim to identify and eliminate model components associated with backdoor behaviour, while keeping those responsible for the clean task. To achieve this, various strategies have been adopted, including metric-based, masking-based, and additive techniques. In this section, we provide a comprehensive review of these subcategories, along with a comparative analysis of the approaches within them.

3.1 Metric-based Pruning

Works adopting a metric-based approach aim to directly quantify the contribution of each model component to the backdoor task. By applying a defined metric, these works distinguish between clean and backdoor components based on their respective metric values.

3.1.1 FP

The work of Liu et al. (2018) is considered a pioneering effort in backdoor mitigation. In their initial investigation, Liu et al. (2018) compares the channel-wise average activation produced by $\mathcal{D}_c$ and $\mathcal{D}_b$ in the final convolutional layer. Their analysis reveals that backdoor components, specifically a small subset of filters within this layer, are only activated by $\mathcal{D}_b$.

Building on this observation, Liu et al. (2018) proposes pruning the final convolutional layer based on the average channel-wise activation given $\mathcal{D}_m$, the mitigation dataset available to the defender. The approach involves iteratively pruning filters with the lowest average activation until the accuracy on a validation dataset $\mathcal{D}_v$, which is segmented from $\mathcal{D}_m$ before pruning, falls below a defined threshold. After pruning, Liu et al. (2018) fine-tune θ using $\mathcal{D}_m$ to recover any performance lost.

3.1.2 BNP

Similar to FP Liu et al. (2018), Zheng et al. (2022a) examine the activation values of backdoored models. However, Zheng et al. (2022a) compare the pre-activation distribution of clean and backdoor components. Pre-activation refers to the activation values before any non-linear transformation, e.g., ReLU, is applied. Their analysis reveals that clean components typically follow a unimodal distribution, while backdoor components exhibit a bimodal distribution (see (Zheng et al., 2022a, Fig. 2(a)-(b))).

As an extension of the above analysis, Zheng et al. (2022a) compare the batch normalization (BN) statistics (μ_{bn} and σ_{bn}) tracked in the proceeding layer with the pre-activation statistics of $\mathcal{D}_c$ (μ_c and σ_c). Their comparison shows that μ_{bn} and σ_{bn} are biased relative to μ_c and σ_c, a consequence of μ_{bn} and σ_{bn} stemming from the bimodal input distribution produced by $\widetilde{\mathcal{D}}_t$ (see (Zheng et al., 2022a, Fig. 2(c)-(d))).

To leverage this characteristic, Zheng et al. (2022a) calculate the Kullback-Leibler (KL) divergence between $\mathcal{N}(\mu_{bn}, \sigma_{bn})$ and $\mathcal{N}(\mu_m, \sigma_m)$. The KL divergence is calculated for each filter within each convolutional layer. The set of KL-divergence values for the l^{th} layer, denoted as $K_l = \{k_1, k_2 \ldots, k_n\}$, is then used to determine a layer-specific pruning threshold τ_l, calculated as

$$\tau_l = \bar{K}_l + \lambda s_l, \tag{2}$$

where $\bar{K}_l$ and s_l are the mean and standard deviation of K_l, and λ is a hyperparameter selected by the defender. The filters are then pruned based on τ_l. However, it is important to note that BNP assumes that a subset of filters exhibits biased BN statistics compared to the pre-activation statistics of $\mathcal{D}_c$. While this characteristic is demonstrated for a single filter in (Zheng et al., 2022a, Fig. 2(c)-(d)), the distribution of K_l across all layers is not provided, leaving limited evidence to support the assertion that filters in each layer can be distinctly separated using the proposed metric. Moreover, given the inherent complexity and high non-linearity in modern classification architectures, the assumption that the pre-activation distribution of all layers is Gaussian is also unlikely to hold.

3.1.3 CLP

Unlike other metric-based methods, Zheng et al. (2022b) propose using the *Lipschitz* constant associated with each filter matrix ϕ to guide pruning. For the i^{th} filter in the l^{th} layer, the upper bound channel Lipschitz constant (UCLC) of $\phi_{l,i}$ is estimated as the largest singular value from the spectral decomposition of $\phi_{l,i}$. Note that $\phi_{l,i}$ is reshaped such that $\phi_{l,i} \in \mathbb{R}^{c \times (hw)}$, where c, h, and w represent the channel, height, and width dimensions of $\phi_{l,i}$.

Operating under the hypothesis that backdoor components exhibit distinct activation patterns for $\mathcal{D}_c$ and $\mathcal{D}_b$, Zheng et al. (2022b) argue that the UCLC can effectively quantify this difference without needing access to either dataset. To validate this idea, Zheng et al. (2022b) introduces the trigger activation change (TAC), which quantifies the average activation difference between $\mathcal{D}_c$ and $\mathcal{D}_b$ for the i^{th} filter in the l^{th} layer, as

$$\frac{1}{|\mathcal{D}_c|} \sum_{(x,y) \in \mathcal{D}_c, (\hat{x}, \hat{y}) \in \mathcal{D}_b} \| f_{l,i}(x, \theta) - f_{l,i}(\hat{x}, \theta) \|_2, \tag{3}$$

where $(x, y) \in \mathcal{D}_c, (\hat{x}, \hat{y}) \in \mathcal{D}_b$ is a clean and backdoor image pair. By plotting UCLC and TAC against each other, Zheng et al. (2022b) demonstrate a strong positive correlation between the two metrics (see (Zheng et al., 2022b, Fig. 3)). They subsequently suggest that UCLC can reliably quantify the sensitivity of each filter to ρ, allowing for a distinction between clean and backdoor components. For model pruning, Zheng et al. (2022b) adopts the same layer-based thresholding approach as BNP Zheng et al. (2022a).

While the proposed metric is well-correlated in Zheng et al. (2022b) initial analysis, its generalizability to diverse model architectures is not well demonstrated. Given that the proposed UCLC metric is an upper bound, its sensitivity to changes in setting is largely unknown.

3.1.4 FMP

Distinct from other metric-based pruning methods, FMP Huang & Bu (2023) directly evaluates the contribution of individual convolutional filters to the model's classification performance. The key idea is to measure how sensitive a filter's feature maps are to input perturbations that maximize the difference between clean and perturbed feature maps.

To do this, FMP constructs adversarial examples using the Fast Gradient Sign Method (FGSM) to maximize the feature map change for a given filter. Formally, for the i^{th} filter in the l^{th} layer, they solve

$$\max_{\{\delta_{l,i}^{(x)}\}_{(x,y)\in\mathcal{D}_m}} \sum_{(x,y)\in\mathcal{D}_c,(\tilde{x},\tilde{y})\in\mathcal{D}_m} \|f_{l,i}(x,\theta) - f_{l,i}(\tilde{x},\theta)\|_2 \tag{4}$$

where

$$\widetilde{\mathcal{D}}_m = \{(\tilde{x},\tilde{y},x,y) \mid \tilde{x} = x + \delta_{l,i},\ (x,y)\in\mathcal{D}_m\}$$

denotes the set of perturbed inputs generated specifically for that filter. Here, $\delta_{l,i}$ is the filter-specific adversarial perturbation, $f_{l,i}(\cdot,\theta)$ denotes the feature map output of the i^{th} filter in layer l, and $\mathcal{D}_c$, $\mathcal{D}_m$ refer to the clean and mixed (or target) datasets respectively.

After computing the adversarial perturbations for each filter, the filters are ranked according to their induced feature map change under $\widetilde{\mathcal{D}}_m$. Finally, the top p proportion of filters with the largest impact on classification accuracy, i.e., those most sensitive to the perturbations—are pruned from the network.

A critical assumption of this approach is that any filter associated with the backdoor task is capable of triggering the backdoor behaviour on its own. In other words, even if multiple filters collectively contribute to the backdoor, the method assumes that each filter can independently induce the malicious behaviour. If the backdoor effect only emerges from the joint activation of multiple filters, FMP may underestimate the importance of such jointly-responsible filters.

3.2 Masking-based Pruning

Beyond metric-based approaches, some pruning methods focus on learning a parameter mask $\mathbf{m}$ that, when applied to θ, effectively removes the backdoor behaviour. These methods formulate an objective function and use optimization techniques to find an optimal mask $\mathbf{m}$. The mask is applied using the Hadamard (element-wise) product as $\mathbf{m}\odot\theta$. Hence, when an entry of $\mathbf{m}$ is zero, the corresponding entry in θ is pruned. Additionally, since $\mathbf{m}$ is used to mask convolutional filters, a mask value is learned for each filter in each layer.

3.2.1 ANP

In Wu & Wang (2021), the authors initially frame the backdoor mitigation task as a masking problem, where they examine the impact of a perturbation ξ applied to θ on the classification error over $\mathcal{D}_m$ for both clean and backdoored models. The perturbation set ξ consists of two subsets: ξ_w, applied to the weights (w) in θ, and ξ_b, applied to the biases (b) in θ. Therefore, $\xi = \xi_w \cup \xi_b$ and $\theta = w \cup b$. To determine ξ, the following optimization problem is solved

$$\max_{\xi_w,\xi_b\in[-\epsilon,\epsilon]} \mathcal{L}_{\text{CE}}(\mathcal{D}_m, [(1+\xi_w)\odot w \ \cup\ (1+\xi_b)\odot b]), \tag{5}$$

where ϵ constrains the values of ξ. In Wu & Wang (2021), it is shown that a backdoored model exhibits higher classification error (see (Wu & Wang, 2021, Fig. 1(a))). It is also observed that classification errors made by the backdoored model are biased towards the target class (see (Wu & Wang, 2021, Fig. 1(b))). Based on these results, Wu & Wang (2021) hypothesise that ξ targets backdoor components of the model. To leverage this characteristic, they propose solving the following minimax optimization problem

$$\min_{\mathbf{m}\in[0,1]} \left\{ \lambda\mathcal{L}_{\text{CE}}\left(\mathcal{D}_m, [(\mathbf{m}+\xi_w)\odot w, b]\right) + \max_{\substack{\xi_w,\xi_b \\ \in[-\epsilon,\epsilon]}} (1-\lambda)\mathcal{L}_{\text{CE}}\left(\mathcal{D}_m, [(\mathbf{m}+\xi_w)\odot w, (1+\xi_b)\odot b]\right) \right\} \tag{6}$$

where $\lambda \in [0, 1]$ is a trade-off coefficient chosen by the defender. Solving equation 6 yields a perturbation ξ that maximises the classification error of $\mathcal{D}_m$. At the same time, the outer minimisation results in a mask $\mathbf{m}$ that minimises the classification error of $\mathcal{D}_m$ given ξ. In Wu & Wang (2021), they alternate between solving the inner and outer sub-problems multiple times. Once a solution for $\mathbf{m}$ is found, it is binarised using a threshold value or a fixed pruning fraction.

While the proposed method is unique, the weak direct relationship between successive inner-maximisation steps has the potential to limit ANP's effectiveness. Given that the inner step applies the learned mask to the model weights, repeated execution creates a weak coupling between the past and future iterations.

3.2.2 AWM

The robustness of ANP Wu & Wang (2021) in limited data settings is analysed in Chai & Chen (2022). The analysis findings indicate that ANP becomes ineffective when fewer than 100 data samples are available (see (Chai & Chen, 2022, Fig. 1)). To overcome the impact of limited data on the performance of weight masking, Chai & Chen (2022) propose applying perturbations to inputs rather than to θ. They redesign the inner maximization problem to identify an input perturbation δ that maximises classification error when applied to $\mathcal{D}_m$. Given $\widetilde{\mathcal{D}}_m = \{(\tilde{x}, y) \mid \tilde{x} = x + \delta \mid (x, y) \in \mathcal{D}_m\}$, the inner maximization problem is expressed as

$$\max_{\|\delta\|_1 \leq \epsilon} \mathcal{L}_{\text{CE}}(\widetilde{\mathcal{D}}_m, \theta), \tag{7}$$

where $\|\delta\|_1$ is bound by ϵ. Notably, the same δ is applied to all elements of $\mathcal{D}_m$. Thus, Chai & Chen (2022) solve the following minimax optimization problem

$$\min_{\mathbf{m} \in [0,1]} \left\{ \lambda_1 \mathcal{L}_{\text{CE}}(\mathcal{D}_m, \mathbf{m} \odot \theta) + \lambda_2 \max_{\|\delta\|_1 \leq \epsilon} \left[\mathcal{L}_{\text{CE}}(\widetilde{\mathcal{D}}_m, \mathbf{m} \odot \theta) \right] + \lambda_3 \|\mathbf{m}\|_1 \right\}, \tag{8}$$

where λ_1, λ_2, and λ_3 are hyperparameters that balance the influence of the three loss terms, and $\|\mathbf{m}\|_1$ serves an additional regularization term to encourage sparsity. Note that Chai & Chen (2022) omit the final binarization step employed in ANP, retaining $\mathbf{m}$ as a soft mask.

Promoting sparsity in $\mathbf{m}$ leads to significant pruning of θ, which is expected to result in low bias and high variance given $\mathcal{D}_m$. Moreover, this additional term is not normalised to account for the size of $\mathbf{m}$. Therefore, identifying optimal values for λ_1, λ_2, and λ_3 that yield consistent performance across different model architectures is challenging. Moreover, this design also suffers from the same weak coupling issue discussed above.

3.2.3 RNP

Distinct from ANP Wu & Wang (2021) and AWN Chai & Chen (2022), Li et al. (2023) introduce a unique unlearning strategy. Rather than learning ξ or δ, Li et al. (2023) first *unlearn* the clean task by solving the following optimization problem

$$\max_{\theta} \mathcal{L}_{\text{CE}}(\mathcal{D}_m, \theta). \tag{9}$$

The resulting set of unlearned parameters is denoted as $\hat{\theta}$. In Li et al. (2023), it is argued that this process leads to unlearning of the model's clean components while preserving the backdoor. Moreover, they suggest that the unlearned model exhibits biased misclassification toward the backdoor target.

Using $\hat{\theta}$, Li et al. (2023) then proceed to learn $\mathbf{m}$ by solving the following optimization problem

$$\min_{\mathbf{m} \in [0,1]} \mathcal{L}_{\text{CE}}(\mathcal{D}_m, \mathbf{m} \odot \hat{\theta}). \tag{10}$$

The authors of Li et al. (2023) assert that this recovery procedure can differentiate between clean and backdoor filters, given that $\hat{\theta}$ exhibits biased misclassification towards the backdoor target. Specifically,

they suggest that this procedure removes backdoor filters by setting their corresponding elements in $\mathbf{m}$ to 0. Importantly, once an optimal solution to $\mathbf{m}$ is obtained, it is applied to the original parameters, θ. In a manner similar to ANP, Li et al. (2023) binarize $\mathbf{m}$ using a threshold value or a fixed pruning ratio.

3.2.4 NFT

replaces the adversarial trigger synthesis used in prior backdoor mitigation methods, such as AWN, with the MixUp data augmentation strategy. Given two clean datapoints i and j from $\mathcal{D}_c$, MixUp forms augmented samples

$$\tilde{x}_{i,j} = \gamma x_i + (1-\gamma)x_j, \quad \tilde{y}_{i,j} = \gamma y_i + (1-\gamma)y_j, \tag{11}$$

where $\gamma \in [0,1]$ is drawn from a Beta distribution $\beta_{\alpha,\beta}$ and y_i, y_j are one-hot vectors. Using this data augmentation strategy, NFT minimises the MixUp objective

$$\mathcal{L}_{\text{MixUp}}(\mathcal{D}_c, \theta, \mathbf{m}) = \frac{1}{|\mathcal{D}_c|} \sum_{(x_i,y_i)\in\mathcal{D}_c} \mathbb{E}_{\gamma\sim\beta_{\alpha,\beta}} \left[\mathcal{L}_{\text{CE}}\big(\tilde{x}_{i,j}, \tilde{y}_{i,j}; \theta \odot \mathbf{m}\big)\right], \tag{12}$$

$$\min_{\mathbf{m}} \ \mathcal{L}_{\text{MixUp}}(\mathcal{D}_c, \theta, \mathbf{m}) + \lambda\|\mathbf{m}\|_1, \quad \lambda = \frac{5\times10^{-4}}{n_c}, \tag{13}$$

where n_c is the number of classes. Moreover, they set a dynamic layer-wise lower bound on $\mathbf{m}$, which decreases with depth to encourage pruning of deeper layers.

Theoretical analysis in shows that $\mathcal{L}_{\text{MixUp}}$ serves as an upper bound on the ideal purification loss

$$\mathcal{L}^*_{\text{purify}}(\mathcal{D}_b, \theta, \mathbf{m}) = \frac{1}{|\mathcal{D}_b|} \sum_{(x,y)\in\mathcal{D}_b} \mathcal{L}_{\text{CE}}\big(\hat{x}, y; \theta \odot \mathbf{m}\big), \tag{14}$$

where $\mathcal{D}_b$ is the inaccessible backdoor dataset and y is the original label associated with $\hat{x}$.

A critical limitation of the paper's theoretical argument is its assumption about the nature of the backdoor data. The original MixUp proof posits that an effective defence rests on the premise that a backdoor trigger can be modelled as a small-magnitude perturbation of a clean image. This allows the ideal loss on a triggered sample to be mathematically bounded by the loss on samples created through MixUp. However, this assumption doesn't hold for many common backdoor attacks. Triggers like BadNets or other distinct colored regions are not small, distributed noise. Instead, they are large, localized changes to the input. In these cases, the backdoor data point does not necessarily lie on the manifold between pairs of clean data points in the way the proof assumes.

3.3 Additive Pruning

The pruning methods discussed thus far either directly prune or mask existing model components. In contrast, Wang et al. (2023), Cheng et al. (2024) and Zhu et al. (2024b) introduce additional model components that integrate with the existing structure. These additional components function as quasi-filters, targeting the removal of backdoor tasks through a mechanism akin to pruning. Consequently, we categorize these approaches loosely under the umbrella of pruning.

3.3.1 MM-BD and UNIT

In Wang et al. (2023) and Cheng et al. (2024), the task of backdoor mitigation is framed as a bounding problem. Initially, both works observe that backdoor samples tend to trigger large activations, with Wang et al. (2023) showing that this results in unusually high decision-making confidence (see (Wang et al., 2023, Fig. 9)). To quantify this difference in confidence, Wang et al. (2023) calculate the maximum margin statistic of x given y as

$$\mathcal{G}(x, y, \theta) = s_y(a) - \max_{k\in\mathcal{M}\setminus y} s_k(a), \tag{15}$$

where s_n selects the nth logit from the softmax output a given x, and $\mathcal{M}$ represents the set of possible labels. Their analysis reveals that backdoor samples exhibit significantly larger confidence compared to clean samples (see (Wang et al., 2023, Fig. 4(a))). The authors conjecture that this is due to the abnormally large activations influencing the model's decision-making.

To counteract the impact of these large activations have on decision-making, Wang et al. (2023) propose learning a set of upper-bound values $\mathbf{B} = \{b_1, \ldots, b_L\}$ for each non-linear activation layer, such as ReLU. These bounds are learned in a channel-wise manner and used during the forward pass to constrain activation range within each channel. To learn $\mathbf{B}$, the following optimization problem is solved

$$\min_{\mathbf{B}} \frac{1}{|\mathcal{D}_m| \times |\mathcal{M}|} \sum_{(x,y)\in\mathcal{D}_m} [f(x, \theta_{\mathbf{B}}) - f(x, \theta)]^2 + \lambda \|\mathbf{B}\|_2, \tag{16}$$

where $\theta_{\mathbf{B}}$ represents the model parameters θ combined with the learned bounding values $\mathbf{B}$. The objective is to find the bounding values $\mathbf{B}$ that minimally impact the classification of $\mathcal{D}_m$.

Similar to Wang et al. (2023), Cheng et al. (2024) find the minimum layer-wise upper bounds $\mathbf{B}$ that do not impact the classification of $\mathcal{D}_m$. The upper bound b_i is initialized as $b_i = \mu_i + 4\sigma_i$, where μ_i and σ_i are the mean and standard deviation of the activation distribution produced by $\mathcal{D}_m$ within layer i. They then solve the following optimization problem

$$\min_{\mathbf{B}} \mathcal{L}_{\text{CE}}(\mathcal{D}_m, \theta_{\mathbf{B}}) + \lambda \|\mathbf{B}\|_1, \tag{17}$$

Similar to AWN, the regularization term used by both approaches is not normalised to account for variations in model architectures. In addition, this term does not adjust for differences in the scale of activation values across layers. Variations in activation scale are likely to directly affect the values of $\mathbf{B}$ and influence the choice of λ. Another critical assumption made is that setting an upper bound for each channel is sufficient to restore correct classification of $\mathcal{D}_b$, with both proposals only reporting if ASR is reduced.

3.3.2 NPD

Unlike MM-BD Wang et al. (2023) and UNIT Cheng et al. (2024), Zhu et al. (2024b) approaches backdoor mitigation from a more traditional pruning perspective. To implement pruning, Zhu et al. (2024b) introduces a 1x1 convolutional layer into the model, suggesting that this layer is added close to the final layer of the feature extraction. Referred to as a polarizer, this layer learns a set of parameters $\boldsymbol{w}$ that filter channels associated with the backdoor task. The 1x1 convolutional layer maintains the same number of channels as the preceding layer, and therefore scales each output channel by the corresponding value in $\boldsymbol{w}$. The augmented model $f_{w,\theta}$ is parameterized by θ and $\mathbf{w}$. However, θ remains fixed throughout and therefore is excluded from subsequent equations.

Similar to AWN Chai & Chen (2022), Zhu et al. (2024b) first approximates the backdoor trigger ρ as an input perturbation δ. However, unlike AWN, Zhu et al. (2024b) models the trigger distribution in a sample-specific manner, learning a distinct value of δ for each $(x, y) \in \mathcal{D}_m$. Given x, δ is learned by solving the following optimization problem

$$\min_{\|\delta\|_p \leq \epsilon} \mathcal{L}_{\text{CE}}(x + \delta, \tilde{y}, \mathbf{w}), \tag{18}$$

where $\tilde{y}$ is an estimate of the target class, for which Zhu et al. (2024b) provide several heuristics. Under the assumption that the defender does not know the backdoor target, Zhu et al. (2024b) suggests using the second-largest logit for x. However, Zhu et al. (2024b) does not quantitatively validate how frequently the second-largest logit corresponds to the backdoor target. Using $\widetilde{\mathcal{D}}_m = \{(\tilde{x}, \tilde{y}, x, y) \mid \tilde{x} = x + \delta \mid (x, y) \in \mathcal{D}_m\}$, they solve the following optimization problem

$$\mathcal{L}_{\text{ASR}}(\tilde{x}, \tilde{y}, \mathbf{w}) = -\log(1 - s_{\tilde{y}}(\tilde{a})), \tag{19}$$

$$\mathcal{L}_{\text{BCE}}(\tilde{x}, y, \mathbf{w}) = -\log(s_y(\tilde{a})) - \log(1 - \max_{k \neq y} s_k(\tilde{a})), \tag{20}$$

$$\min_{\mathbf{w}} \left\{ \frac{1}{|\widetilde{\mathcal{D}}_m|} \sum_{(\tilde{x}, \tilde{y}, x, y) \in \widetilde{\mathcal{D}}_m} \lambda_1 \mathcal{L}_{\text{CE}}(x, y, \mathbf{w}) + \lambda_2 \mathcal{L}_{\text{ASR}}(\tilde{x}, \tilde{y}, \mathbf{w}) + \lambda_3 \mathcal{L}_{\text{BCE}}(\tilde{x}, y, \mathbf{w}) \right\}, \tag{21}$$

where $\tilde{a} = f(\tilde{x}, \theta)$, and λ_1, λ_2 and λ_3 are hyperparameters that control the influence of each term. This design optimizes $\mathbf{w}$ to alleviate the impact of δ when applied to $\mathcal{D}_m$. The term $\mathcal{L}_{\text{ASR}}$ penalizes the classification of $\tilde{x}$ as $\tilde{y}$, while $\mathcal{L}_{\text{BCE}}$, similar to the margin statistic proposed in Wang et al. (2023), encourages confident classification of $\tilde{x}$ as y, its correct label. Finally, the loose coupling issue associated with ANP and AWN is not resolved in NPD, as the polariser is used when estimating δ.

3.4 Summary

Model pruning is an important strategy in mitigating backdoor attacks, building on the hypothesis that neural networks can be decomposed into distinct components responsible for either clean or backdoor tasks. By selectively pruning the components related to backdoor behaviours, researchers aim to restore model integrity while preserving performance on the original task.

Metric-based pruning approaches quantify each model component's contribution to backdoor behavior through various metrics. Approaches like FP remove filters based on their activation patterns, while BNP utilizes distribution statistics to identify backdoor components. CLP's Lipschitz-based approach uses approximations of the channel Lipschitz constants to guide pruning decisions. Masking-based pruning techniques optimize a parameter mask to remove backdoor functionality. Approaches such as ANP, AWM, RNP and NFT employ optimization frameworks to iteratively refine the mask, targeting backdoor components while retaining model accuracy. Lastly, additive pruning approaches introduce new components into the network, functioning as filters that mitigate backdoor influences without directly removing existing filters.

4 Fine Tuning

An alternative approach to model pruning for backdoor mitigation is fine-tuning. Instead of removing a subset of θ, fine-tuning methods adjust the values of θ to eliminate the backdoor. A key aspect of these methods is the objective function, which typically incorporates one or more carefully designed regularisation terms. These terms are usually selected based on specific insights gained from preliminary investigations. In this section, we review the most prominent fine-tuning approaches, categorized into distinct subgroups based on their unique methodologies.

4.1 Conventional Fine-Tuning

The first subcategory of fine-tuning methods follows a more conventional approach. Here, conventional refers to the proposed optimization problem being closely aligned with equation 1.

4.1.1 FST

In their preliminary investigation, Min et al. (2024) evaluate the effectiveness of minimisation equation 1 given $\mathcal{D}_m$ for mitigating backdoor tasks. Their analysis decomposes θ into φ, parameters associated with the feature extractor, and ω, parameters associated with the linear classifier. They tested fine-tuning various combinations of φ and ω, concluding that the minimisation equation 1 for any combination of these parameters was largely ineffective in removing backdoors (see (Min et al., 2024, Table 1)).

Building on these findings, Min et al. (2024) propose reinitializing ω by assigning new random values to ω before jointly fine-tuning both φ and ω. To enhance this process, they introduce a regularisation term that

encourages divergence between the original and new values of ω, referred to as $\hat{\omega}$ and ω, respectively. This leads to minimizing the following objective function

$$\min_{\theta} \mathcal{L}_{\text{CE}}(\mathcal{D}_m, \theta) + \lambda \omega^{\intercal} \hat{\omega}, \quad \text{s.t. } \|\omega\|_2 = \|\hat{\omega}\|_2, \tag{22}$$

where λ is a hyperparameter controlling the influence of the regularization term. According to Min et al. (2024), $\omega^{\intercal}\hat{\omega}$ discourages ω from learning the same relationships between the penultimate set of features and class labels. The constraint $\|\omega\|_2 = \|\hat{\omega}\|_2$ is applied to minimise the impact of $\omega^{\intercal}\hat{\omega}$ on the overall loss during fine-tuning. Critically, this additional term improves the unnormalised loss components used by AWN and MM-BD.

4.1.2 FT-SAM

In addition, Zhu et al. (2023) investigate the effectiveness of traditional fine-tuning in mitigating backdoors. Similar to FST Min et al. (2024), they find it ineffective. Their analysis of the ω norms, the magnitudes of each parameter, revealed minimal changes following fine-tuning (see (Zhu et al., 2023, Fig. 2)). They hypothesise that fine-tuning fails because it is unable to escape the local minima it finds, allowing the backdoor to persist. Additionally, Zhu et al. (2023) demonstrates that ω norms are positively correlated with TAC [see equation 3], a metric previously used in CLP Zheng et al. (2022b) to quantify activation difference between $\mathcal{D}_c$ and $\mathcal{D}_b$.

Inspired by sharpness-aware minimisation (SAM) methods Foret et al. (2020), Zhu et al. (2023) propose a minimax optimisation method designed to escape sharp local minima. Similar to ANP Wu & Wang (2021), the inner minimisation seeks an ℓ_2-bounded perturbation ξ that maximises the classification loss for $\mathcal{D}_m$. However, rather than identifying a weight mask $\mathbf{m}$, they fine-tune θ directly in the outer minimisation step. This leads to solving the following optimization problem

$$\min_{\theta} \max_{\left\|\mathbf{T}_{\theta}^{-1}\xi\right\|_2 \leq \epsilon} \mathcal{L}_{\text{CE}}(\mathcal{D}_m, \theta + \xi), \tag{23}$$

where $\mathbf{T}_{\theta} = \text{diag}\left(|\theta_1|, \ldots, |\theta_L|\right)$, with θ_i being the i^{th} parameter in θ and ϵ serving as a hyperparameter controlling the perturbation budget. The traditional ℓ_2-bounding constraint is modified to $\|\mathbf{T}_{\theta}^{-1}\xi\|_2 \leq \epsilon$, allowing larger perturbations to be applied to elements of θ with larger norms, as their corresponding component in $\mathbf{T}_{\theta}^{-1}$ approaches zero. This evaluates the stability of θ, quantified as $\mathcal{L}_{\text{CE}}$ on $\mathcal{D}_m$, when perturbed by ξ.

The perturbation constraint $\|\mathbf{T}_{\theta}^{-1}\xi\|_2 \leq \epsilon$ used by Zhu et al. (2023) is applied during each gradient decent step. This implies that θ can drift significantly from its initial values after several steps. Moreover, since the estimation of ξ relies on $\mathcal{D}_m$, it is prone to having low bias and high variance.

4.2 Knowledge Distillation

Inspired by its success in other learning settings, two proposals explore how knowledge distillation (KD) can be leveraged for backdoor mitigation. Traditionally used to *transfer knowledge* from larger to smaller models, KD has been effectively applied in tasks such as image classification Gou et al. (2021). In the context of backdoor mitigation, the distillation process is reframed as a *knowledge filtering* task. Instead of transferring all knowledge from the original model, the goal of *knowledge filtering* is to distill only the information relevant to the clean task, thereby eliminating the backdoor-related information.

A common approach to KD involves a teacher-student architecture. In typical applications, the teacher model is a larger, more capable model whose knowledge is transferred to a smaller student model. However, in the case of backdoor mitigation, access to a non-backdoored teacher model is not possible. Subsequently, approaches that employ this method must overcome this challenge.

4.2.1 NAD

To implement KD, Li et al. (2021a) introduce a new attention-based method. Rather than relying on feature maps (i.e., intermediate activation outputs) to perform KD, Li et al. (2021a) suggest using attention maps. These maps compress the channel dimension of the feature maps, utilizing an attention operator $\mathcal{A}$, which maps from $\mathbb{R}^{c\times h\times w}$ to $\mathbb{R}^{h\times w}$. They propose the following two variants

$$\mathcal{A}^p_{\text{sum}}(x,\theta,l)=\sum_{i=1}^{c}|f_{l,i}(x,\theta)|^p,\;\; \mathcal{A}^p_{\text{mean}}(x,\theta,l)=\frac{1}{c}\sum_{i=1}^{c}|f_{l,i}(x,\theta)|^p \tag{24}$$

where $f_{l,i}$ is the i^{th} channel activation of x at the l^{th} layer and $p>1$. To perform KD within the teacher-student framework, Li et al. (2021a) first fine-tune θ using $\mathcal{D}_m$ resulting in a teacher model with parameters θ_T. The student's parameters, θ_S, are the original parameters θ that have not been fine-tuned. The KD is then performed by comparing the activation maps between the teacher and student models. To achieve this, Li et al. (2021a) use $\mathcal{A}^p_{sum}$ to define a distillation loss as

$$\mathcal{L}_{\text{NAD}}(x,\theta_T,\theta_S,l)=\left\|\frac{\mathcal{A}(x,\theta_T,l)}{\|\mathcal{A}(x,\theta_T,l)\|_2}-\frac{\mathcal{A}(x,\theta_S,l)}{\|\mathcal{A}(x,\theta_S,l)\|_2}\right\|_2, \tag{25}$$

and solve the following optimization problem

$$\min_{\theta_S}\mathcal{L}_{\text{CE}}(\mathcal{D}_m,\theta_S)+\frac{\lambda}{|\mathcal{D}_m|}\sum_{(x,y)\in\mathcal{D}_m}\sum_{l=1}^{L}\mathcal{L}_{\text{NAD}}(x,\theta_T,\theta_S,l), \tag{26}$$

where λ controls the contribution of the distillation loss. According to Li et al. (2021a), the inclusion of $\mathcal{L}_{\text{NAD}}$ helps regularise θ by aligning the activation maps of θ_S and θ_T thus removing the backdoor behaviour. However, since knowledge is distilled from a fine-tuned version of θ, the effectiveness of this approach is unclear if fine-tuning does not successfully remove the backdoor. This concern was underscored by the initial findings of both FST and FT-SAM, where fine-tuning alone proved ineffective at eliminating the backdoor.

4.2.2 BCU

In contrast to NAD Li et al. (2021a), Pang et al. (2023) propose using softmax probabilities a of x to perform KD rather than attention maps. Specifically, Pang et al. (2023) compare the temperature-scaled softmax probability scores $\tilde{a}_T=f(x,\theta_T)$ and $\tilde{a}_S=f(x,\theta_S)$, produced by θ_S and θ_T, respectively, to facilitate KD. To compare $\tilde{a}_T$ and $\tilde{a}_S$, Pang et al. (2023) employ KL-Divergence $\mathcal{L}_{\text{KL}}$ and solve the following optimization problem

$$\min_{\theta_S}\mathcal{L}_{\text{KL}}(\tilde{a}_T,\tilde{a}_S). \tag{27}$$

Rather than fine-tuning θ to produce θ_T, Pang et al. (2023) reinitialise a subset of θ_S, note that, $\theta_T=\theta$. Unlike FST Min et al. (2024), Pang et al. (2023) uniformly reinitialize n proportion ($0\leq n\leq 1$) of the parameters within each layer, with n increasing for deeper layers. They identified this reinitialization strategy as the best approach through a series of experiments. However, Pang et al. (2023) assert that the reinitialization step significantly impairs the model's ability to perform both clean and backdoor tasks. Consequently, when minimizing the proposed objective function, only the knowledge related to the clean task is transferred from the teacher to the student, severing the link between the trigger pattern and the backdoor task.

Unlike previous proposals, the use of KL-divergence by Pang et al. (2023) makes their approach dataset-agnostic. As a result, a defender can use any labeled or unlabeled in- or out-of-distribution dataset compatible with their model (i.e., having the same input dimensionality) to perform backdoor mitigation.

4.3 Synthesis Unlearn

The fine-tuning approaches discussed thus far utilise $\mathcal{D}_m$ to fine-tune θ. An alternative strategy involves synthesizing a set of surrogate backdoor data $(\tilde{x}, \tilde{y}) \in \widetilde{\mathcal{D}}_m$, which is used alongside $\mathcal{D}_m$ for fine-tuning. Here, $\tilde{x}$ and $\tilde{y}$ represent the surrogate backdoor data. The inclusion of this initial synthesis step allows these approaches to exploit information from $\widetilde{\mathcal{D}}_m$ to directly unlearn the backdoor task.

4.3.1 MESA

To synthesize $\tilde{x}$, Qiao et al. (2019) propose training a generative model G, parameterized by γ, to replicate the trigger distribution used by the adversary. To train G, they introduce a new maximum-entropy staircase approximation algorithm. This algorithm trains G as a combination of n sub-models that collectively generate a candidate trigger for a given input x. However, using a set of sub-models to train G requires the defender to know the trigger's position, approximate size, and the backdoor target. The optimization problem they solve is

$$\min_{\theta} \frac{1}{|\mathcal{D}_m|} \sum_{x,y \in \mathcal{D}_m} [\lambda \mathcal{L}_{\text{CE}}(x, y, \theta) + (1 - \lambda)\mathcal{L}_{\text{CE}}(\tilde{x}, y, \theta)] \tag{28}$$

where $\tilde{x} = x + G(\gamma)$ and $\lambda \in [0, 1]$ is a hyperparameter selected by the defender to control how many elements of $\mathcal{D}_m$ have δ applied. This approach aims to balance restoring the classification of $\tilde{x}$ to y while preserving the original classification performance.

4.3.2 BAERASER

Inspired by MESA Qiao et al. (2019), Liu et al. (2022) adopt the same synthesis method for generating surrogate backdoor data but proposes a different fine-tuning step. Using G, they generate a surrogate backdoor dataset $\widetilde{\mathcal{D}}_m$ that is used in conjunction with $\mathcal{D}_m$ to unlearn the backdoor task. The surrogate dataset is defined as $\widetilde{\mathcal{D}}_m = \{(\tilde{x}, \tilde{y}) \mid \tilde{x} = x + \delta \mid \delta = G(x, \gamma) \mid (x, y) \in \mathcal{D}_m\}$, assuming the defender has access to $\tilde{y}$. To perform unlearning, Liu et al. (2022) solve the following optimization problem

$$\min_{\theta} \lambda_1 [\mathcal{L}_{\text{CE}}(\mathcal{D}_m, \theta) - \mathcal{L}_{\text{CE}}(\widetilde{\mathcal{D}}_m, \theta)] + \lambda_2 \sum_{l=1}^{L} w_l \|\theta_l - \bar{\theta}_l\|_1, \tag{29}$$

where λ_1 and λ_2 control the strength of the two loss terms. The first term encourages misclassification of $\widetilde{\mathcal{D}}_m$ by subtracting its loss from that of $\mathcal{D}_m$. However, this term is unbounded and can dominate the optimization after a few iterations. The second loss term regularizes the solution by minimising the layer-wise distance between θ and the original value $\bar{\theta}$, using a layer-wise scalar weight w_l.

4.3.3 NC

Unlike both MESA and BAEARSER, Wang et al. (2019a) propose a method that removes the assumption of knowing the backdoor target and the approximate size and position of ρ. To achieve this, Wang et al. (2019a) learn an input perturbation δ that replaces specific image pixels using a binary mask $\mathbf{m}$. Here, δ and $\mathbf{m}$ are 3D and 2D matrices, respectively, with width and height dimensions matching x. To apply δ to x given $\mathbf{m}$, the function $A(x, \mathbf{m}, \delta) \rightarrow \hat{x}$ is utilized where if $\mathbf{m}_{j,i} = 1$, A replaces the pixel in the j^{th} row and i^{th} column of x with the corresponding value in δ. To learn δ and $\mathbf{m}$, Wang et al. (2019a) solve the following optimization problem

$$\min_{\mathbf{m}, \delta} \sum_{x \in \mathcal{D}_m} \mathcal{L}_{\text{CE}}(A(x, \mathbf{m}, \delta), t) + \lambda \|\mathbf{m}\|_1, \tag{30}$$

where λ controls the strength of the second regularisation term, which promotes sparsity in the solution for $\mathbf{m}$. Since t is unknown to the defender, a unique solution for δ and $\mathbf{m}$ is determined separately for

each class. An anomaly detection mechanism, using median absolute deviation, is then employed to identify anomalous class pairs. If such a pair is found, the proposed approach prunes the final dense layer of the model to mitigate the effect of the backdoor. To perform model pruning, the TAC metric [cf. equation 3] is used. Neurons that exhibit the largest average activation difference when δ is applied to D_m using A are iteratively pruned. Despite relying on model pruning, the method shares key similarities with MESA and BAEARSER, making it relevant to this section.

4.3.4 BTI-DBF

Rather than view pruning and fine-tuning as two separate approaches, Xu et al. (2024a) considers how they can be used in combination. More specifically, they use a three-stage process to first decouple the benign and backdoor features through feature masking, then learn a generator network to synthesise triggers, and finally fine-tune the original model parameters using the generator network.

During the first stage, a soft mask $\mathbf{m}$ is learned and applied to the feature map produced by the final feature extraction layer. To formalise this, let $f(x, \theta; \mathbf{m})$ denote the model's output where the feature map is element-wise multiplied by the mask $\mathbf{m}$. To learn $\mathbf{m}$ given a set of clean data D_m, they solve the following optimisation problem:

$$\min_{\mathbf{m}} \mathcal{L}_{\text{CE}}(\mathcal{D}_m, \theta; \mathbf{m}) - \mathcal{L}_{\text{CE}}(\mathcal{D}_m, \theta; (1 - \mathbf{m})). \tag{31}$$

The objective encourages the features selected by $\mathbf{m}$ to be sufficient for correct classification, while encouraging the remaining features (selected by $1 - \mathbf{m}$) to be insufficient.

Using the learned mask $\mathbf{m}$, a generator network $G : \mathcal{X} \to \mathcal{X}$ is then trained to generate perturbed versions of D_m, denoted as $\widetilde{\mathcal{D}}_m = \{(\tilde{x}, y) \mid \tilde{x} = G(x) \text{ for } (x, y) \in \mathcal{D}_m\}$. To train G, they solve the following optimisation problem:

$$\begin{aligned} \min_{G} \quad & \|(f_l(x, \theta) - f_l(G(x), \theta)) \odot \mathbf{m}\|_2 - \|(f_l(x, \theta) - f_l(G(x), \theta)) \odot (1 - \mathbf{m})\|_2, \\ s.t. \quad & \|x - G(x)\|_2 \leq \tau, \ \forall (x, y) \in \mathcal{D}_m, \end{aligned} \tag{32}$$

where f_l is the feature map from the last feature extraction layer. This objective trains the generator to create a poisoned sample $G(x)$ whose representation is similar to the original sample x in the benign features (masked by $\mathbf{m}$) but dissimilar in the presumed backdoor features (masked by $1 - \mathbf{m}$).

Finally, using G, the original set of model parameters θ is fine-tuned by solving the following optimisation problem:

$$\min_{\theta} \ \mathcal{L}_{\text{CE}}(\mathcal{D}_m, \theta) + \mathcal{L}_{\text{CE}}(\widetilde{\mathcal{D}}_m, \theta) + \sum_{(x,y) \in \mathcal{D}_m} \|f_l(x, \theta) - f_l(G(x), \theta)\|_2. \tag{33}$$

Together, Xu et al. (2024a) claim that these stages decouple benign and backdoor features, use this decoupling to learn a trigger generator, and then leverage the generator to unlearn the backdoor associations. While this approach is unique, we highlight a few critical limitations. Firstly, the decoupling and generative training steps each use a negative term that acts as a maximisation objective. Despite both terms being indirectly bound by a constraint (i.e., $\mathbf{m} \in [0, 1]$ and $\|x - G(x)\|_2 \leq \tau$), the dominance of each term within Equation 31 and 32 across each iteration is not evaluated. Moreover, in Equation 32, classification differences are not strictly enforced, as the ℓ_2 norm between the channels selected by $1 - \mathbf{m}$ is maximised.

4.4 Adversarial Training

Instead of approximating δ in a discrete step, recent works have incorporated concepts from adversarial training to perform mitigation. In essence, these approaches alternate between an adversarial objective and a mitigation objective. However, the critical distinction lies in the design of the adversarial objective.

Unlike traditional adversarial examples, the adversarial objective is specifically tailored to generate surrogate backdoor images. This ensures that the outer mitigation objective remains effective, allowing the model to unlearn the backdoor task while preserving performance on the clean task.

4.4.1 PBE

In Mu et al. (2023), the authors explore the behaviour of untargeted adversarial attacks on backdoored models. Given x and y, they generate an input perturbation δ by solving the following adversarial optimization problem

$$\max_{\|\delta\|_2 \leq \epsilon} \mathcal{L}_{\text{CE}}(\tilde{x}, y, \theta), \tag{34}$$

where ϵ controls the strength of the perturbation and $\tilde{x} = x + \delta$. Upon analyzing the classification of $\tilde{x}$, Mu et al. (2023) observed that a backdoored model tends to classify $\tilde{x}$ as the backdoor target, whereas a benign model produces a uniform distribution (see (Mu et al., 2023, Fig. 4)). Hence, they hypothesised that $\tilde{x}$ interacts with the backdoored model similarly to $\hat{x}$, the actual backdoor version of x. However, it is important to note that the proportion of samples classified as the target class in (Mu et al., 2023, Fig. 4) does not exceed 61%.

To exploit this observation, Mu et al. (2023) propose a fine-tuning strategy where θ is trained using both $\mathcal{D}_m$ and $\widetilde{\mathcal{D}}_m = \{(\tilde{x}, y) \mid \tilde{x} = x + \delta \mid (x, y) \in \mathcal{D}_m\}$. They alternate between solving the following two optimization problems

$$\min_{\theta} \mathcal{L}_{\text{CE}}(\mathcal{D}_c, \theta), \quad \min_{\theta} \mathcal{L}_{\text{CE}}(\widetilde{\mathcal{D}}_m, \theta), \tag{35}$$

where δ is computed using the PGD attack Madry et al. (2018).

4.4.2 i-BAU

In Zeng et al. (2022), the authors propose a redesigned adversarial objective aimed at identifying a universal input perturbation δ, which functions similarly to AWN Chai & Chen (2022). Here, universal refers to a perturbation that applies to all elements within $\mathcal{D}_m$. Using $\widetilde{\mathcal{D}}_m = \{(\tilde{x}, y) \mid \tilde{x} = x + \delta \mid (x, y) \in \mathcal{D}_m\}$, Zeng et al. (2022) set up the following minimax optimization problem

$$\min_{\theta} \max_{\|\delta\|_2 \leq \epsilon} \mathcal{L}_{\text{CE}}(\widetilde{\mathcal{D}}_m, \theta). \tag{36}$$

However, their experiments (see (Zeng et al., 2022, Fig. 1)) reveal that solving this minimax problem directly often yields unstable and unreliable results. This instability is attributed to the inner maximisation step failing to find an optimal solution for δ. To alleviate this issue, Zeng et al. (2022) propose solving the outer minimisation step using the following gradient

$$\nabla_{\theta} \mathcal{L}_{\text{CE}}(\widetilde{\mathcal{D}}_m, \theta) + (\nabla \delta)^T \nabla_{\delta} \mathcal{L}_{\text{CE}}(\widetilde{\mathcal{D}}_m, \theta), \tag{37}$$

where $\nabla \delta$ is the *response Jacobian* of the inner maximisation problem. Given that the inner maximisation step produces a suboptimal solution for δ, they calculate $\nabla \delta$ as

$$\nabla \delta = -\left(\nabla^2_{\delta} \mathcal{L}_{\text{CE}}(\widetilde{\mathcal{D}}_m, \theta)\right)^{-1} \nabla^2_{\delta,\theta} \mathcal{L}_{\text{CE}}(\widetilde{\mathcal{D}}_m, \theta). \tag{38}$$

This ensures that the response Jacobian captures the sensitivity of δ to changes in θ, while $\nabla_{\delta} \mathcal{L}_{\text{CE}}(\widetilde{\mathcal{D}}_m, \theta)$ captures the direct sensitivity of δ. These adjustments allow the gradient update for θ to incorporate the sensitivity of δ, resulting in a more stable and reliable solution for adversarial fine-tuning. The complexity of the gradient estimation method in Zeng et al. (2022) increases the likelihood of overfitting, as ∇_{δ} is dependent

on the estimation of $\nabla\delta$. Notably, $\nabla\delta$ requires estimation using second-order algorithms. While Zeng et al. (2022) asserts that these methods are robust to inaccuracies in the Hessian, the referenced literature assumes access to a large training dataset.

4.4.3 SAU

To enhance existing approaches, Wei et al. (2023) propose filtering candidate perturbations δ based on their ability to induce consistent misclassification across two classifiers. In this context, consistent misclassification means that both classifiers classifying $\tilde{x}$ as the same incorrect class $\tilde{y}$. Formally, Wei et al. (2023) optimize δ for each $(x, y) \in \mathcal{D}_m$ by solving the following optimization problem

$$\max_{\|\delta\|_p \leq \epsilon} \left\{ \frac{\lambda_1}{2} [\mathcal{L}_{\text{CE}}(\tilde{x}, y, \theta) + \mathcal{L}_{\text{CE}}(\tilde{x}, y, \bar{\theta})] - \lambda_2 \, \text{JS}(f(\tilde{x}, \theta), f(\tilde{x}, \bar{\theta})) \right\}, \tag{39}$$

where JS is the Jensen-Shannon divergence, and λ_1 and λ_2 are hyperparameters. Since the defender does not have access to two classifiers, the original model parameters $\bar{\theta}$ are used as the second classifier, $\bar{\theta}$ remaining fixed. Unlike PBE and NPD, SAU aims to distinguish between adversarial examples and backdoor triggers. To achieve this, they ensure that δ causes both θ and $\bar{\theta}$ to misclassify $\tilde{x}$ consistently, a behaviour more characteristic of backdoor triggers than typical adversarial examples (see (Wei et al., 2023, Fig. 2)).

Once the surrogate backdoor dataset $\widetilde{\mathcal{D}}_m = \{(\tilde{x}, x, \tilde{y}, y) \mid \tilde{y} \leftarrow f(\tilde{x}, \theta) \mid \tilde{x} = x + \delta \mid (x, y) \in \mathcal{D}_m\}$ is generated, Wei et al. (2023) solve the following optimization problem to fine-tuned the model

$$\min_{\theta} \left\{ \frac{1}{|\widetilde{\mathcal{D}}_m|} \sum_{(\tilde{x}, x, \tilde{y}, y) \in \widetilde{\mathcal{D}}_m} \lambda_3 \, \mathcal{L}_{\text{CE}}(x, y, \theta) - I(\tilde{y} \neq y) \log[1 - s_{\tilde{y}}(f(\tilde{x}, \theta))] \right\}, \tag{40}$$

where I is the indicator function that is 1 if $\tilde{x}$ is misclassified and λ_3 is another hyperparameter. This formulation balances the performance on $\mathcal{D}_m$, represented by the first term, with correcting the classification of $\tilde{x}$ to y, captured by the second term.

4.5 Summary

Fine-tuning, as an alternative to model pruning for backdoor mitigation, adjusts the parameters of a model rather than removing them. This strategy is typically governed by an objective function incorporating tailored regularization terms. Conventional fine-tuning approaches, such as FST and FT-SAM, attempt to adjust model weights to eliminate backdoors, though they often struggle with escaping local minima and thus fail to fully mitigate the backdoor threat. More advanced approaches based on KD, such as NAD and BCU, reframe fine-tuning as a process of filtering out harmful information. These approaches leverage the distillation of knowledge from a teacher model to a student model to effectively mitigate backdoor attacks while retaining the model's performance on clean data. Additionally, approaches such as MESA and BAERASER employ surrogate data generation to support the fine-tuning process. However, these approaches, rely on the defender having prior knowledge about the specific trigger used by an adversary. To remove this assumption, approaches such as PBE, i-BAU, and SAU modify existing adversarial training techniques. Similar to pruning approaches, each fine-tuning approach presents unique strengths and weaknesses, necessitating thorough evaluation across a diverse range of settings to fully understand their effectiveness.

5 Experimental Setup

In this section, we describe the experimental setup of our extensive evaluations covering 16 of the 18 approaches discussed in Sections 3 and 4. We benchmark each approach across a wide variety of settings as our evaluations span various backdoor attacks, model architectures, datasets, and poisoning ratios, resulting in a total of 288 distinct attack scenarios. Moreover, unlike Wu et al. (2022), we test each considered mitigation approach across three data availability settings, leading to 122,236 individual experiments in total. For our evaluations, we employ the *BackdoorBench* toolkit Wu et al. (2022), as it provided most of the required

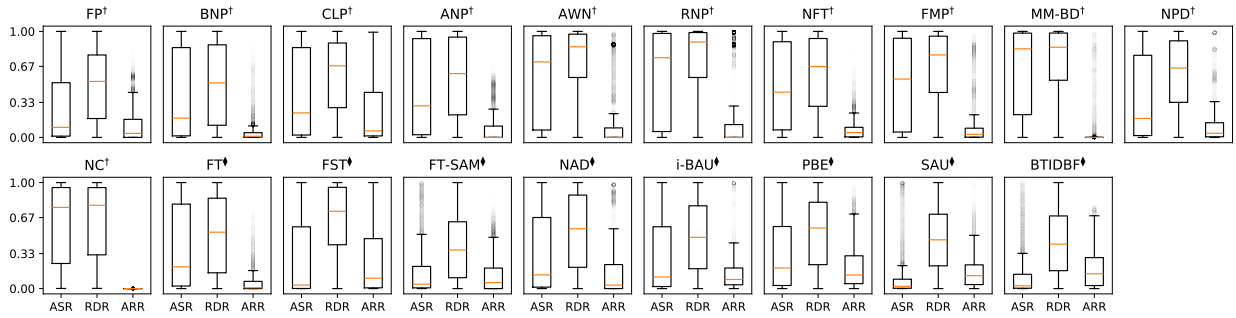


Figure 3: Box plots of the ASR, RDR, and ARR results for each approach across all considered settings. † = Pruning and ♦ = Fine-tuning. Note: NC results are only for CIFAR-10.

functionality. However, we have made several key modifications to this toolkit, such as incorporating the implementation of five additional mitigation approaches.

5.1 Attacks

Our evaluations include all backdoor attacks introduced in section 2.2, with the exception of LIRA. We exclude LIRA due to its poor performance during an initial set of experiments. Therefore, the attacks we consider are BadNets Gu et al. (2017), Blended Chen et al. (2017), Signal Barni et al. (2019), LF Zeng et al. (2021), SSBA Li et al. (2021c), IAB Nguyen & Tran (2020), BPP Wang et al. (2022), and WaNet Nguyen & Tran (2021). For each attack, we use the default configurations provided in *BackdoorBench*. We implement the attacks using poisoning ratios of 1%, 5%, and 10%, selected based on the findings reported in Wu et al. (2022).

5.2 Mitigation Methods

With the exception of MESA, BAERASER, BCU and UNIT, all approaches discussed in sections 3 and 4 are evaluated. Thus, we consider FP Liu et al. (2018) BNP Zheng et al. (2022a), CLP Zheng et al. (2022b), ANP Wu & Wang (2021), AWN Chai & Chen (2022), RNP Li et al. (2023), NFT Karim et al. (2024), FMP Huang & Bu (2023), MM-BD Wang et al. (2023), NPD Zhu et al. (2024b), FT Liu et al. (2018), FST Min et al. (2024), FT-SAM Zhu et al. (2023), NAD Li et al. (2021a), NC Wang et al. (2019a), BTI-DBF Xu et al. (2024a), PBE Mu et al. (2023), i-BAU Zeng et al. (2022), and SAU Wei et al. (2023). We evaluate NC only using CIFAR-10 due to its computational complexity scaling with the number of classes. In addition, exclude MESA Qiao et al. (2019) and BAERASER Liu et al. (2022) given their additional assumptions. Furthermore, we exclude BCU since it assumes access to an out-of-distribution dataset. While not a limitation, a fair evaluation of BCU would require benchmarking on multiple datasets, which is computationally prohibitive. Finally, due to the methodological similarities between MM-BD and UNIT, we only evaluate MM-BD.

For approaches already implemented in BackdoorBench, we use the default configurations. We implement AWN, MM-BD, RNP, NFT, FMP, BTIDBF, FST and PBE using the code provided by the authors, utilising the training configurations reported in each paper. In cases where hyperparameter values are not explicitly reported, we use the values from the respective codebase. In the Supplementary Materials (Table I), we summarise the hyperparameter values used for each approach.

5.3 Other Settings

We consider three datasets: CIFAR-10, German Traffic Sign Recognition Benchmark (GTSRB), and Tiny-ImageNet, containing 10, 43, and 200 classes, respectively. To evaluate the effect of data availability on the performance of each approach, we assess then under three data settings, based on the sample per class (SPC) value. Specifically, we evaluate each approach considering SPC values of 2, 10, and 100. For each data setting, we conduct 10 iterations, with each iteration utilizing a different random data partition. Moreover,

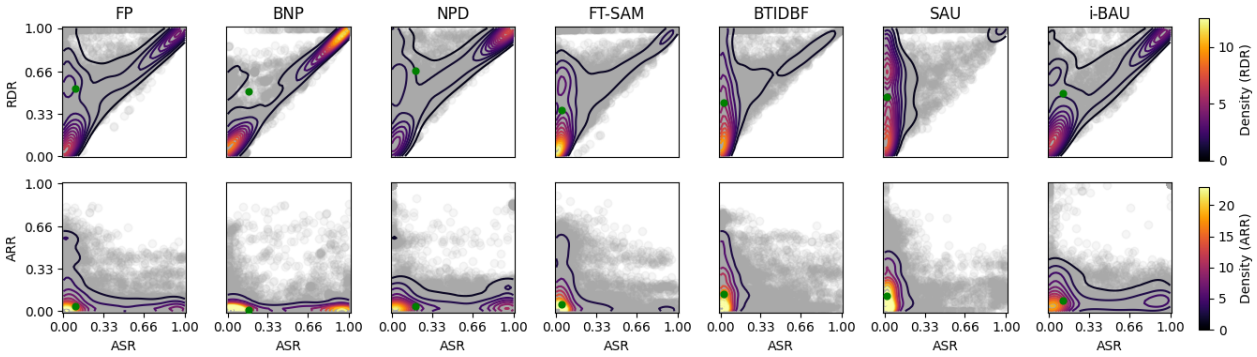


Figure 4: Scatter plots of the ASR, RDR, and ARR results for the best performing approach across all considered settings.

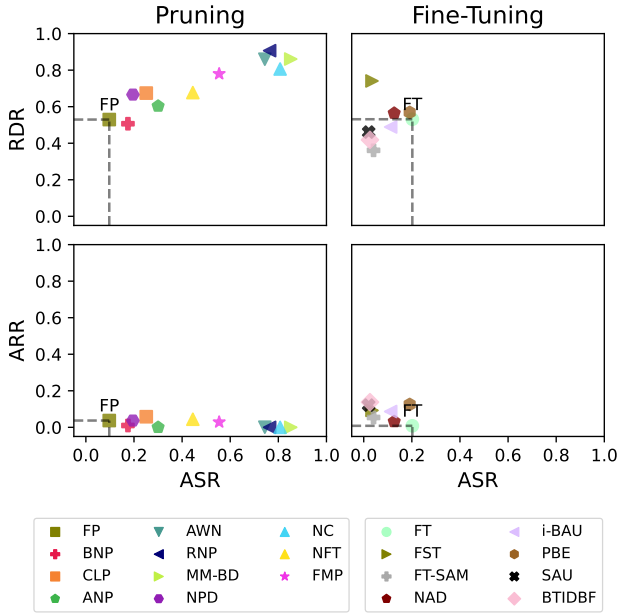


Figure 5: Scatter plot of the median RDR and ARR versus ASR for each approach across all considered settings. Note: NC results are only for CIFAR-10.

we employ four model architectures: PreAct-ResNet18 (ResNet), VGG-19 with batch normalisation (VGG), EfficientNet-B3 (EfficientNet), and MobileNetV3-Large (MobileNet), using their default configurations as provided by *BackdoorBench*.

5.4 Performance measures

To evaluate the effectiveness of backdoor mitigation, we use three key performance measures, commonly used in the literature: clean accuracy (ACC), attack success rate (ASR), and recovery accuracy (RA). These metrics, though sometimes referred to by different names in various works, serve the same purposes:

ACC: represents the accuracy of the original classification task. It is measured as the accuracy of the model on the testing data without the backdoor trigger applied (i.e., the clean data).

ASR: measures the effectiveness of the backdoor attack. It is calculated as the accuracy on testing data with the backdoor trigger applied (i.e., the backdoor data) and corresponding labels changed to the target label. Note that, testing data that originally belongs to the target class is omitted from this calculation.

RA: quantifies how effective a mitigation approach is at restoring the model's classification performance after backdoor mitigation. It measures the accuracy of testing data with the backdoor trigger applied but using the original (correct) labels.

In essence, ASR indicates whether the application of the trigger results in targeted misclassification, while RA shows whether the model can correctly classify backdoor samples after mitigation is applied, restoring them to their original labels. In our evaluations, we use normalised variants of ACC and RA, while ASR remains unchanged. These normalized metrics provide a clearer understanding of the model's performance before and after the application of backdoor mitigation approaches.

5.4.1 Accuracy Reduction Ratio (ARR)

To quantify the impact of mitigation on the accuracy of the original classification task, we take into account the accuracy values before and after mitigation, denoted as $\mathrm{ACC}_{\mathrm{pre}}$ and $\mathrm{ACC}_{\mathrm{post}}$, respectively. Therefore, we calculate the accuracy reduction ratio (ARR) as

$$\mathrm{ARR} = \frac{\mathrm{ACC}_{\mathrm{pre}} - \mathrm{ACC}_{\mathrm{post}}}{\mathrm{ACC}_{\mathrm{pre}}}. \tag{41}$$

Dividing the difference by the pre-mitigation accuracy accounts for variations in $\mathrm{ACC}_{\mathrm{pre}}$. For instance, most Tiny-ImageNet models exhibit lower $\mathrm{ACC}_{\mathrm{pre}}$ compared to their CIFAR-10 counterparts. Ideally, this ratio approaches 0, indicating minimal reduction in accuracy due to mitigation.

5.4.2 Recovery Difference Ratio (RDR)

To measure the effectiveness of a mitigation strategy in restoring the classification of backdoor samples to their original classes, we calculate the recovery difference ratio (RDR) as

$$\mathrm{RDR} = \frac{\mathrm{ACC}_{\mathrm{pre}} - \mathrm{RA}_{\mathrm{post}}}{\mathrm{ACC}_{\mathrm{pre}}}. \tag{42}$$

Similar to ARR, this ratio evaluates the difference between the post-mitigation RA and the pre-mitigation accuracy. In an optimal scenario, the post-mitigation RA, denoted by $\mathrm{RA}_{\mathrm{post}}$, ideally matches $\mathrm{ACC}_{\mathrm{pre}}$, while $\mathrm{ACC}_{\mathrm{post}}$ can often be impacted by the applied mitigation approach. As with ASR and ARR, a value of zero indicates an optimal outcome.

6 Evaluation Results

In this section, we present the results of our comprehensive evaluation of the considered backdoor attack mitigation approaches across various scenarios. We first discuss the overall performance of mitigation approaches across all experimental settings. Subsequently, we analyse the impact of data availability, backdoor attack type, model architecture, and dataset on the effectiveness of the examined mitigation approaches.

6.1 Overall Results

Figures 3 and 4 present an overview of the results for each considered approach. In Fig. 3, we use box plots to summarize the range of ASR, RDR, and ARR values, while, in Fig. 4, we plot the ARR and RDR performance of each proposal against the respective ASR. In Fig. 3, the top row and NC are pruning approaches, while the second row, excluding NC, are fine-tuning approaches. Moreover, in Fig. 5 we plot the median ARR and RDR performance of each proposal against the median ASR. Moreover, we draw a rectangle using the median ARR, RDR, and ASR results of FP and FT, with results that fall within it improving upon the performance of these baseline approaches. Note that the optimal performance in Fig. 5 corresponds to the bottom-left corner in all cases, and that FP and FT serve as the baseline pruning and fine-tuning approaches, respectively. Below, we discuss the pruning and fine-tuning approaches individually.

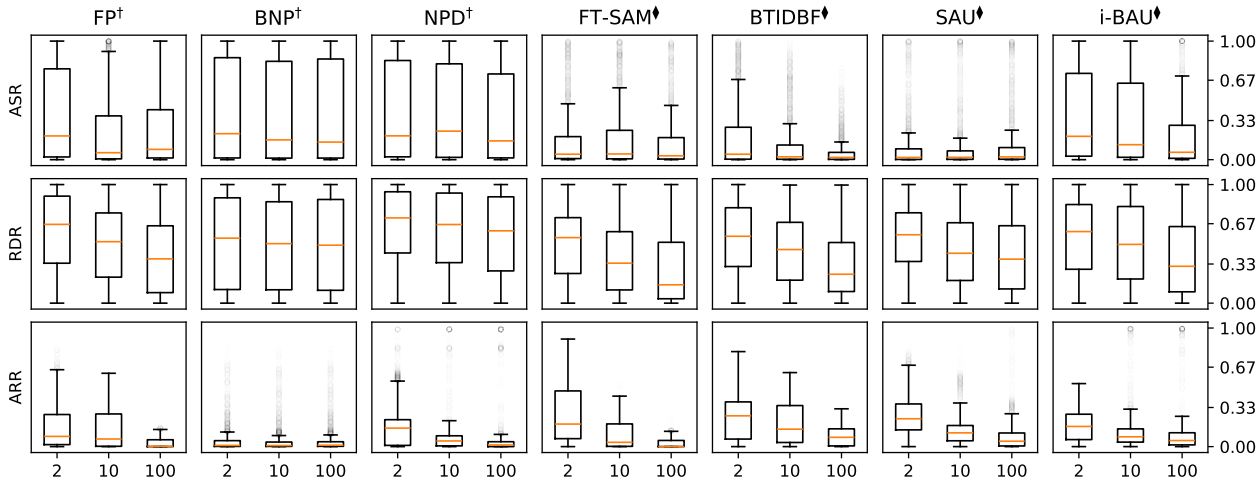


Figure 6: Box plots of the ASR, RDR, and ARR results for each approach and SPC values of 2, 10, and 100. † = Pruning and ♦ = Fine-tuning.

6.1.1 Pruning Methods

From Fig. 5, it is evident that none of the evaluated approaches appear in the bottom left rectangle, defined by the median ARR, RDR, and ASR results of FP. Thus, none of the evaluated pruning approaches improve the median performance of FP. However, it is worth noting that ANP, BNP, CLP, and NPD perform comparably to FP across all three performance measures.

For metric-based pruning approaches (i.e., FP, BNP, CLP and FMP), their overall effectiveness appears limited. Despite these approaches having low median ASR values, as shown in Fig. 4, they all exhibit a heavy tail in the distribution of ASR results. In terms of ARR, FP, FMP and BNP demonstrate good performance, with low medians and small variances. Although CLP has a median ARR comparable to FP, BNP and FMP, it displays significantly higher variance. In contrast, the ASR and RDR performance of NC and FMP is notably worse than FP. However, NC does not have the lowest median and variance for ARR among the evaluated approaches.

Masking-based pruning approaches (i.e., ANP, AWN, RNP and NFT) also demonstrate limited overall effectiveness. ANP's distribution of ASR, RDR, and ARR values is similar to FP, but with increased variation in ASR and RDR. While RNP and AWN perform well in terms of ARR, their high median ASR and RDR offsets this benefit. When comparing ANP with AWN and RNP, we find that, although AWN and RNP have been designed to improve upon ANP, in our evaluations, ANP consistently outperforms both. Finally, NFT finds itself as a middle ground between ANP and AWN/RNP in terms of ASR performance, while exhibiting comparable RDR median and variance to ANP.

Additive pruning methods (i.e., MM-BD and NPD) also show limited effectiveness. Similar to RNP, MM-BD has high median ASR and RDR values. Although NPD appears to outperform MM-BD overall, it fails to improve upon FP in any performance measure.

6.1.2 Fine-Tuning Methods

In contrast to model pruning approaches, we find that FT-SAM, SAU i-BAU and BTIDBF outperform the baseline fine-tuning approach, FT. These approaches fall within the RDR and ASR rectangle in Fig. 5 while also exhibiting a reduced ARR median with only a small ARR increase. On the other hand, FST, NAD and PBE do not surpass FT's performance, though NAD and, to some extent, PBE exhibit comparable performance to FT.

Among the conventional fine-tuning approaches (i.e., FST and FT-SAM), only FT-SAM outperforms FT, as indicated by its lower median ASR and RDR, as well as reduced variance, in Fig. 3. However, this

improvement comes at the expense of a higher median ARR and increased variance. While FST achieves improved ASR relative to FT, it underperforms in terms of RDR. Additionally, FST exhibits higher variance in ASR and ARR compared to FT. NAD demonstrates overall performance comparable to FT but with increased variance in ARR.

Adversarial training approaches (i.e., PBE, i-BAU, and SAU) exhibit varied performance. Compared to FT, i-BAU shows reduced median ASR and RDR performance with slightly increase ARR median and variance. PBE achieves similar ASR and RDR compared to FT, though it exhibits a higher median ARR and a longer tail. In contrast, SAU demonstrates significant improvement over FT, with a lower median ASR and reduced variance. However, similar to FT-SAM, this improvement comes with a trade-off in ARR performance.

Finally, BTIDBF exhibits similar performance to SAU, with a low median ASR and significantly reduced variance compared to FT. Moreover, its RDR median and variance are comparable to FT-SAM and SAU.

6.1.3 Density Analysis

When the density results in Fig. 4 are compared between pruning-based methods (FT, BNP, and NPD) and fine-tuning approaches (FT-SAM, BTIDBF, SAU, and i-BAU), we observe the following. (i) Pruning methods generally trade off RDR and ASR performance in a near 1:1 manner. For example, comparing BNP and SAU highlights that BNP's RDR performance is closely coupled with ASR, whereas SAU exhibits lower ASR variance but substantially higher RDR variance. (ii) This distinction is critical, as a defense achieving low ASR but high or unstable RDR does not meaningfully restore correct classification. In such cases, images may no longer be misclassified into the attacker's target class, but they remain misclassified into other incorrect classes. Thus, reducing ASR without simultaneously stabilizing RDR offers limited practical benefit.

6.1.4 Summary

With the exception of SAU, FT-SAM, i-BAU, and BTIDBF, all evaluated approaches exhibit variable performance across the full range of tested settings. Most methods show considerable variability in ASR and RDR, underscoring the need for caution when applying them in real-world scenarios. By contrast, SAU, FT-SAM, i-BAU, and BTIDBF achieve relatively low median ASR with reduced variance; however, this comes at the cost of lower ARR and persistently poor RDR performance. Furthermore, the overall improvement over FP and FT (introduced in 2018) is less substantial than claimed in much of the literature. Future research should therefore prioritize improving RDR, as it remains a key challenge. Importantly, while state-of-the-art solutions such as SAU, FT-SAM, i-BAU, and BTIDBF outperform traditional baselines, their inconsistent RDR performance limits practical applicability since images containing the trigger are still misclassified when RDR is high.

In the following subsections, we discuss BNP and NPD as they are the best-performing pruning approaches. Similarly, we discuss FT-SAM, BTIDBF, SAU, andi-BAU as the top-performing fine-tuning approaches.

6.2 Data Availability

To assess the effect of data availability, we evaluate each approach using 2, 10, and 100 samples per class (SPC). Fig. 6 show the SPC results of the best-performing approaches, as identified in the previous section. In Fig. 6, we summarize the range of ASR, RDR, and ARR values using box plots. Overall, we observe that data availability impacts ASR, RDR, and ARR performance of each work, as indicated by the increased median lines. However, pruning-based approaches tend to perform more consistently when SPC is reduced.

6.2.1 Model-Pruning

Data availability appears to have a minimal impact on the performance of most pruning methods. In particular, BNP and NPD demonstrate consistent ARR and RDR performance. In contrast, NPD and FP are more significantly affected by a reduction in SPC compared to BNP. While there are only minor differences in ASR and RDR, a noticeable increase in ARR median occurs as SPC decreases.

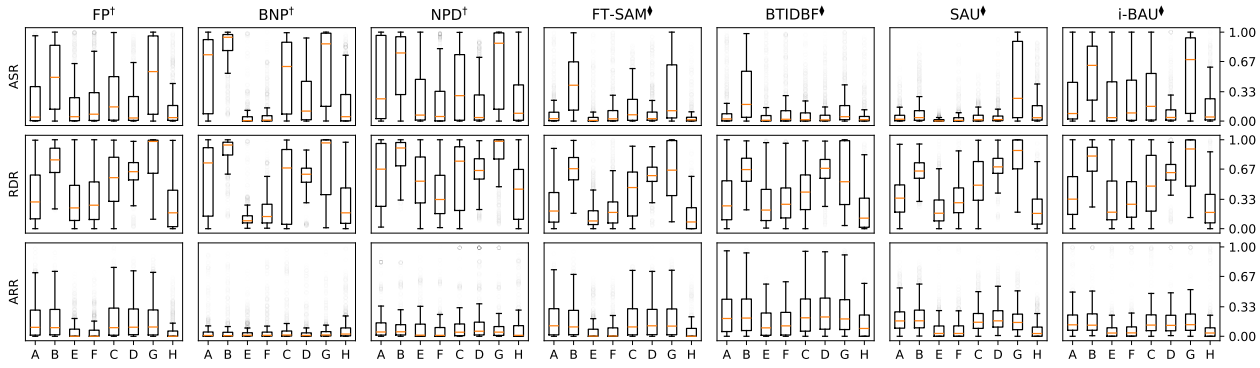


Figure 7: Box plots of the ASR, RDR, and ARR results for the selected approaches and different attack types. † = Pruning and ♦ = Fine-tuning. A = BadNet, B = Blended, C = LF, D = Signal, E = BPP, F = IAB, G = SSBA and H = WaNet.

6.2.2 Fine-Tuning

Unlike the evaluated pruning approaches, SPC has a greater impact on the performance of fine-tuning approaches. Specifically, median ARR and RDR increase significantly for FT-SAM and BTIDBF when SPC is reduced. Although a similar trend is observed for SAU, the impact of SPC is less pronounced.

6.3 Backdoor Attack

In Fig. 7, we present the results for the select set of approaches across the eight considered backdoor attacks. The performance of most approaches varies significantly across the range of tested attacks, particularly in terms of ASR and RDR. For example, although BNP performs well against the BPP attack, its performance against the Blended attack is the worst among all approaches. In contrast, FT-SAM, BTIDBF and SAU demonstrate more consistent performance across the attacks. Notable exceptions are SSBA for SAU, Blended for BTIDBF, as well as Blended and SSBA for FT-SAM. However, it is important to note that FT-SAM and SAU's RDR performance fluctuates similarly to those of other approaches.

Among the tested attacks, Blended and SSBA are the most difficult to defend against, as evidenced by higher median ASR median values and greater variance across most approaches. Additionally, the ASR and RDR results for the BadNets attack vary significantly for most approaches, with exception of FT-SAM and SAU. This is surprising given that BadNets is considered the foundational attack. Moreover, despite the attempts of recent approaches, including NPD and SAU to target dynamic backdoor attacks, mitigation performance is still varied, as demonstrated by poor SSBA performance.

6.4 Model Architecture

Fig. 8 shows the results for each selected approach across the four tested model architectures. Except for SAU, most approaches exhibit inconsistent performance across the considered architecture types, as indicated by fluctuations in ASR or ARR values. Notably, SAU demonstrates the most consistent performance across all architectures.

For FP, ARR variance increases significantly when using the EfficientNet architecture. Similarly, although BNP's ARR performance remains mostly stable, its median RDR and ASR show considerable variation. For NPD, median ASR noticeably increases when the MobileNet architecture is used.

While FT-SAM, BTIDBF and i-BAU maintain relatively consistent median ARR and ASR values across considered architectures, the tails of the distributions of these measures expand in certain cases. Moreover, there appears to be an inverse relationship between the tails of FT-SAM's ARR and ASR distributions. That is, a reduction in the size of the ARR distribution tail is often accompanied by an increase in the length and

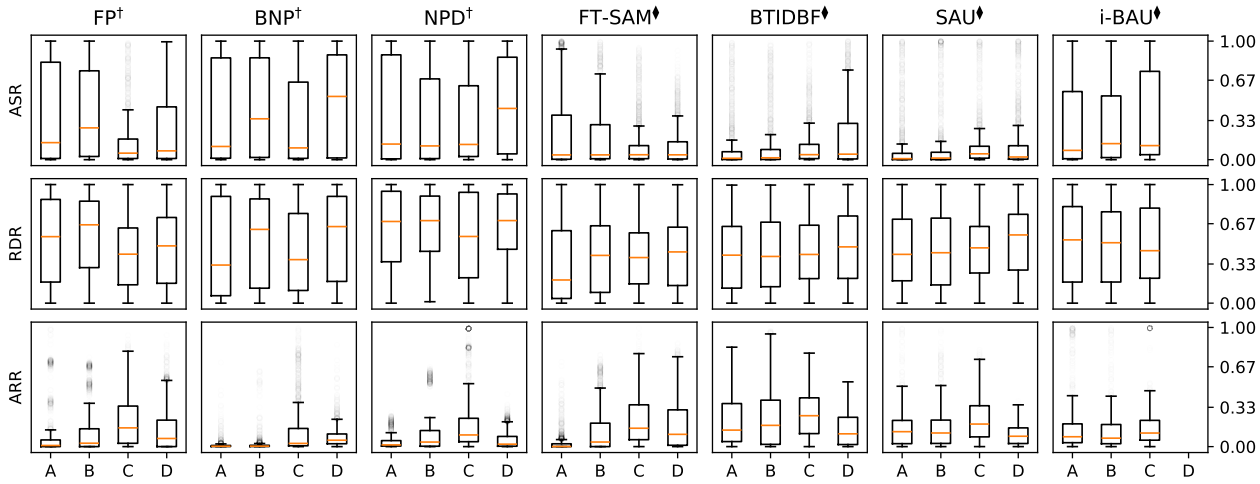


Figure 8: Box plots of the ASR, RDR and, ARR results for the selected approaches and different model architectures. † = Pruning and ♦ = Fine-tuning. A = VGG, B = ResNet, C = EfficientNet and D = MobileNet. Note: i-BAU results for MobileNet are not shown as its implementation is incompatible with this.

weight of the ASR distribution tail, and vice versa. In contrast, SAU exhibits only minor differences in all three performance measures across considered architectures.

6.5 Dataset

Fig. 9 shows the results for each selected approach across the three considered datasets. Similar to the performance variability observed with different model architectures, there is noticeable variability across datasets, often more pronounced.

Except for BNP, each approach exhibits variations in median or distribution tail of ARR across datasets. In particular, FP and NPD show significant variation in tail weight for the CIFAR-10 and Tiny-ImageNet. For RDR, increasing the complexity of the classification task generally leads to worse performance, with SAU being the exception. This is evident from the increase in median RDR from CIFAR-10 to GTSRB and from GTSRB to Tiny-ImageNet. A similar trend is present for ASR, where NPD, FT-SAM, SAU and i-BAU perform worse as task complexity increases.

7 Discussion

In this section, we discuss the major findings of our survey, connecting the literature reviewed in sections 3 and 4 with the evaluation results in section 6.

7.1 Sensitivity

Most of the evaluated approaches exhibit highly variable performance when tested across a broader set of scenarios compared to those originally considered in the respective papers. Although it is unrealistic to expect any particular approach to have invariant performance, the variance observed in most approaches is considerable. Specifically, ASR and RDR show significant variability in most cases. Among the evaluated variables, data availability, attack type, and dataset have the largest impact on performance.

For data availability, fine-tuning approaches are notably affected when SPC is reduced. Particularly, the performance of the original classification task, as quantified by ARR, deteriorates with reduced SPC. This is a significant observation, as a substantial decrease in ARR can render an approach impractical for real-world applications, irrespective of the removal of the backdoor. The inherent complexity of the optimization

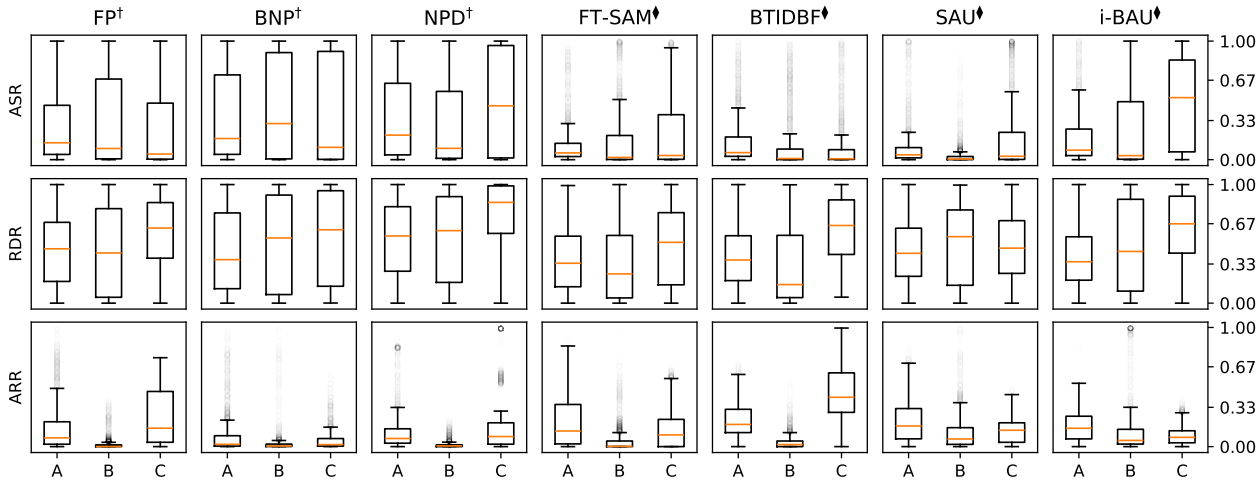


Figure 9: Box plots of the ASR, RDR, and ARR results for the selected approaches and different datasets. † = Pruning and ♦ = Fine-tuning. A = CIFAR-10, B = GTSRB and C = Tiny-ImageNet.

problems proposed by each fine-tuning approach often results in low bias but high variance, which increases the likelihood of overfitting to $\mathcal{D}_m$ when SPC is reduced. Although many approaches attempt to alleviate this issue by constraining the weight or input perturbation space or by defining a multi-objective optimization problem where one objective is to preserve the performance on $\mathcal{D}_m$, these approaches do not effectively constraint θ to account for overfitting.

Although FST constrains the solution space of θ, its minimisation of the inner product encourages differences among parameters. In contrast, while FT-SAM restricts the perturbation applied to θ, the outer minimisation imposes no direct constraint to θ. Given that a practical backdoor mitigation procedure typically operates under the assumption of limited access to clean data, it is crucial for future fine-tuning methods to develop effective strategies that constrain θ to mitigate the risk of overfitting. By achieving this, we posit that the mitigation task will become more tractable and better positioned to address the bias-variance trade-off while maintaining the performance of the original classification task.

Regarding attack type, dataset, and model architecture, most approaches show variable performance in removing and restoring the backdoor task, as measured by ASR and RDR. Unlike ARR, both ASR and RDR are inaccessible to the defender after mitigation, making large uncertainty in these measures a significant risk for practical use.

The sensitivity of the performance of most approaches to the values of their hyperparameters is a major factor contributing to this variability. While some approaches evaluate this sensitivity in different settings, the majority of evaluations are often limited in extent due to practical constraints. This underscores that although hyperparameters are often unavoidable, their inclusion requires careful consideration. Unlike traditional deep learning applications, using validation datasets to find optimal hyperparameter values is impractical for two main reasons. First, as mentioned earlier, ASR and RDR are not observable by the defender in practice, making hyperparameter tuning effective only for minimising ARR. Second, the use of a validation dataset further limits the available mitigation data, which can exacerbate the impact of data availability on overall performance.

Another factor contributing to performance variance across attack types, datasets, and model architectures is the reliance on limited observational evidence. Specifically, most backdoor mitigation strategies are built upon analyzing backdoored models and characterizing their behaviour. For instance, CLP investigates the correlation between UCLC and TAC, employing UCLC to prune filters with outlier values. However, our benchmarking results indicate that none of these observations can be deemed universal characteristics associated with backdoor attacks given the varied performance of most proposals. Rather, these observations are likely indicative of backdoor attacks within the specific settings tested by each approach. While we

recognise the practical implications of examining the range of settings evaluated in this work, our findings underscore the importance of considering variations in attack type, dataset, and model architecture.

Future mitigation approaches ought to carefully account for the impact of limited data availability on performance. Specifically, a more deliberate consideration of the bias-variance trade-off is essential when designing optimisation-based approaches. In addition, the inclusion of hyperparameters demands careful thought, as their values can significantly affect practical performance. Since hyperparameter optimization is often challenging or impractical in real-world scenarios, sensitivity to hyperparameter values becomes even more critical in such scenarios. Finally, future investigations into the underlying mechanisms that drive backdoor attacks (assuming such mechanisms exist) need to carefully account for variations in attack type, dataset, and model architecture.

7.2 Recovery Accuracy

Among the surveyed works, we find that only the proposers of SAU evaluate the ability of their mitigation approach to restore the classification of the backdoor task, quantified as RDR in this paper. While ASR measures how effectively the backdoor induces the misclassification of samples containing the trigger to the target class, BackdoorBench Wu et al. (2022) highlights that this measure alone does not determine the overall effectiveness of a mitigation approach. Specifically, minimizing ASR without a corresponding increase in RA, the performance measure used to calculate RDR, is not indicative of optimal performance. If ASR is minimised but RDR remains high, the model is still unable to accurately classify samples containing the trigger. Although this may not align with the adversary's original objective, it still has significant implications for models deployed in real-world settings.

Across the tested settings, RDR varies significantly. Even though FT-SAM, i-BAU, BTIDBF and SAU are state-of-the-art approaches, their RDR performance still exhibit notable variability. While RDR is dependent on ASR and ARR, as $\text{ASR} + \text{RA} \leq 1$ and $\text{RA} \lessapprox \text{ACC}$, the median and variance of RDR often exceed the values expected from the relationship between these performance measures.

Since samples containing the trigger are inaccessible to the defender, restoring the classification of the backdoor task presents a major challenge. Despite efforts by the proposers of many approaches to model the trigger distribution, a substantial improvement in RDR has not been observed. Therefore, moving forward, more focused exploration of alternative methodologies targeting RDR is necessary.

7.3 Trigger Modeling

Modelling the trigger distribution is a widely adopted technique, used in nearly half of the surveyed works. Approaches that utilize this technique rely on the insights derived from the trigger model to mitigate corresponding backdoor attacks. A key feature shared among these approaches is the use of a constrained optimisation to determine δ, though the specific implementation details differ across approaches. This constrained optimisation requires selecting a norm and an upperbound (ϵ) for the norm of δ. This inherently involves certain assumptions about ρ, the actual trigger employed by the adversary.

Among the discussed approaches, NC, MESA, and BAERASER constrain the ℓ_1 norm of δ, while NPD, AWN, PBE, i-BAU, BTIDBF, and SAU utilize the ℓ_2 norm. In the evaluated attacks, triggers employed by the BadNets and, arguably, IAB are sparse in nature, whereas other attacks apply smoother, less perceptible triggers. Interestingly, our results suggest that using an ℓ_1 norm constraint in trigger modelling does not significantly enhance mitigation performance against attacks with sparse triggers compared to using an ℓ_2 norm constraint. Furthermore, we observe that constraining the ℓ_2 norm does not guarantee successful mitigation against attacks with smoother triggers, as illustrated by the results for the Blended and SSBA attacks in section 6.3.

One of the main challenges with using the ℓ_2 norm is the natural occurrence of adversarial examples within the input space. To tackle this, AWN and i-BAU model δ as a global input perturbation. However, the results for AWN and i-BAU indicate that this additional constraint leads to suboptimal performance. In contrast, others adopt a sample-specific approach to modelling the trigger distribution. NPD assigns a unique δ for each x based on its second-largest logit, ensuring that the perturbation δ causes targeted

misclassification. However, our findings in section 6.3 reveal that this selection method yields variable performance. Conversely, PBE employs an existing untargeted adversarial example generation method, demonstrating that untargeted adversarial examples generated using the PGD attack tend to exhibit biased classification towards the adversary's target class. However, our findings suggest that the error rate of the PGD attack (i.e., the proportion of adversarial examples not classified as the target class) impacts PBE's performance, particularly in scenarios with limited data availability.

To account for the presence of adversarial examples in sample-specific trigger modelling with ℓ_2 norm constraint, SAU filters candidate triggers using the original model parameters. Specifically, SAU identifies sample-specific perturbations δ that induce consistent misclassification given the original model parameters θ and the modified ones $\bar{\theta}$. Our results show that this strategy more effectively distinguishes between adversarial examples and candidate triggers, as evidenced by robust ASR performance across the tested attacks. However, it is important to note that this approach alone is insufficient for restoring the correct classification of backdoor samples, as discussed earlier (see section 7.2).

8 Conclusion

We critically evaluated various state-of-the-art backdoor attack mitigation strategies within the context of image recognition. Our analysis, spanning a broad spectrum of attacks, datasets, model architectures, data availabilities, and poisoning ratios uncovers several key insights into the effectiveness and limitations of current approaches:

- While many approaches demonstrate strong performance in specific settings, most exhibit significant variability, particularly when faced with diverse attack types and constrained data availability. Pruning-based approaches such as BNP, ANP, and CLP offer a degree of robustness but struggle with performance consistency. Fine-tuning approaches, notably FT-SAM and SAU, show promise by outperforming their respective baselines, though they come with trade-offs, especially in terms of accuracy reduction and recovery performance.
- The widespread reliance on hyperparameters and constrained optimization techniques introduces significant challenges, particularly in real-world deployment. While hyperparameters are crucial to the success of many approaches, they must be carefully tuned to avoid overfitting and ensure robustness across varied attack scenarios. Balancing the bias-variance trade-off in optimization-based approaches is critical for future improvements.
- Trigger modeling remains a key technique in mitigating backdoor attacks. However, our results suggest that common assumptions about the trigger distribution do not universally hold across all attack scenarios. This highlights the need for more adaptive approaches that can account for variations in attack types and the properties of input data.
- A major challenge across all evaluated approaches is the restoration of backdoor-affected classifications, as quantified by RA and RDR. Most approaches, despite reducing attack success rates, fail to fully recover the correct classification of backdoor samples, underscoring the need for future research to focus on improving RDR.

In summary, while considerable progress has been made in developing backdoor mitigation strategies, our findings highlight the need for more adaptive, robust, and generalizable solutions. Future research can focus on addressing trade-offs between accuracy, recovery, and computational efficiency, while also exploring new approaches to mitigate backdoor attacks in diverse real-world scenarios.

References

Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 101–105. IEEE, 2019.

Shuwen Chai and Jinghui Chen. One-shot neural backdoor erasing via adversarial weight masking. *Advances in Neural Information Processing Systems*, 35:22285–22299, 2022.

Shih-Han Chan, Yinpeng Dong, Jun Zhu, Xiaolu Zhang, and Jun Zhou. Baddet: Backdoor attacks on object detection. In *European Conference on Computer Vision*, pp. 396–412. Springer, 2022.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Pengzhou Cheng, Zongru Wu, Wei Du, and Gongshen Liu. Backdoor attacks and countermeasures in natural language processing models: A comprehensive security review. *arXiv preprint arXiv:2309.06055*, 2023.

Siyuan Cheng, Guangyu Shen, Kaiyuan Zhang, Guanhong Tao, Shengwei An, Hanxi Guo, Shiqing Ma, and Xiangyu Zhang. Unit: Backdoor mitigation via automated neural distribution tightening. In *European Conference on Computer Vision*, pp. 262–281. Springer, 2024.

Khoa Doan, Yingjie Lao, Weijie Zhao, and Ping Li. Lira: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11966–11976, 2021.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference*, pp. 113–125, 2019.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

Kathrin Grosse, Lukas Bieringer, Tarek Richard Besold, and Alexandre Alahi. Towards more practical threat models in artificial intelligence security. *arXiv preprint arXiv:2311.09994*, 2023.

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

Yang He and Lingao Xiao. Structured pruning for deep convolutional neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2023.

Dong Huang and Qingwen Bu. Adversarial feature map pruning for backdoor. *arXiv preprint arXiv:2307.11565*, 2023.

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*, pp. 401–418. Springer, 2024.

Quentin Le Roux, Eric Bourbao, Yannick Teglia, and Kassem Kallas. A comprehensive survey on backdoor attacks and their defenses in face recognition systems. *IEEE Access*, 2024.

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*, 2021a.

Yige Li, Xixiang Lyu, Xingjun Ma, Nodens Koren, Lingjuan Lyu, Bo Li, and Yu-Gang Jiang. Reconstructive neuron pruning for backdoor defense. In *International Conference on Machine Learning*, pp. 19837–19854. PMLR, 2023.

Yiming Li, Yanjie Li, Yalei Lv, Yong Jiang, and Shu-Tao Xia. Hidden backdoor attack against semantic segmentation models. *arXiv preprint arXiv:2103.04038*, 2021b.

Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16463–16472, 2021c.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, pp. 273–294. Springer, 2018.

Yang Liu, Mingyuan Fan, Cen Chen, Ximeng Liu, Zhuo Ma, Li Wang, and Jianfeng Ma. Backdoor defense with machine unlearning. In *IEEE INFOCOM 2022-IEEE conference on computer communications*, pp. 280–289. IEEE, 2022.

Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1265–1282, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Rui Min, Zeyu Qin, Li Shen, and Minhao Cheng. Towards stable backdoor purification through feature shift tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

Bingxu Mu, Zhenxing Niu, Le Wang, Xue Wang, Qiguang Miao, Rong Jin, and Gang Hua. Progressive backdoor erasing via connecting backdoor and adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20495–20503, 2023.

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

Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *Advances in Neural Information Processing Systems*, 33:3454–3464, 2020.

Lu Pang, Tao Sun, Haibin Ling, and Chao Chen. Backdoor cleansing with unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12218–12227, 2023.

Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):1–36, 2018.

Ximing Qiao, Yukun Yang, and Hai Li. Defending neural backdoors via generative distribution modeling. *Advances in neural information processing systems*, 32, 2019.

Xuan Sheng, Zhaoyang Han, Piji Li, and Xiangmao Chang. A survey on backdoor attack and defense in natural language processing. In *2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS)*, pp. 809–820. IEEE, 2022.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Yichen Wan, Youyang Qu, Wei Ni, Yong Xiang, Longxiang Gao, and Ekram Hossain. Data and model poisoning backdoor attacks on wireless federated learning, and the defense mechanisms: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2024.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE symposium on security and privacy (SP)*, pp. 707–723. IEEE, 2019a.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 707–723. IEEE, 2019b.

Hang Wang, Zhen Xiang, David J Miller, and George Kesidis. Mm-bd: Post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 15–15. IEEE Computer Society, 2023.

Xianmin Wang, Jing Li, Xiaohui Kuang, Yu-an Tan, and Jin Li. The security of machine learning in an adversarial setting: A survey. *Journal of Parallel and Distributed Computing*, 130:12–23, 2019c.

Zhenting Wang, Juan Zhai, and Shiqing Ma. Bppattack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15074–15084, 2022.

Shaokui Wei, Mingda Zhang, Hongyuan Zha, and Baoyuan Wu. Shared adversarial unlearning: Backdoor mitigation by unlearning shared adversarial examples. *Advances in Neural Information Processing Systems*, 36:25876–25909, 2023.

Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoorbench: A comprehensive benchmark of backdoor learning. *Advances in Neural Information Processing Systems*, 35:10546–10559, 2022.

Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. *Advances in Neural Information Processing Systems*, 34:16913–16925, 2021.

Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, and Kui Ren. Towards reliable and efficient backdoor trigger inversion via decoupling benign features. In *The Twelfth International Conference on Learning Representations*, 2024a.

Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, and Kui Ren. Towards reliable and efficient backdoor trigger inversion via decoupling benign features. In *The Twelfth International Conference on Learning Representations*, 2024b.

Baochen Yan, Jiahe Lan, and Zheng Yan. Backdoor attacks against voice recognition systems: A survey. *arXiv preprint arXiv:2307.13643*, 2023.

Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, 2019.

Yi Zeng, Won Park, Z Morley Mao, and Ruoxi Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16473–16481, 2021.

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.

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

Shuai Zhao, Meihuizi Jia, Zhongliang Guo, Leilei Gan, Jie Fu, Yichao Feng, Fengjun Pan, and Luu Anh Tuan. A survey of backdoor attacks and defenses on large language models: Implications for security measures. *arXiv preprint arXiv:2406.06852*, 2024.

Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Pre-activation distributions expose backdoor neurons. *Advances in Neural Information Processing Systems*, 35:18667–18680, 2022a.

Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Data-free backdoor removal based on channel lipschitzness. In *European Conference on Computer Vision*, pp. 175–191. Springer, 2022b.

Mingli Zhu, Shaokui Wei, Li Shen, Yanbo Fan, and Baoyuan Wu. Enhancing fine-tuning based backdoor defense with sharpness-aware minimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4466–4477, 2023.

Mingli Zhu, Siyuan Liang, and Baoyuan Wu. Breaking the false sense of security in backdoor defense through re-activation attack. *Advances in Neural Information Processing Systems*, 37:114928–114964, 2024a.

Mingli Zhu, Shaokui Wei, Hongyuan Zha, and Baoyuan Wu. Neural polarizer: A lightweight and effective backdoor defense via purifying poisoned features. *Advances in Neural Information Processing Systems*, 36, 2024b.

A Appendix

B Experimental Parameters

Table 5: Experimental parameters used by each approach. AR = Accuracy Ratio, UT = Unlearn Threshold, RD = Recovery Drop

Reference	Approach	Implementation	Training Parameters			Hyperparameters		
			CIFAR-10	GTSRB	Tiny	CIFAR-10	GTSRB	Tiny
Liu et al. (2018)	FP	BackdoorBench	AR = 0.1	AR = 0.1	AR = 0.1	N/A	N/A	N/A
Zheng et al. (2022a)	BNP	BackdoorBench	N/A	N/A	N/A	$\lambda = 3$	$\lambda = 3$	$\lambda = 3$
Zheng et al. (2022b)	CLP	BackdoorBench	N/A	N/A	N/A	$\lambda = 3$	$\lambda = 3$	$\lambda = 3$
Wu & Wang (2021)	ANP	BackdoorBench	AR = 0.1	AR = 0.1	AR = 0.1	$\epsilon = 0.4$ $\lambda = 0.2$	$\epsilon = 0.4$ $\lambda = 0.2$	$\epsilon = 0.4$ $\lambda = 0.2$
Chai & Chen (2022)	AWN	GitHub	N/A	N/A	N/A	$\lambda_1 = 0.9$ $\lambda_2 = 0.1$ $\lambda_3 = 10^{-7}$	$\lambda_1 = 0.9$ $\lambda_2 = 0.1$ $\lambda_3 = 10^{-7}$	$\lambda_1 = 0.9$ $\lambda_2 = 0.1$ $\lambda_3 = 10^{-7}$
Li et al. (2023)	RNP	GitHub	UT = 0.1 RD = 0.02	UT = 0.1 RD = 0.02	UT = 0.1 RD = 0.02	N/A	N/A	N/A
Karim et al. (2024)	NFT	GitHub	$\beta = 0.5$ $\alpha = 0.8$	$\beta = 0.5$ $\alpha = 0.8$	$\beta = 0.5$ $\alpha = 0.8$	$\lambda = 0.001$	$\lambda = 0.001$	$\lambda = 0.001$
Huang & Bu (2023) N/A	FMP	GitHub		N/A	N/A	N/A	N/A	N/A
Wang et al. (2023)	MM-BD	GitHub	AR = 0.05 $\alpha = 1.2$	AR = 0.05 $\alpha = 1.2$	AR = 0.05 $\alpha = 1.2$	$\lambda = 0.5$	$\lambda = 0.5$	$\lambda = 0.5$
Zhu et al. (2024b)	NPD	BackdoorBench	N/A	N/A	N/A	$\lambda_1 = 1$ $\lambda_2 = 0.4$ $\lambda_3 = 0.4$	$\lambda_1 = 1$ $\lambda_2 = 0.5$ $\lambda_3 = 0.5$	$\lambda_1 = 1$ $\lambda_2 = 0.4$ $\lambda_3 = 0.4$
Wang et al. (2019a)	NC	BackdoorBench	N/A	N/A	N/A	$\lambda = 10^{-3}$	$\lambda = 10^{-3}$	$\lambda = 10^{-3}$
Liu et al. (2018)	FT	BackdoorBench	N/A	N/A	N/A	N/A	N/A	N/A
Min et al. (2024)	FST	GitHub	N/A	N/A	N/A	$\lambda = 0.2$	$\lambda = 0.01$	$\lambda = 0.001$
Zhu et al. (2023)	FT-SAM	BackdoorBench	N/A	N/A	N/A	N/A	N/A	N/A
Li et al. (2021a)	NAD	BackdoorBench	N/A	N/A	N/A	$\lambda \in \{500, 1000\}$	$\lambda \in \{500, 1000\}$	$\lambda \in \{500, 1000\}$
Mu et al. (2023)	PBE	GitHub	N/A	N/A	N/A	N/A	N/A	N/A
Zeng et al. (2022)	i-BAU	BackdoorBench	N/A	N/A	N/A	N/A	N/A	N/A
Wei et al. (2023)	SAU	BackdoorBench	N/A	N/A	N/A	$\lambda_1 = 1$ $\lambda_2 = 0$ $\lambda_1 = 1$	$\lambda_1 = 1$ $\lambda_2 = 0$ $\lambda_1 = 1$	$\lambda_1 = 1$ $\lambda_2 = 0$ $\lambda_1 = 1$
Wei et al. (2023)	BTI-DBF	GitHub	N/A	N/A	N/A	N/A	N/A	N/A

C Complete Results

C.1 Data Availability

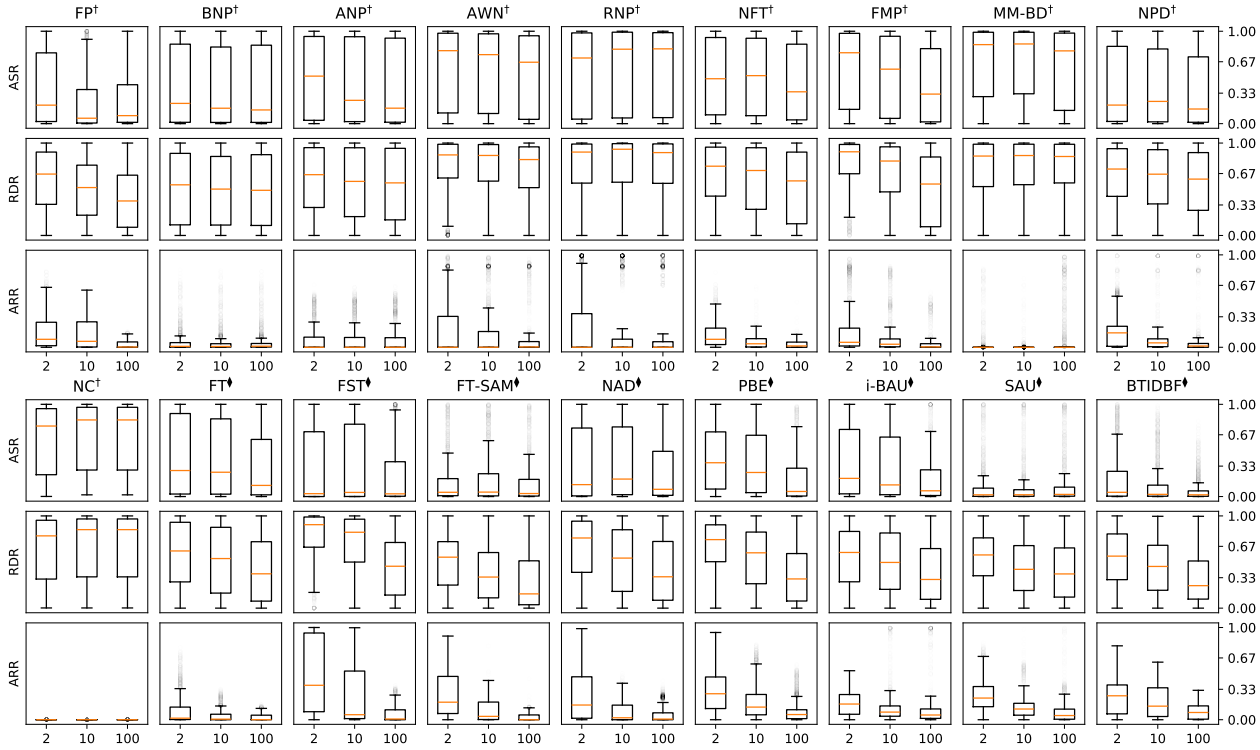


Figure 10: Box plots of the ASR, RDR, and ARR results for each approach across all considered scenarios. † = Pruning and ♢ = Fine-tuning. Note: NC results are only for CIFAR-10.

C.2 Backdoor Attack

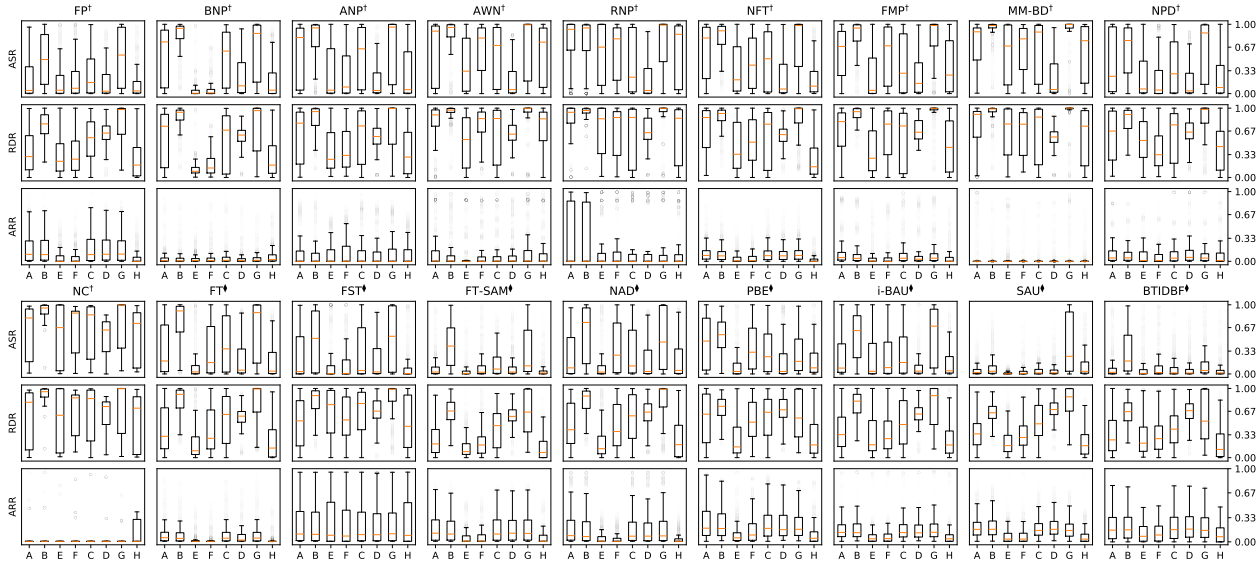


Figure 11: Box plots of the ASR, RDR, and ARR results for each approach and different attack types. † = Pruning and ♢ = Fine-tuning. A = BadNet, B = Blended, C = LF, D = Signal, E = BPP, F = IAB, G = SSBA and H = WaNet. Note: NC results are only for CIFAR-10.

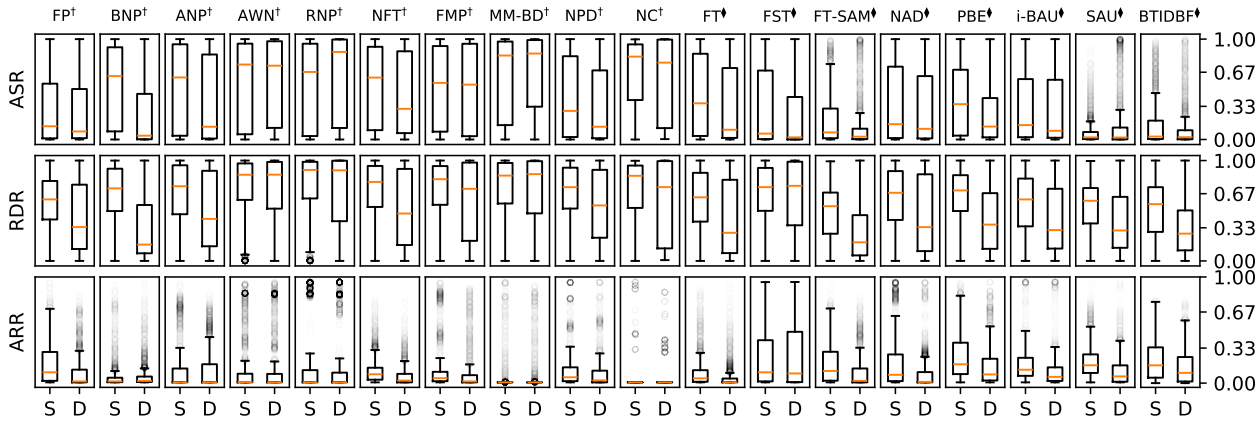


Figure 12: Box plots of the ASR, RDR, and ARR results for each approach and both static and dynamic attacks. † = Pruning and ♢ = Fine-tuning. S = Static and D = Dynamic. Note: NC results are only for CIFAR-10.

C.3 Model Architecture

Figure 13: Box plots of the ASR, RDR, and ARR results for each approach and different model architectures. † = Pruning and ♢ = Fine-tuning. A = VGG, B = ResNet, C = EfficientNet and D = MobileNet. Note: The current i-BAU implementation is incompatible with the MobileNet architecture and NC results are only for CIFAR-10.

C.4 Dataset

Figure 14: Box plots of the ASR, RDR, and ARR results for the selected approaches and different datasets. † = Pruning and ♢ = Fine-tuning. A = CIFAR-10, B = GTSRB and C = Tiny-ImageNet. Note: NC results are only for CIFAR-10.

C.5 Poisoning Ratio

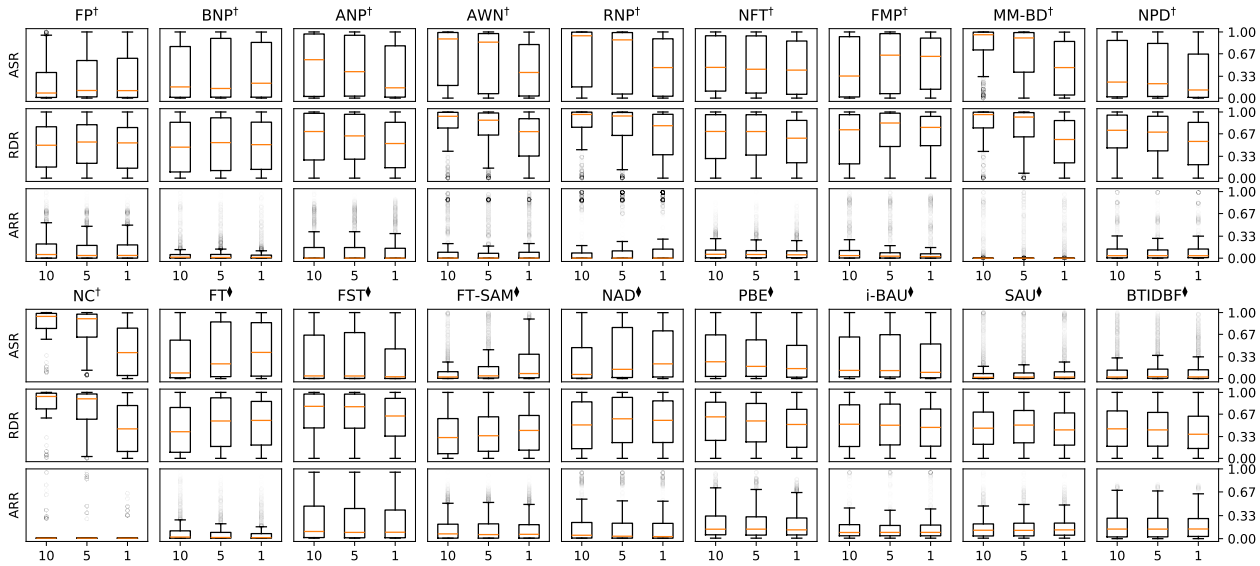


Figure 15: Box plots of the ASR, RDR, and ARR results for the selected approaches and different poisoning ratios (%). † = Pruning and ♢ = Fine-tuning. Note: NC results are only for CIFAR-10.