Uncovering, Explaining, and Mitigating the Superficial Safety of Backdoor Defense

Rui Min[1]*, Zeyu Qin[1]*, Nevin L. Zhang[1], Li Shen, Minhao Cheng[2]
[1]Hong Kong University of Science and Technology, [2]Pennsylvania State University
{rminaa, zeyu.qin}@connect.ust.hk, lzhang@cse.ust.hk
mathshenli@gmail.com, minhaocheng@ust.hk

Abstract

Backdoor attacks pose a significant threat to Deep Neural Networks (DNNs) as they allow attackers to manipulate model predictions with backdoor triggers. To address these security vulnerabilities, various backdoor purification methods have been proposed to purify compromised models. Typically, these purified models exhibit low Attack Success Rates (ASR), rendering them resistant to backdoored inputs. However, *Does achieving a low ASR through current safety purification methods truly eliminate learned backdoor features from the pretraining phase?* In this paper, we provide an affirmative answer to this question by thoroughly investigating the *Post-Purification Robustness* of current backdoor purification methods. We find that current safety purification methods are vulnerable to the rapid re-learning of backdoor behavior, even when further fine-tuning of purified models is performed using a very small number of poisoned samples. Based on this, we further propose the practical Query-based Reactivation Attack (QRA) which could effectively reactivate the backdoor by merely querying purified models. We find the failure to achieve satisfactory post-purification robustness stems from the insufficient deviation of purified models from the backdoored model along the backdoor-connected path. To improve the post-purification robustness, we propose a straightforward tuning defense, Path-Aware Minimization (PAM), which promotes deviation along backdoor-connected paths with extra model updates. Extensive experiments demonstrate that PAM significantly improves post-purification robustness while maintaining a good clean accuracy and low ASR. Our work provides a new perspective on understanding the effectiveness of backdoor safety tuning and highlights the importance of faithfully assessing the model's safety.

1 Introduction

Backdoor attacks [2, 7, 16] have emerged as one of the most significant concerns [4, 5, 20, 25] in deep learning. These attacks involve the insertion of malicious backdoor triggers into the training set, which can be further exploited to manipulate the behavior of the model during the inference stage. To defend against these threats, researchers have proposed various safety tuning methods [20, 27, 32, 44, 49, 50, 55] to purify well-trained backdoored models. These methods can be easily incorporated into the existing model deployment pipeline and have demonstrated state-of-the-art effectiveness in reducing the Attack Success Rate (ASR) of backdoored models [32, 48].

However, a critical question arises: *does achieving a low Attack Success Rate (ASR) through current safety tuning methods genuinely indicate the complete removal of learned backdoor features from the pretraining phase?* If the answer is no, this means that the adversary may still easily reactivate the

*Equal contribution. Correspondence to: Zeyu Qin (zeyu.qin@connect.ust.hk), Li Shen, Minhao Cheng.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

implanted backdoor from the residual backdoor features lurking within the purified model, thereby exerting insidious control over the model's behavior. This represents a significant and previously unacknowledged safety concern, suggesting that current defense methods may only offer *superficial safety* [1]. Moreover, if an adversary can successfully re-trigger the backdoor, it raises another troubling question: how can we assess the model's robustness against such threats? This situation underscores the urgent need for a more comprehensive and faithful evaluation of the model's safety.

In this work, we provide an affirmative answer to these questions by thoroughly investigating the **Post-Purification Robustness** of state-of-the-art backdoor safety tuning methods. Specifically, we employ the *Retuning Attack* (RA) [36, 42] where we first retune the purified models using an extremely small number of backdoored samples and tuning epochs. Our observations reveal that current safety purification defense methods quickly reacquire backdoor behavior after just a few epochs, resulting in significantly high ASR levels. In contrast, the clean model (which does not have backdoor triggers inserted during the pretraining phase) and *Exact Purification* (EP)—which fine-tunes models using real backdoored samples with correct labels during safety purification, maintain a low ASR even after the RA. This discrepancy suggests that existing safety tuning methods do not thoroughly eliminate the learned backdoor, creating a *superficial impression of backdoor safety*. Since the vulnerability revealed by the Retuning Attack (RA) relies on the use of retuned models, we further propose the more practical Query-based Reactivation Attack (QRA). This attack is capable of generating sample-specific perturbations that can trigger the backdoor in purified models, which were previously believed to have eliminated such threats, simply by querying these purified models.

To understand the inherent vulnerability of current safety purification methods concerning post-purification robustness, we further investigate the factors contributing to the disparity in post-purification robustness between EP and other methods. To this end, we utilize Linear Mode Connectivity (LMC) [13, 33] as a framework for analysis. We find that *EP not only produces a solution with low ASR like other purification methods but also pushes the purified model further away from the backdoored model along the backdoor-connected path, resulting in a more distantly robust solution.* As a result, it becomes challenging for the retuning attack to revert the EP model back to the basin with high ASR where the compromised model is located. Inspired by our findings, we propose a simple tuning defense method called Path-Aware Minimization (PAM) to enhance post-purification robustness. By using reversed backdoored samples as a proxy to measure the backdoored-connected path, PAM updates the purified model by applying gradients from a model interpolated between the purified and backdoored models. This approach helps identify a robust solution that further deviates our purified model from the backdoored model along the backdoor-connected path. Extensive experiments have demonstrated that PAM achieves improved post-purification robustness, retaining a low ASR after RA across various settings. To summarize, our contributions are:

- Our work first offers a new perspective on understanding the effectiveness of current backdoor safety tuning methods. Instead of merely focusing on the commonly used Attack Success Rate, we investigate the Post-Purification Robustness of the purified model to enhance our comprehensive understanding of backdoor safety in deep learning models.
- We employ the Retuning Attack by retuning purified models on backdoored samples to assess the post-purification robustness. Our primary observations reveal that current safety purification methods are vulnerable to RA, as evidenced by a rapid increase in the ASR. Furthermore, we propose the more practical Query-based Reactivation Attack, which can reactivate the implanted backdoor of purified models solely through model querying.
- We analyze the inherent vulnerability of current safety purification methods to the RA through Linear Mode Connectivity and attribute the reason to the insufficient deviation of purified models from the backdoored model along the backdoor-connected path. Based on our analysis, we propose Path-Aware Minimization, a straightforward tuning-based defense mechanism that promotes deviation by performing extra model updates using interpolated models along the path. Extensive experiments verify the effectiveness of the PAM method.

2 Related Work

Backdoor Attacks. Backdoor attacks aim to manipulate the backdoored model to predict the target label on samples containing a specific backdoor trigger while behaving normally on benign samples. They can be roughly divided into two categories [48]: **(1)** Data-poisoning attacks: the attacker inserts a backdoor trigger into the model by manipulating the training sample $(\boldsymbol{x}, \boldsymbol{y}) \in (\mathcal{X}, \mathcal{Y})$, such as adding a small patch to a clean image $\boldsymbol{x}$ and change the sample's label to an attacker-designated target label $\boldsymbol{y}_t$ [4, 7, 15, 16, 28, 43]; **(2)** Training-control attacks: the attacker has control over both the training process and the training data simultaneously [34]. Note that data-poisoning attacks are more practical in real-world scenarios as they make fewer assumptions about the attacker's capabilities [5, 15, 41] and have resulted in increasingly serious security risks [4, 39]. In this work, *we focus on data-poisoning backdoor attacks*.

Backdoor Defense. Existing backdoor defense strategies could be broadly categorized into robust pretraining [19, 26] and robust fine-tuning methods [32, 49, 53, 55]. Robust pretraining aims to prevent the learning of backdoor triggers during the pretraining phase. However, these methods often suffer from accuracy degradation and can significantly increase model training costs, making them impractical for large-scale applications. In contrast, robust purification methods focus on removing potential backdoor features from a well-trained model. Generally, purification techniques involve reversing potential backdoor triggers [44, 45, 46, 50] and applying fine-tuning or pruning to address backdoors using a limited amount of clean data [29, 32, 49, 55]. While these purification methods reduce training costs, they also achieve state-of-the-art defense performance [32, 55]. Therefore, in this work, *we mainly focus on evaluations of robust purification methods against backdoor attacks.*

Loss Landscape and Linear Mode Connectivity. Early works [12, 14, 22] conjectured and empirically verified that different DNN loss minima can be connected by low-loss curves. In the context of the pretrain-fine-tune paradigm, Neyshabur et al. [33] observe that the pretrained weights guide purified models to the same flat basin of the loss landscape, which is close to the pretrained checkpoint. Frankle et al. [13] also observe and define the linear case of mode connectivity, *Linear Mode Connectivity* (LMC). LMC refers to the absence of the loss barrier when interpolating linearly between solutions that are trained from the same initialization. The shared initialization can either be a checkpoint in early training [13] or a pretrained model [33]. *Our work is inspired by [33] and also utilizes LMC to investigate the properties of purified models in relation to backdoor safety.*

Deceptive AI and Superficial Safety. Nowadays, DNNs are typically pretrained on large-scale datasets, such as web-scraped data, primarily using next-token prediction loss [3], along with simple contrastive [6] and classification [23] objective. While these simplified pretraining objectives can lead to the learning of rich and useful representations, they may also result in deceptive behaviors that can mislead humans [10]. One such deceptive behavior is the presence of backdoors [9, 20]. A compromised model can be indistinguishable from a normal model to human supervisors, as both behave similarly in the absence of the backdoor trigger. To address this critical safety risk, researchers propose post-training alignment procedures, such as safety fine-tuning [35, 48]. However, several studies indicate that the changes from fine-tuning are *superficial* [31, 54]. As a result, these models retain harmful capabilities and knowledge from pretraining, which can be elicited by harmful fine-tuning [37, 51] or specific out-of-distribution (OOD) inputs [47, 52]. *We study this phenomenon in the context of backdoor threats and offer a deeper understanding along with mitigation strategies.*

3 Revealing Superficial Safety of Backdoor Defenses by Accessing Post-purification Robustness

While current backdoor purification methods can achieve a very low Attack Success Rate (ASR) against backdoor attacks, this does not necessarily equate to the complete elimination of inserted backdoor features. Adversaries may further exploit these residual backdoor features to reconstruct and reactivate the implanted backdoor, as discussed in Section 3.3. This is particularly important because purified models are often used in various downstream scenarios, such as customized fine-tuning [37] for critical tasks [20]. Therefore, it is crucial to provide a way to measure the robustness of purified models in defending against backdoor re-triggering, which we define as **"Post-Purification Robustness"**.

In this section, we first introduce a simple and straightforward strategy called the Retuning Attack (RA) to assess post-purification robustness. Building on the RA, we then present a practical threat known as the Query-based Reactivation Attack (QRA), which exploits the vulnerabilities in post-purification robustness to reactivate the implanted backdoor in purified models, using only model querying. First, we will introduce the preliminaries and evaluation setup.

3.1 Problem Setup

Backdoor Purification. In this work, we focus on the poisoning-based attack due to its practicality and stealthiness. We denote the original training dataset as $\mathcal{D}_T \subset (\mathcal{X}, \mathcal{Y})$. A few training examples $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}_T$ have been transformed by attackers into poisoned examples $(\boldsymbol{x}_p, \boldsymbol{y}_t)$, where $\boldsymbol{x}_p$ is poisoned example with inserted trigger and a target label $\boldsymbol{y}_t$. Following previous works [29, 32, 39, 48, 49], only a limited amount of clean data D_t are used for fine-tuning or pruning. For trigger-inversion methods [44, 50], we denote the reversed backdoored samples obtained through reversing methods as $(\boldsymbol{x}_r, \boldsymbol{y}) \in \mathcal{D}_r$. We evaluate several mainstreamed purification methods, including pruning-based defense *ANP* [49]; robust fine-tuning defense *I-BAU* [53] (referred to as BAU for short), *FT-SAM* [55] (referred to as SAM for short), *FST* [32], as well as the state-of-the-art trigger-reversing defense *BTI-DBF* [50] (referred to as BTI for short). BTI purifies the backdoored model by using both reversed backdoored samples $\mathcal{D}_r$ and the clean dataset $\mathcal{D}_t$ while the others use solely the clean dataset $\mathcal{D}_t$. We also include *exact purification (EP)* that assumes that the defender has full knowledge of the exact trigger and fine-tunes the models using real backdoored samples with correct labels $(\boldsymbol{x}_p, \boldsymbol{y})$.

Attack Settings. Following [32], we evaluate four representative data-poisoning backdoors including three dirty-label attacks (BadNet [16], Blended [7], SSBA [28]), and one clean-label attack (LC [43]). All experiments are conducted on BackdoorBench [48], a widely used benchmark for backdoor learning. We employ three poisoning rates, $10\%, 5\%$, and 1% (in *Appendix*) for backdoor injection and conduct experiments on three widely used image classification datasets, including CIFAR-10 [24], Tiny-ImageNet [8], and CIFAR-100 [24]. For model architectures, we following [32], and adopt the ResNet-18, ResNet-50 [17], and DenseNet-161 [18] on CIFAR-10. For CIFAR-100 and Tiny-ImageNet, we adopt pretrained ResNet-18 on *ImageNet1K* to obtain high clean accuracy as suggested by [32, 50]. More details about experimental settings are shown in *Appendix* B.

Evaluation Metrics. Following previous backdoor works, we take two evaluation metrics, including *Clean Accuracy (C-Acc)* (i.e., the prediction accuracy of clean samples) and *Attack Success Rate (ASR)* (i.e., the prediction accuracy of poisoned samples to the target class) where a lower ASR indicates a better defense performance. We further adopt *O-ASR* and *P-ASR* metrics. The O-ASR metric represents the defense performance of original defense methods, while the P-ASR metric indicates the ASR after applying the RA or QRA.

3.2 Purified Models Are Vulnerable to Retuning Attack

Our objective is to investigate whether purified models with low ASR completely eliminate the inserted backdoor features. To accomplish this, it is essential to develop a method for assessing the degree to which purified models have indeed forgotten these triggers. In this section, *we begin with a white-box investigation where the attacker or evaluator has access to the purified model's parameters.* Here we introduce a simple tuning-based strategy named the **Retuning Attack (RA)** [37, 42] to conduct an initial evaluation. Specifically, we construct a dataset for model retuning, which comprises a few backdoored samples (less than 1% of backdoored samples used during the training process). To maintain C-Acc, we also include benign samples from the training set, resulting in a total RA dataset with 1000 samples. We subsequently retune the purified models using this constructed dataset through a few epochs (5 epochs in our implementation). This approach is adopted because a clean model can not be able to learn a backdoor; thus, if the purified models quickly regain ASR during the retuning process, it indicates that some residual backdoor features still exist in these purified models. Implementation details of the RA can be found in the *Appendix* B.2.

As shown in Figure 1, we observe that *despite achieving very low ASR, all purification methods quickly recover backdoor ASR with Retuning Attack.* Their quickly regained ASR presents a stark contrast to that of clean models and remains consistent across different datasets, model architectures, and poisoning rates. Note that the pruning method (ANP) and the fine-tuning method (FST), which achieve state-of-the-art defense performance, still exhibit vulnerability to RA, with an average recovery of approximately 82% and 85% ASR, respectively. In stark contrast, the EP method stands out as *it consistently maintains a low ASR even after applying RA, demonstrating exceptional post-purification robustness.* Although impractical with full knowledge of the backdoor triggers, the EP method validates the possibility of maintaining a low attack success rate to ensure post-purification robustness against RA attacks.

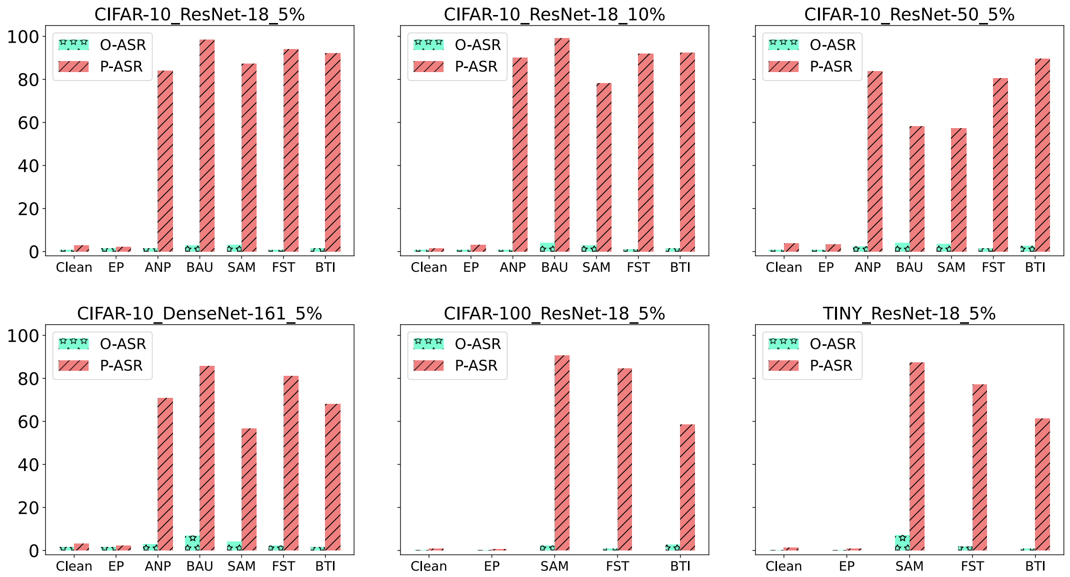

Figure 1: The robustness performance against various attack settings. The title consists of the used dataset, model, and poisoning rate. The *O-ASR* metric represents the defense performance of original defense methods, while the *P-ASR* metric indicates the ASR after applying the RA. All metrics are measured in percentage (%) Here we report the average results among backdoor attacks and defer more details in *Appendix* C.1.

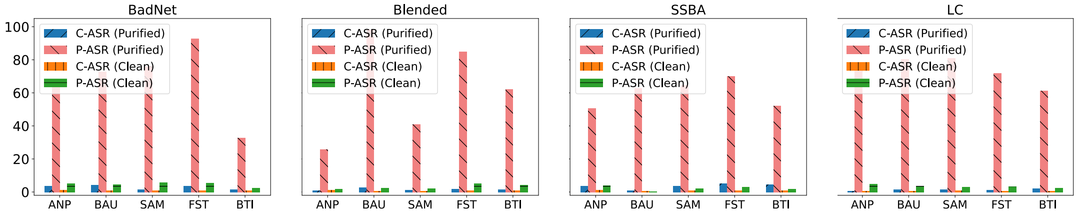

Figure 2: Experimental results of QRA on both the *purified* and *clean* models against four types of backdoor attacks. We evaluate the QRA on CIFAR-10 with ResNet-18 and the poisoning rate is set to 5%. Additional results of QRA are demonstrated in *Appendix* C.3.

Moreover, this evident contrast highlights the significant security risks associated with current backdoor safety tuning methods. While these methods may initially appear robust due to a significantly reduced ASR, they are fundamentally vulnerable to backdoor reactivation, which can occur with just a few epochs of model tuning. This superficial safety underscores the urgent need for more comprehensive evaluations to ensure lasting protection against backdoor attacks. It is crucial to implement faithful evaluations that thoroughly assess the resilience of purified models, rather than relying solely on superficial metrics, to truly safeguard against the persistent threat of backdoor vulnerabilities.

3.3 Reactivating Backdoor on Purified Models through Queries

Although our previous experiments on RA demonstrate that current purification methods insufficiently eliminate learned backdoor features, it is important to note that the success of this tuning-based method relies on the attackers' capability to change purified models' weights. This is not practical in a real-world threat model. To address this limitation, we propose **Query-based Reactivation Attack (QRA)**, which generates sample-specific perturbations that can reactivate the backdoor using only model querying. Specifically, instead of directly retuning purified models, QRA captures the parameter changes induced by the RA process and translates them into input space as perturbations. These perturbations can then be incorporated into backdoored examples, facilitating the successful reactivation of backdoor behaviors in purified models.

To effectively translate the parameter changes into the input space, it is crucial to ensure that when applying the perturbation generated by QRA, the output of the purified model on perturbed inputs should be aligned with that of the post-RA model on original inputs without perturbations. Formally, we denote the purified model as $\boldsymbol{f}(\boldsymbol{W}_p;\boldsymbol{x})$, the model after RA as $\boldsymbol{f}(\boldsymbol{W}_{ra};\boldsymbol{x})$ and their corresponding logit output as $\boldsymbol{l}(\boldsymbol{W}_p;\boldsymbol{x})$ and $\boldsymbol{l}(\boldsymbol{W}_{ra};\boldsymbol{x})$. Our QRA aims to learn a perturbation generator $\boldsymbol{\phi}(\boldsymbol{\theta};\boldsymbol{x}) : R^d \rightarrow [-1,1]^d$, to produce perturbation $\boldsymbol{\phi}(\boldsymbol{\theta};\boldsymbol{x})$ for each input $\boldsymbol{x}$. We formulate this process into the following optimization problem:

$$\min_{\boldsymbol{\theta}} \left\{ \mathbb{E}_{(\boldsymbol{x})\sim\mathcal{D}_c}[\mathcal{S}(\boldsymbol{l}(\boldsymbol{W}_{ra};\boldsymbol{x}), \boldsymbol{l}(\boldsymbol{W}_p;\epsilon*\boldsymbol{\phi}(\boldsymbol{\theta};\boldsymbol{x})+\boldsymbol{x}))] \right\}. \tag{1}$$

Here $\mathcal{S}$ is the distance metric between two output logits, $\mathcal{D}_c$ is a compositional dataset incorporating both benign and backdoored samples, and ϵ controls the strength of perturbation ($\epsilon = 16/255$ in our implementation). We utilize the Kullback–Leibler (KL) divergence [11] for $\mathcal{S}$ and a Multilayer Perception (MLP) for $\boldsymbol{\phi}(\boldsymbol{\theta};\boldsymbol{x})$. Specifically, we flatten the input image into a one-dimensional vector before feeding into the $\boldsymbol{\phi}(\boldsymbol{\theta};\boldsymbol{x})$, and obtain the generated perturbation by reshaping it back to the original size. Details of the MLP architecture and training hyperparameters can be found in the *Appendix* B.2.

However, we observe that directly optimizing Equation 1 not only targets purified models but also successfully attacks the clean model. We conjecture that this may stem from the inaccurate inversion of reactivated backdoor triggers, which can exploit a clean model in a manner similar to adversarial examples [21, 30, 38]. To mitigate such adversarial behavior, we introduce a regularization term aimed at minimizing backdoor reactivation on the clean model. Given that accessing the clean model may not be practical, we utilize the EP model $\boldsymbol{f}(\boldsymbol{W}_e;\boldsymbol{x})$ as a surrogate model instead. In sum, we formulate the overall optimization objective as follows:

$$\min_{\boldsymbol{\theta}} \left\{ \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}_c}[\mathcal{S}(\boldsymbol{l}(\boldsymbol{W}_{ra};\boldsymbol{x}), \boldsymbol{l}(\boldsymbol{W}_p;\epsilon*\boldsymbol{\phi}(\boldsymbol{\theta};\boldsymbol{x})+\boldsymbol{x}))+\alpha*\mathcal{L}(\boldsymbol{f}(\boldsymbol{W}_e;\epsilon*\boldsymbol{\phi}(\boldsymbol{\theta};\boldsymbol{x})+\boldsymbol{x}),\boldsymbol{y})] \right\}, \tag{2}$$

where α represents the balance coefficient and the cross-entropy loss is used for $\mathcal{L}$.

We demonstrate our experimental results against five purification methods on CIFAR-10 in Figure 2. Here, we report the *C-ASR* and *P-ASR*, which represent the ASR when evaluating with perturbed clean and perturbed poisoned images, respectively. Notably, our QRA could effectively reactivate the backdoor behaviors on purified models, resulting in a significant increase of 66.13% on average in P-ASR. Our experiments also demonstrate a consistently low C-ASR on purified models, which indicates that the perturbations generated by QRA effectively reactivate the backdoored examples without affecting the predictions of benign images. Besides, the perturbation generated with QRA exclusively works on the output of backdoored samples on purified models, leading to both a low C-ASR and P-ASR on clean models. This observation further indicates that *the reversed pattern generated by QRA is not a typical adversarial perturbation but rather an accurate depiction of the parameter changes necessary for backdoor reactivation.*

Furthermore, it is worth noting that attackers may lack knowledge about the specific backdoor defense techniques utilized by the defender in practice. Thus, we embark on an initial investigation to explore the transferability of our QRA method across unknown purification methods. Specifically, we aim to determine whether the reversed perturbations optimized for one particular defense method can effectively attack purified models with other purification techniques. As shown in Figure 3, our QRA demonstrates a degree of successful transferability across various defense techniques, achieving an average P-ASR of 32.1% against all purification techniques. These results underscore the potential of QRA to attack purified models, even without prior knowledge of the defense methods employed by defenders. It also highlights the practical application of QRA in real-world situations.

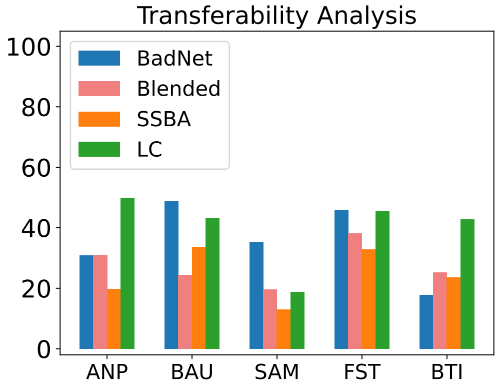

Figure 3: The results of the QRA transferability. The defense method used in the attack is represented on the x-axis, while the y-axis shows the average P-ASR across other purifications.

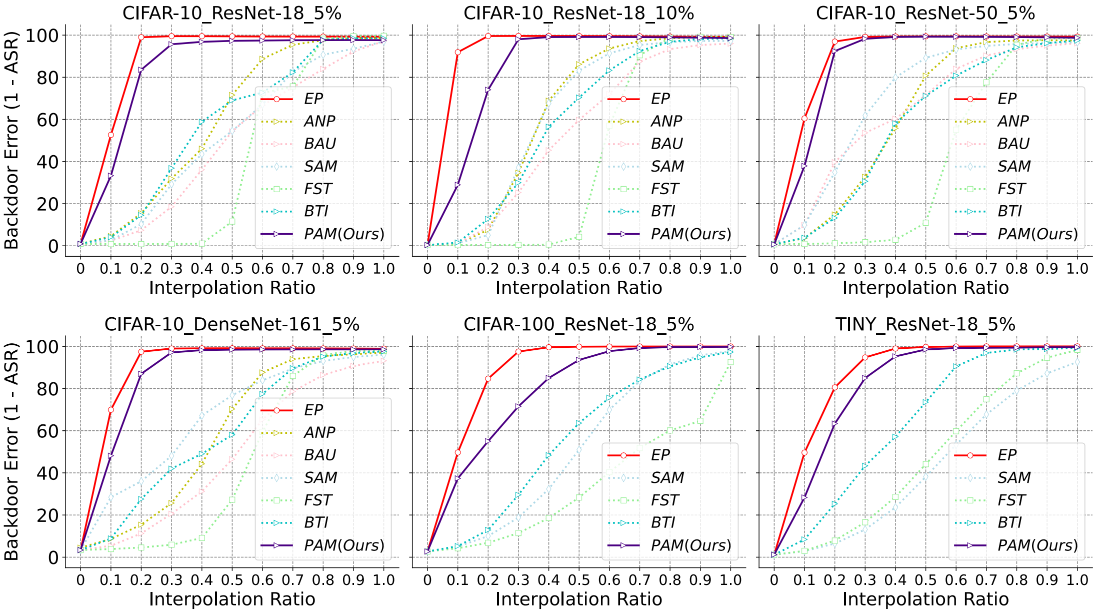

Figure 4: The evaluation of backdoor-connected path against various attack settings. The x-axis and y-axis denote the interpolation ratio t and backdoor error (1-ASR) respectively. For each attack setting, we report the average results among backdoor attacks.

4 Investigating and Mitigating Superficial Safety

4.1 Investigating the Superficial Safety through Linear Mode Connectivity

While our previous evaluations indicate that only the EP model demonstrates exceptional post-purification robustness compared to current backdoor safety tuning methods, the factors contributing to the effectiveness of EP remain unclear. Motivated by prior studies examining fine-tuned models [13, 14, 33], we propose to investigate this intriguing phenomenon from the perspective of the loss landscape using Linear Mode Connectivity (LMC).

Following [13, 33], let $\mathcal{E}(\boldsymbol{W}; \mathcal{D}_l)$ represent the testing error of a model $\boldsymbol{f}(\boldsymbol{W}; \boldsymbol{x})$ evaluated on a dataset $\mathcal{D}_l$. For $\mathcal{D}_l$, we use backdoor testing samples. $\underline{\mathcal{E}_t(\boldsymbol{W}_0, \boldsymbol{W}_1; \mathcal{D}_l) = \mathcal{E}((1-t)\boldsymbol{W}_0 + t\boldsymbol{W}_1; \mathcal{D}_l)}$ for $t \in [0, 1]$ is defined as the error path of model created by linear interpolation between the $\boldsymbol{f}(\boldsymbol{W}_0; \boldsymbol{x})$ and $\boldsymbol{f}(\boldsymbol{W}_1; \boldsymbol{x})$. We also refer to it as the backdoor-connected path. Here we denote the $\boldsymbol{f}(\boldsymbol{W}_0; \cdot)$ as backdoored model and $\boldsymbol{f}(\boldsymbol{W}_1; \cdot)$ as the purified model. We show the LMC results of the backdoor error in Figure 4 and 5. For each attack setting, we report the average results among backdoor attacks. More results on other datasets and models are shown in *Appendix* C.2.

The backdoored model and purified models reside in separate loss basins, linked by a backdoor-connected path. We present the results of LMC between purified and backdoored models in Figure 4. It is clear from the results that all purified models exhibit significant error barriers along the backdoor-connected path to backdoored model. This indicates that backdoored and purified models reside in different loss basins. Additionally, we conduct LMC between purified models with EP and with other defense techniques, as depicted in Figure 5. We observe a consistently high error without barriers, which indicates that these purified models reside within the same loss basin. Based on these two findings, we conclude that backdoored and purified models reside in two distinct loss basins connected through a backdoor-connected path.

EP deviates purified models from the backdoored model along the backdoor-connected path, resulting in a more distantly robust solution. Although introducing a high loss barrier, we observe notable distinctions between the LMC of the EP model (red solid line) and purified models (dotted lines). We observe a stable high backdoor error along the backdoor-connected path of EP until $t < 0.2$, where the interpolated model parameter $\boldsymbol{W}$ has over 80% weight from the backdoored model. In

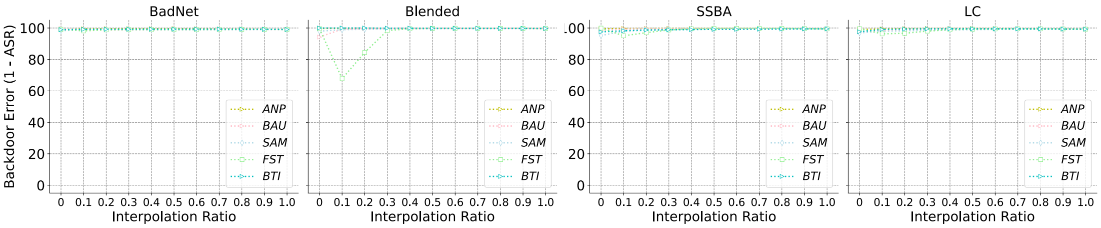

Figure 5: The LMC path connected from other defense techniques to EP. We evaluate the LMC results on CIFAR-10 with ResNet-18, and set the poisoning rate to 5%.

contrast, other purification models show a tendency to exhibit significant increases in ASR along the path, recovering more than 20% ASR when $t < 0.5$, while the ASR for the EP model remains low ($\leq 2\%$). This clear contrast suggests: ***1)*** *the current purification methods prematurely converge to a non-robust solution with low ASR, which is still close to the backdoored model along the backdoor-connected path;* ***2)*** *compared with purified models, EP makes the purified model significantly deviate from the backdoored checkpoint along the backdoor-connected path, resulting in a more robust solution against RA.*

Accurately specified supervision is crucial for achieving stable backdoor safety. As demonstrated in our observations, the EP method attains stable robustness in the context of the RA, whereas its proxy version, the BTI method, employs reversed backdoor data as a substitute for real backdoor data, resulting in suboptimal post-purification robustness. Furthermore, notable discrepancies are evident in the Backdoor LMC results. These findings underscore that current methods for reversing backdoor triggers are still unable to accurately recover all backdoor features [40], thereby emphasizing the importance of precisely specified supervision in achieving stable backdoor safety. Although data generated by the BTI method does not accurately recover all backdoor features, it could serve as an effective and usable supervision dataset. In the following section, we propose an improved safety tuning method designed to mitigate superficial safety concerns based on this proxy dataset.

4.2 Enhancing Post-Purification Robustness Through Path-Aware Minimization

Motivated by our analysis, we propose a simple tuning defense method called **Path-Aware Minimization (PAM)**, which aims to enhance post-purification robustness by promoting more deviation from the backdoored model along the backdoor-connected path like the EP method.

Since there are no real backdoor samples $\boldsymbol{x}_p$ available, we employ the synthetic backdoored samples $\boldsymbol{x}_r$ from the trigger-reversing method BTI [50] as a substitute to get the backdoor-connected path. Although BTI has a similar LMC path curve with the EP model in Figure 5, as we have discussed, tuning solely with $\boldsymbol{x}_r$ would lead to finding a non-robust solution with low ASR.

To avoid converging to such a solution, we propose utilizing the gradients of an interpolated model $\boldsymbol{W}_d$ between $\boldsymbol{W}_0$ and $\boldsymbol{W}$ to update the current solution $\boldsymbol{W}$. As illustrated in Figure 4, the interpolated model, which lies between $\boldsymbol{W}_0$ and $\boldsymbol{W}$, exhibits a higher ASR compared to $\boldsymbol{W}$. By leveraging the gradients from the interpolated model, we can perform additional updates on the $\boldsymbol{W}$ which prevents premature convergence towards local minima and results in a solution that deviates from the backdoored model along this path. Specifically, for $\boldsymbol{W}$, we first take a path-aware step $\rho\frac{\boldsymbol{W}_d}{\|\boldsymbol{W}_d\|_2}$ ($\boldsymbol{W}_d = \boldsymbol{W}_0 - \boldsymbol{W}$) towards to $\boldsymbol{W}_0$ and obtain the interpolated model $\boldsymbol{W} + \rho\frac{\boldsymbol{W}_d}{\|\boldsymbol{W}_d\|_2}$. Then we compute its gradient on $\boldsymbol{x}_r$ to update $\boldsymbol{W}$. We formulate our objective function as follows:

$$\min_{\boldsymbol{W}} \left\{ \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}_r\cup\mathcal{D}_t}[\mathcal{L}(\boldsymbol{f}(\boldsymbol{W} + \rho\frac{\boldsymbol{W}_d}{\|\boldsymbol{W}_d\|_2}; \boldsymbol{x}), \boldsymbol{y})] \right\}, \quad s.t.\ \boldsymbol{W}_d = \boldsymbol{W}_0 - \boldsymbol{W}, \tag{3}$$

where ρ represents the size of the path-aware step. Typically, a larger ρ indicates a larger step towards the backdoored model $\boldsymbol{W}_0$ along our backdoor-connected path and also allows us to obtain a larger gradient update for $\boldsymbol{W}$, which results in more deviation from the backdoored model along the backdoor-connected path. The detailed algorithm is summarized in the Algorithm 1.

Table 1: The post-purification robustness performance of PAM on CIFAR-10. The *O-Backdoor* indicates the original performance of backdoor attacks, *O-Robustness* metric represents the purification performance of the defense method, and the *P-Robustness* metric denotes the post robustness after applying RA. All metrics are measured in percentage (%).

Method	Evaluation Mode	BadNet		Blended		SSBA		LC		Avg	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
ResNet18 (5%)	O-Backdoor	94.04	99.99	94.77	100.0	94.60	96.38	94.86	99.99	94.57	99.09
	O-Robustness (EP)	93.57	0.94	93.47	3.37	93.39	0.53	93.64	0.52	93.52	1.34
	P-Robustness (EP)	92.27	1.08	93.36	4.93	92.05	2.47	93.69	0.41	92.84	2.22
	O-Robustness (PAM)	92.11	1.14	93.34	1.67	92.96	1.24	92.32	4.92	92.68	2.24
	P-Robustness (PAM)	91.66	3.90	93.38	2.69	92.20	3.31	92.15	8.31	92.35	4.55
ResNet18 (10%)	O-Backdoor	93.73	100.0	94.28	100.0	94.31	98.71	85.80	100.0	92.03	99.68
	O-Robustness (EP)	92.78	0.93	93.34	1.52	93.10	0.71	92.12	0.16	92.84	0.83
	P-Robustness (EP)	92.17	1.58	93.06	4.78	92.04	3.70	91.79	2.03	92.27	3.02
	O-Robustness (PAM)	92.43	1.73	92.63	0.22	92.89	1.37	91.06	2.31	92.25	1.41
	P-Robustness (PAM)	91.77	1.47	91.85	7.44	92.01	2.63	90.98	3.37	91.65	3.73
ResNet50 (5%)	O-Backdoor	93.81	99.92	94.53	100.0	93.65	97.70	94.57	99.90	94.14	99.38
	O-Robustness (EP)	92.84	0.98	92.10	1.12	91.84	0.43	92.91	0.41	92.42	0.74
	P-Robustness (EP)	92.33	1.48	91.36	3.51	89.26	6.28	92.33	1.71	91.32	3.25
	O-Robustness (PAM)	92.58	0.94	92.59	0.24	92.36	1.62	91.99	2.14	92.38	1.23
	P-Robustness (PAM)	91.49	1.46	92.75	0.64	92.32	14.23	90.60	3.54	91.79	4.97
DenseNet161 (5%)	O-Backdoor	89.85	100.0	89.59	98.72	88.83	86.75	90.13	99.80	89.60	96.32
	O-Robustness (EP)	88.58	1.52	88.00	3.21	88.17	0.69	88.59	0.56	88.34	1.50
	P-Robustness (EP)	88.03	2.77	87.13	3.54	86.33	1.47	88.94	0.95	87.61	2.18
	O-Robustness (PAM)	88.70	1.22	87.02	1.08	87.62	1.61	87.70	1.81	87.76	1.43
	P-Robustness (PAM)	87.03	3.04	86.49	1.94	85.89	3.72	86.44	9.39	86.42	4.52

Table 2: The post-purification robustness performance of PAM on CIFAR-100 and Tiny-ImageNet. Note that we omit the LC attack for both the CIFAR-100 and Tiny-ImageNet, as it does not consistently achieve successful backdoor implantation. All metrics are measured in percentage (%).

Method	Evaluation Mode	BadNet		Blended		SSBA		Avg	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
CIFAR-100 (5%)	O-Backdoor	78.91	99.51	78.98	100.0	78.59	92.38	78.83	97.30
	O-Robustness (EP)	76.89	0.04	76.95	0.04	76.49	0.05	76.78	0.43
	P-Robustness (EP)	76.82	0.10	76.18	0.07	76.12	1.32	76.37	0.50
	O-Robustness (PAM)	74.97	0.29	76.64	0.19	75.07	0.14	75.56	0.21
	P-Robustness (PAM)	75.72	0.76	76.12	0.13	74.75	1.95	75.53	0.95
TINY (5%)	O-Backdoor	72.60	99.01	73.68	99.99	73.02	97.05	73.10	98.63
	O-Robustness (EP)	70.90	0.02	71.11	0.01	70.24	0.01	70.75	0.13
	P-Robustness (EP)	70.59	0.65	70.93	0.09	69.98	1.90	70.50	0.88
	O-Robustness (PAM)	68.78	0.06	67.89	0.14	68.01	4.47	68.23	1.56
	P-Robustness (PAM)	68.97	7.81	66.86	0.37	67.39	14.26	67.74	7.48

Algorithm 1 Path-Aware Minimization (PAM)

```
Input: Tuning dataset $\mathcal{D}_T = \mathcal{D}_r \cup \mathcal{D}_t$; Backdoored model $\boldsymbol{f}(\boldsymbol{W}_0; \boldsymbol{x})$; Learning rate $\eta$; Path-aware step size $\rho$; Tuning iterations $I$
Output: Purified model
Initialize $\boldsymbol{W}_1$ with $\boldsymbol{W}_0$
for $i = 1, \ldots, I$ do
    Sample a mini-batch $\mathcal{B}_i$ from the tuning set $\mathcal{D}_T$;
    Calculate parameter difference: $\boldsymbol{W}_d = \boldsymbol{W}_0 - \boldsymbol{W}_i$;
    Obtain interpolated parameters: $\tilde{\boldsymbol{W}}_i = \boldsymbol{W}_i + \rho \frac{\boldsymbol{W}_d}{\|\boldsymbol{W}_d\|_2}$;
    Calculate gradients of $\tilde{\boldsymbol{W}}_i$:
    $\boldsymbol{g}_{\tilde{\boldsymbol{W}}_i} = \nabla_{\tilde{\boldsymbol{W}}_i} \frac{1}{|\mathcal{B}_i|} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{B}_i} \mathcal{L}(\boldsymbol{f}(\tilde{\boldsymbol{W}}_i; \boldsymbol{x}, \boldsymbol{y})$
    Update current parameters: $\boldsymbol{W}_{i+1} = \boldsymbol{W}_i - \eta \boldsymbol{g}_{\tilde{\boldsymbol{W}}_i}$
end for
return Purified model $\boldsymbol{f}(\boldsymbol{W}_I; \boldsymbol{x})$
```

Post-Purification Robustness of PAM. We evaluate the post-purification robustness of PAM against RA and make a comparison with EP. Using the same experimental settings in Section 3.1, we set ρ to 0.5 for Blended and SSBA and 0.9 for the BadNet and LC attack on CIFAR-10 and set ρ to 0.4 for both CIFAR-100 and Tiny-ImageNet. The results on CIFAR-10, CIFAR-100 and Tiny-ImageNet are shown in Table 1 and Table 2. We could observe that PAM significantly improves post-purification robustness against RA. It achieves *a comparable robustness performance to the EP*, with an average ASR lower than 4.5% across all three datasets after RA. In comparison to previous experimental results in Section 3.2, our PAM outperforms existing defense methods by a large margin in terms of post-purification robustness. Our PAM also achieves a stable purification performance (O-ASR), reducing the ASR below 2% on all three datasets and preserves a high C-Acc as well, yielding only around 2% drop against the original performance of C-Acc.

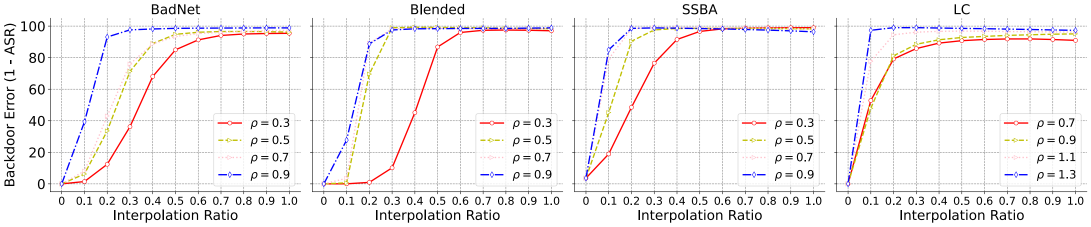

Figure 6: Ablation studies of PAM across different values of ρ against four types of backdoor attacks. We conduct our evaluations on CIFAR-10 with ResNet-18.

To further verify the post-purification with PAM, following the experimental setting in Section 4.1, we also show the LMC results of PAM in Figure 4. It is clearly observed that our PAM significantly deviates purified models from the backdoored model along the backdoor-connected path, leading to a robust solution similar to the EP method. This further confirms our findings about post-purification robustness derived from LMC.

Sensitivity Analysis of ρ. We evaluate the performance of PAM with various values of ρ and conduct experiments on CIFAR-10 with ResNet-18 against four attacks. The experimental results are shown in Figure 6. Note that as ρ increases, we increase the error barrier along the connected path, indicating an enhanced deviation of our purified model from the backdoored model. However, simply increasing ρ would also compromise the competitive accuracy (C-Acc) of the purified model. In practice, it is essential to select an appropriate ρ to achieve a balance between post-purification robustness and C-Acc. We present the model performance across various ρ values in Table 10 of the *Appendix*. We can observe that as ρ rises, there is a slight decrease in clean accuracy alongside a significant enhancement in robustness against the RA. Additionally, we note that performance is relatively insensitive to ρ when it exceeds 0.3. Given that we primarily monitor C-Acc (with the validation set) in practice, we aim to achieve a favorable trade-off between these two metrics. Therefore, we follow the approach of FST [32] and select ρ to ensure that C-Acc remains above a predefined threshold, such as 92%.

5 Conclusions and Limitations

In this paper, we seek to address the following question: Do current backdoor safety tuning methods genuinely achieve reliable backdoor safety by merely relying on reduced Attack Success Rates? To investigate this issue, we first employ the Retuning Attack to evaluate the post-purification robustness of purified models. Our primary experiments reveal a significant finding: existing backdoor purification methods consistently exhibit an increased ASR when subjected to the RA, highlighting the superficial safety of these approaches. Building on this insight, we propose a practical Query-based Reactivation Attack, which enables attackers to re-trigger the backdoor from purified models solely through querying. We conduct a deeper analysis of the inherent vulnerabilities against RA using Linear Mode Connectivity, attributing these vulnerabilities to the insufficient deviation of purified models from the backdoored model along the backdoor-connected path. Inspired by our analysis, we introduce a simple tuning defense method, Path-Aware Minimization, which actively promotes deviation from the backdoored model through additional model updates along the interpolated path. Extensive experiments demonstrate the effectiveness of PAM, surpassing existing purification techniques in terms of post-purification robustness.

This study represents an initial attempt to evaluate post-purification robustness via RA. While we propose the practical QRA method, future work is essential to develop more efficient evaluation techniques that can faithfully assess post-purification robustness. The need for such evaluations is critical, as they ensure that the perceived safety of purified models is not merely superficial. Additionally, we recognize the significant potential for enhancing QRA in the context of transfer attacks, which we aim to explore in future research. Furthermore, we plan to broaden our research by incorporating additional backdoor attack strategies and safety tuning methods applicable to generative models, such as LLMs and diffusion models [1, 20, 37, 40], in future work. We will also apply our existing framework of evaluation, analysis, and safety tuning method to research on unlearning in large models.

References

[1] Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, et al. Foundational challenges in assuring alignment and safety of large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. Survey Certification, Expert Certification. 1, 5

[2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2938–2948. PMLR, 2020. 1

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. 2

[4] Nicholas Carlini and Andreas Terzis. Poisoning and backdooring contrastive learning. In *International Conference on Learning Representations*, 2022. 1, 2

[5] Nicholas Carlini, Matthew Jagielski, Christopher A Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. Poisoning web-scale training datasets is practical. *arXiv preprint arXiv:2302.10149*, 2023. 1, 2

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[7] 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. 1, 2, 3.1

[8] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017. 3.1

[9] Paul Christiano. Mechanistic anomaly detection and elk. *Alignment Research Center*, 2022. URL https://ai-alignment.com/mechanistic-anomaly-detection-and-elk-fb84f4c6d0dc. 2

[10] Paul Christiano, Ajeya Cotra, and Mark Xu. Eliciting latent knowledge: How to tell if your eyes deceive you. technical report. *Alignment Research Center*, 2021. URL https://docs.google.com/document/d/1WwsnJQstPq91_Yh-Ch2XRL8H_EpsnjrC1dwZXR37PC8/edit. 2

[11] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999. 3.3

[12] Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pages 1309–1318. PMLR, 2018. 2

[13] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020. 1, 2, 4.1

[14] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018. 2, 4.1

[15] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[16] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019. 1, 2, 3.1

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 3.1

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 3.1

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

[20] Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M Ziegler, Tim Maxwell, Newton Cheng, et al. Sleeper agents: Training deceptive llms that persist through safety training. *arXiv preprint arXiv:2401.05566*, 2024. 1, 2, 3, 5

[21] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019. 3.3

[22] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. 2

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 2

[24] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 3.1

[25] Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning-industry perspectives. In *2020 IEEE security and privacy workshops (SPW)*, pages 69–75. IEEE, 2020. 1

[26] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems*, 34:14900–14912, 2021. 2, B.1

[27] 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*, pages 19837–19854. PMLR, 2023. 1

[28] 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*, pages 16463–16472, 2021. 2, 3.1

[29] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 273–294. Springer, 2018. 2, 3.1

[30] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*, 2016. 3.3

[31] Ekdeep Singh Lubana, Eric J Bigelow, Robert P Dick, David Krueger, and Hidenori Tanaka. Mechanistic mode connectivity. In *International Conference on Machine Learning*, pages 22965–23004. PMLR, 2023. 2

[32] 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. 1, 2, 3.1, 3.1, 4.2, B.1, B.2, C.4

[33] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020. 1, 2, 4.1

[34] Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2021. 2

[35] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022. 2

[36] Xiangyu Qi, Tinghao Xie, Yiming Li, Saeed Mahloujifar, and Prateek Mittal. Revisiting the assumption of latent separability for backdoor defenses. In *The eleventh international conference on learning representations*, 2023. 1

[37] Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv preprint arXiv:2310.03693*, 2023. 2, 3, 3.2, 5

[38] Zeyu Qin, Yanbo Fan, Yi Liu, Li Shen, Yong Zhang, Jue Wang, and Baoyuan Wu. Boosting the transferability of adversarial attacks with reverse adversarial perturbation. *Advances in neural information processing systems*, 35:29845–29858, 2022. 3.3

[39] Zeyu Qin, Liuyi Yao, Daoyuan Chen, Yaliang Li, Bolin Ding, and Minhao Cheng. Revisiting personalized federated learning: Robustness against backdoor attacks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 4743–4755, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030. 2, 3.1

[40] Javier Rando, Francesco Croce, Kryštof Mitka, Stepan Shabalin, Maksym Andriushchenko, Nicolas Flammarion, and Florian Tramèr. Competition report: Finding universal jailbreak backdoors in aligned llms. *arXiv preprint arXiv:2404.14461*, 2024. 4.1, 5

[41] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1354–1371. IEEE, 2022. 2

[42] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 1, 3.2

[43] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019. 2, 3.1

[44] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019. 1, 2, 3.1

[45] Yuhang Wang, Huafeng Shi, Rui Min, Ruijia Wu, Siyuan Liang, Yichao Wu, Ding Liang, and Aishan Liu. Universal backdoor attacks detection via adaptive adversarial probe. *arXiv preprint arXiv:2209.05244*, 2022. 2

[46] Zhenting Wang, Kai Mei, Juan Zhai, and Shiqing Ma. Unicorn: A unified backdoor trigger inversion framework. In *The Eleventh International Conference on Learning Representations*, 2022. 2

[47] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36, 2024. 2

[48] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoorbench: A comprehensive benchmark of backdoor learning. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 1, 2, 2, 3.1, 3.1, B.1

[49] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. *Advances in Neural Information Processing Systems*, 34:16913–16925, 2021. 1, 2, 3.1, B.1

[50] 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*, 2023. 1, 2, 3.1, 3.1, 4.2, B.1

[51] Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models. *arXiv preprint arXiv:2310.02949*, 2023. 2

[52] Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. GPT-4 is too smart to be safe: Stealthy chat with LLMs via cipher. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[53] 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. 2, 3.1

[54] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[55] 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*, pages 4466–4477, 2023. 1, 2, 3.1, B.1, B.2

A Social Impact

The prevalence of Deep Neural Networks (DNNs) in modern society relies heavily on massive amounts of training data from diverse sources. However, in the absence of rigorous monitoring mechanisms, these data resources become susceptible to malicious manipulation, resulting in unforeseen and potentially harmful consequences. Among the various concerns associated with the training dataset, backdoor attacks pose a significant threat. These attacks can manipulate the behavior of a well-trained model by poisoning the training set with backdoored samples, often at a low cost and without requiring complete control over the training process. While existing defense methods have demonstrated effective backdoor purification by achieving low Attack Success Rates (ASR), they still exhibit vulnerabilities that allow adversaries to reactivate the injected backdoor behavior easily. In our work, instead of solely focusing on backdoor ASR, we investigate the effectiveness of modern purification techniques from the perspective of post-purification robustness. We aim to enhance the post-purification robustness of backdoor defense, mitigating the potential for malicious manipulation of deployed models even after backdoor purification. In sum, our work hopes to move an initial step towards improving post-purification robustness while also contributing to another aspect of understanding and enhancing machine learning security.

B Experimental Settings

In this section, we provide detailed information about the experimental settings used in our evaluations. This includes the dataset, training details, and the selection of hyperparameters. All experiments were conducted using 4 NVIDIA 3090 GPUs. We ran all experiments 3 times and averaged all results over 3 random seeds.

B.1 Datasets and Models

We follow previous studies [26, 32, 48, 49] on backdoor learning, and conduct our experiments on three widely used datasets including CIFAR-10, CIFAR-100, and Tiny-ImageNet.

- CIFAR-10 is a widely used dataset in the backdoor literature, comprising images with a resolution of 32×32 and 10 categories. For backdoor training, we utilize the ResNet-18 model for main evaluation, a commonly used architecture in previous studies [32, 50, 55]. Additionally, we explore other architectures, including the ResNet-50 and DenseNet-161.
- CIFAR-100 and Tiny-ImageNet are two large-scale datasets compared to the CIFAR-10, which include 100 and 200 different categories, respectively. Similar to previous work [32, 50], we utilize the pretrained ResNet-18 on ImageNet-1K provided by *PyTorch* to implement backdoor attacks since directly training from scratch would result in an inferior model performance on C-Acc, hence is not practical in real-world scenarios.

B.2 Implementation Details

Attack Configurations We implement 4 representative poisoning-based attacks and generally follow the implementation from the BackdoorBench[1]. For the BadNet, we utilize the 3×3 checkerboard patch as triggers and choose the lower right corner of the image for backdoor injection by default; for the Blended, we adopt the Gaussian Noise as the backdoor trigger. We set the blend ratio to 0.1 for backdoor training and increase the blend ratio to 0.2 during the inference phase; for SSBA and LC, we follow the original implementation from BackdoorBench without making modifications. In our implementation, we set the default poisoning rate to 5%, which is commonly used in previous studies [32, 55] and additionally explore various poisoning rates including both 1% and 10%. Note that we do not adopt a lower poisoning rate since most of the methods suffer from effectively removing backdoor effects when the poisoning rate is extremely low as indicated by [32]. For all backdoor attacks, the target label is set to be 0 by default.

For CIFAR-10, we adopt an initial learning rate of 0.1 to train all the backdoored models for 100 epochs. For both the CIFAR-100 and Tiny-ImageNet, we utilize pretrained backbones and initialize the classifiers with appropriate class numbers. We adopt a smaller learning rate of 0.001 and fine-tune

[1] https://github.com/SCLBD/backdoorbench

the models for 10 epochs. We upscale the image size up to $224 * 224$ during both the training and inference stages following the implementation of [32].

Baseline Defense Configurations We evaluate several mainstream purification techniques including pruning-based defense (ANP), tuning-based defenses (I-BAU, FT-SAM, and FST), and the state-of-the-art trigger-inversing strategy BTI.

- EU: We add real backdoor triggers to 10% of the overall benign tuning samples, and keep their ground-truth labels unchanged. We fine-tune the model on our tuning dataset for 20 epochs.
- ANP: We follow the original implementation in BackdoorBench, using the default hyperparameters. Regarding model pruning, we set the threshold range from 0.4 to 0.9 and report the best purification result with a low ASR and a high model performance on C-Acc.
- I-BAU: We follow the implementation in BackdoorBench with default configurations. We set the fixed-point approximation iterations to 5 and fine-tune backdoored models for 20 epochs.
- FT-SAM: We follow the implementation from [32] and set the neighborhood size to 1 for CIFAR-10 and 0.5 for both the CIFAR-100 and Tiny-ImageNet. We set the initial learning rate to 0.01 and decrease the learning rate to 0.001 for both the CIFAR-100 and Tiny-ImageNet. We fine-tune backdoored models for 20 epochs on all datasets.
- FST: For FST, we follow the original implementation[2]. We set the feature-shift parameter to 0.1 and the learning rate to 0.01 on CIFAR-10. On the CIFAR-100 and Tiny-ImageNet, we decrease the feature-shift parameter to 0.001 and adopt an initial learning rate of 0.005 for better C-Acc.
- BTI: We follow its original implementation[3] and adopt the BTI (U) since it achieves a better purification performance. We adopt the default hyperparameters for trigger inversion including 20 iterations for decoupling benign features, and 30 iterations for training backdoor generators, and set the norm bound to 0.3 by default. We perform BTI (U) until the loss converges.
- PAM: On CIFAR-10, we set the path-aware step size to 0.5 for the Blended and SSBA; while we increase the step size to 0.9 for both the BadNet and the LC attack. We use a step size of 0.4 on CIFAR-100 and Tiny-ImageNet to maintain the model performance on C-Acc.

RA Configurations In this section, we provide the detailed experimental setting of the RA in our revisiting Section 3. For the CIFAR-10, we adopt 5 samples for BadNet, 1 samples for Blended, 10 samples for SSBA, and 2 samples for LC attack to perform RA. For both the CIFAR-100 and Tiny-ImageNet, we increase the number of poisoned images to 25 samples for BadNet, 10 samples for Blended, and 35 samples for SSBA to perform RA. We set the learning rate to 0.01 for the CIFAR-10 and 0.001 for the CIFAR-100 and Tiny-ImageNet to maintain C-Acc. However, directly fine-tuning with these poisoned samples would negatively impact the performance of C-Acc. Therefore, in addition to the backdoored samples, we introduce extra benign samples during the fine-tuning process. We fix the size of the tuning images to 2% of the overall training dataset, which are 1000 samples for CIFAR-10 and CIFAR-100, and 2000 samples for Tiny-ImageNet.

Note that there are primarily two reasons for selecting only a limited number of poisoned examples for fine-tuning during our experiments. Firstly, using a larger number of poisoned samples, such as 10 BadNet samples on CIFAR-10, would significantly increase the ASR of a clean model. This would undermine the reliability of our RA evaluation conducted on the purified models. Secondly, by utilizing only a few poisoned examples, we intentionally expose the purified models to a potential compromise, even if they exhibit a seemingly low ASR. This approach effectively highlights the vulnerability of these purified models, emphasizing their susceptibility to attacks despite achieving a seemingly low ASR.

QRA Configurations We employ a three-layer Multilayer Perceptron (MLP) to generate the reversed perturbation in the input space. For simplicity, the number of neurons in all internal layers is fixed at 1024, followed by the Rectified Linear Unit (ReLU) activation function. To train the generator, we

[2] https://github.com/AISafety-HKUST/stable_backdoor_purification

[3] https://github.com/xuxiong0214/BTIDBF

Table 3: The post-purification robustness performance against diverse defense methods. We evaluate the performance on CIFAR-10 with ResNet-18 and set the overall poisoning rate to 5%. The *O-Backdoor* indicates the original performance of backdoor attacks, *O-Robustness* metric represents the purification performance of the defense method, and the *P-Robustness* metric denotes the post robustness after applying RA. All metrics are measured in percentage (%).

Method	Mode	BadNet		Blended		SSBA		LC	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
Attack	O-Backdoor	94.04	99.99	94.77	100.0	94.60	96.38	94.86	99.99
Clean	O-Robustness	95.07	0.57	95.07	0.90	95.07	0.61	95.07	0.81
	P-Robustness	94.37	0.20	95.02	8.12	94.29	1.26	94.93	1.33
EP	O-Robustness	93.57	0.94	93.47	3.37	93.39	0.53	93.64	0.52
	P-Robustness	92.27	1.08	93.36	4.93	92.05	2.47	93.69	0.41
ANP	O-Robustness	93.75	4.24	94.56	0.30	93.22	0.44	93.40	0.92
	P-Robustness	93.76	100	94.42	54.93	93.99	82.74	94.26	97.92
BAU	O-Robustness	92.63	1.61	93.47	5.77	93.41	1.63	92.80	2.49
	P-Robustness	93.50	100	94.17	100	93.93	93.99	94.38	99.29
SAM	O-Robustness	92.85	2.14	93.14	2.01	93.27	4.80	93.00	3.14
	P-Robustness	92.89	100	93.05	78.93	92.98	72.20	92.57	97.91
FST	O-Robustness	93.89	0.78	93.92	0.81	94.01	0.63	94.09	0.41
	P-Robustness	93.92	100	93.88	99.87	93.91	85.53	94.16	90.36
BTI	O-Robustness	92.56	1.08	92.48	0.00	92.21	2.46	93.28	2.64
	P-Robustness	91.15	100	91.86	99.8	91.18	68.57	92.89	99.97
PAM (Ours)	O-Robustness	92.11	1.14	93.34	1.67	92.96	1.24	92.32	4.92
	P-Robustness	91.66	3.90	93.38	2.69	92.20	3.31	92.15	8.31

use 500 benign examples and 500 backdoored examples from the CIFAR-10 dataset. Our training process is conducted over 50 epochs with a constant learning rate of 0.1. We set the α to 0.1 for the LC attack and 0.2 for the others to achieve a high P-ASR on purified models while simultaneously reducing the attack performance on clean models. During the training process of the perturbation generator, we begin by flattening the input image into a one-dimensional vector before feeding it to the generator. Once we obtain the output from the generator, we reshape it back to the original input size. We then multiply it with the pre-defined budget ϵ before we integrate it into the images. In our implementation, we fix the ϵ to $16/255$ for all experiments.

C Additional Experiments

C.1 Additional Results of RA

Detailed RA performance Our previous experiments in Section 3 only provide the average metrics on CIFAR-10. Therefore, in this section, we provide detailed RA results against each backdoor attack. Specifically, we demonstrate the detailed experimental results of ResNet-18 with 5% poisoning rate in Table 3; ResNet-18 with 10% poisoning rate in Table 4; ResNet-50 with 5% poisoning rate in Table 5 and DenseNet-161 with 5% poisoning rate in Table 6, respectively. Based on these experimental results, our PAM demonstrates effective post-purification robustness against diverse architectures and poisoning rates, which leads to only a small ASR increase (less than 3% on average) after performing RA.

Evaluation of RA under Lower Poisoning Rate In addition to the 5% and 10% poisoning rates used in our previous evaluation, we also include a lower poisoning rate of 1% to assess the performance of RA. Specifically, we experiment on CIFAR-10 and utilize the ResNet-18 for a fair comparison. We exclude the LC attack from our experiments since most defense methods struggle to adequately purify the backdoor behavior. As shown in Table 7, our PAM achieves a tiny increase of the ASR after RA compared to other purification techniques. This further demonstrates the effectiveness of our PAM method in enhancing post-purification robustness.

Evaluation of RA under Other Datasets In this section, we provide additional evaluations on RA against two other datasets. We present our experimental results in Table 8 and Table 9 for the

Table 4: The post-purification robustness performance against diverse defense methods. We evaluate the performance on CIFAR-10 with ResNet-18 and set the overall poisoning rate to 10%. The *O-Backdoor* indicates the original performance of backdoor attacks, *O-Robustness* metric represents the purification performance of the defense method, and the *P-Robustness* metric denotes the post robustness after applying RA. All metrics are measured in percentage (%).

Method	Mode	BadNet		Blended		SSBA		LC	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
Attack	O-Backdoor	93.73	100.0	94.28	100.0	94.31	98.71	85.80	100.0
Clean	O-Robustness	95.07	0.57	95.07	0.90	95.07	0.61	95.07	0.81
	P-Robustness	94.37	0.20	95.02	8.12	94.29	1.26	94.93	1.33
EP	O-Robustness	92.78	0.93	93.34	1.52	93.10	0.71	92.12	0.16
	P-Robustness	92.17	1.58	93.06	4.78	92.04	3.70	91.79	2.03
ANP	O-Robustness	93.29	0.60	94.20	0.07	93.90	2.14	84.72	0.00
	P-Robustness	92.74	100.0	93.77	76.96	93.42	91.92	92.00	91.41
BAU	O-Robustness	91.85	0.82	92.76	5.20	91.99	6.20	91.80	3.89
	P-Robustness	92.96	100.0	93.19	99.99	93.38	97.32	91.87	99.49
SAM	O-Robustness	92.63	1.50	92.64	0.79	91.70	4.01	91.81	4.78
	P-Robustness	92.40	100	92.73	32.16	91.39	81.59	91.73	98.91
FST	O-Robustness	93.41	0.07	93.81	0.08	93.86	0.08	92.43	3.44
	P-Robustness	93.38	100.0	93.77	99.96	93.53	93.91	92.27	73.68
BTI	O-Robustness	92.36	1.69	93.17	0.67	93.16	2.09	91.84	1.59
	P-Robustness	91.14	100	92.44	83.43	92.48	93.82	91.94	91.93
PAM (Ours)	O-Robustness	92.43	1.73	92.63	0.22	92.89	1.37	91.06	2.31
	P-Robustness	91.77	1.47	91.85	7.44	92.01	2.63	90.98	3.37

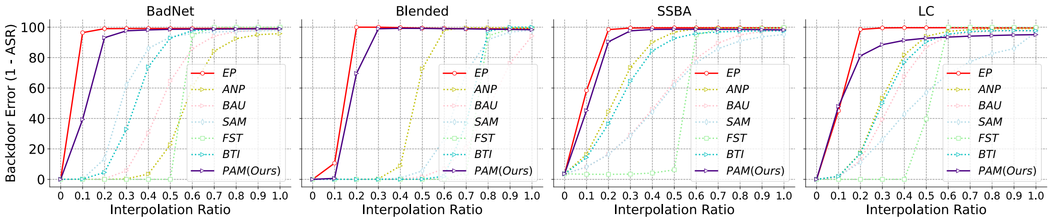

Figure 7: The experimental results of LMC on CIFAR-10. We evaluate the performance on ResNet-18 and set the poisoning rate to 5%.

CIFAR-100 and Tiny-ImageNet, respectively. It is worth noting that our PAM method remains effective when applied to these two datasets, resulting in only a minor increase in ASR.

C.2 Detailed Results of LMC

Detailed LMC performance In addition to the average LMC on CIFAR-10, we provide detailed experimental results for each attack setting. We demonstrate the LMC against ResNet-18 with 5% poisoning rate in Figure 7; ResNet-18 with 10% poisoning rate in Figure 8; ResNet-50 with 5%

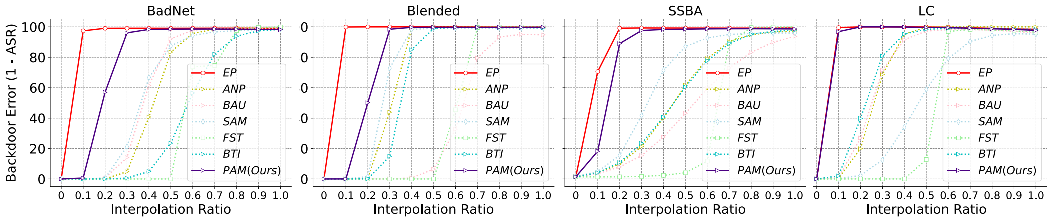

Figure 8: The experimental results of LMC on CIFAR-10. We evaluate the performance on ResNet-18 and set the poisoning rate to 10%.

Table 5: The post-purification robustness performance against diverse defense methods. We evaluate the performance on CIFAR-10 with ResNet-50 and set the overall poisoning rate to 5%. The *O-Backdoor* indicates the original performance of backdoor attacks, *O-Robustness* metric represents the purification performance of the defense method, and the *P-Robustness* metric denotes the post robustness after applying RA. All metrics are measured in percentage (%).

Method	Mode	BadNet		Blended		SSBA		LC	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
Attack	O-Backdoor	93.81	99.92	94.53	100.0	93.65	97.70	94.57	99.90
Clean	O-Robustness	94.6	0.72	94.6	0.19	94.6	0.77	94.60	0.80
	P-Robustness	94.06	4.10	94.25	3.58	93.00	3.03	94.29	4.44
EP	O-Robustness	92.84	0.98	92.10	1.12	91.84	0.43	92.91	0.41
	P-Robustness	92.33	1.48	91.36	3.51	89.26	6.28	92.33	1.71
ANP	O-Robustness	92.26	1.01	94.51	0.30	92.46	2.08	92.05	6.32
	P-Robustness	93.44	92.02	94.43	76.51	93.01	85.51	93.67	81.14
BAU	O-Robustness	92.17	1.23	92.88	3.82	90.57	6.64	92.63	3.91
	P-Robustness	92.85	17.94	93.7	98.49	92.28	92.94	93.14	23.73
SAM	O-Robustness	91.53	2.69	92.27	4.86	91.77	2.56	91.82	3.73
	P-Robustness	91.39	66.87	92.28	69.04	91.43	75.49	91.83	17.96
FST	O-Robustness	92.46	0.38	93.79	0.18	93.09	3.52	92.83	1.62
	P-Robustness	93.12	85.08	93.53	99.37	92.81	81.04	92.97	56.07
BTI	O-Robustness	92.27	4.67	92.15	0.94	92.12	1.39	92.42	3.20
	P-Robustness	92.37	100.0	91.74	99.70	92.05	68.63	91.77	89.93
PAM (Ours)	O-Robustness	92.58	0.94	92.59	0.24	92.36	1.62	91.99	2.14
	P-Robustness	91.49	1.46	92.75	0.64	92.32	14.23	90.68	3.54

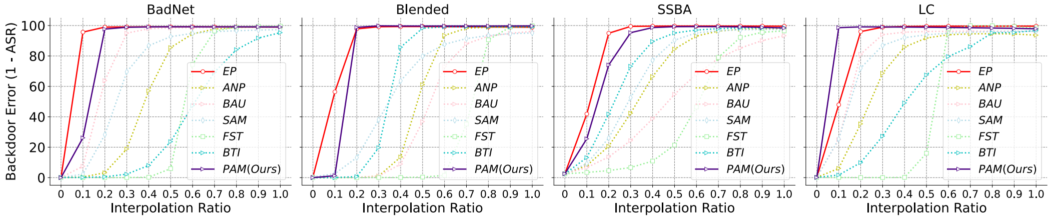

Figure 9: The experimental results of LMC on CIFAR-10. We evaluate the performance on ResNet-50 and set the poisoning rate to 5%.

poisoning rate in Figure 9 and DenseNet-161 with 5% poisoning rate in Figure 10, respectively. Our experiments demonstrate that the proposed PAM method effectively introduces high error barriers along the backdoor-connected path, leading to a greater deviation from the backdoored models.

LMC under Lower Poisoning Rate In this section, we evaluate the LMC under a lower poisoning rate, namely 1%. As shown in Figure 11, our PAM still achieves stable high error barriers along the backdoor-connected path compared to other defense methods.

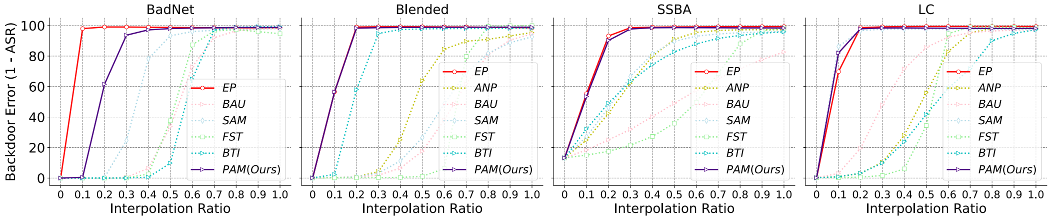

Figure 10: The experimental results of LMC on CIFAR-10. We evaluate the performance on DenseNet-161 and set the poisoning rate to 5%.

Table 6: The post-purification robustness performance against diverse defense methods. We evaluate the performance on CIFAR-10 with DenseNet-161 and set the overall poisoning rate to 5%. We omit the performance of ANP against BadNet since it can not achieve successful backdoor purification. The *O-Backdoor* indicates the original performance of backdoor attacks, *O-Robustness* metric represents the purification performance of the defense method, and the *P-Robustness* metric denotes the post robustness after applying RA. All metrics are measured in percentage (%).

Method	Mode	BadNet		Blended		SSBA		LC	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
Attack	O-Backdoor	89.85	100.0	89.59	98.72	88.83	86.75	90.13	99.80
Clean	O-Robustness	90.17	1.27	90.17	1.14	90.17	1.44	90.17	2.17
	P-Robustness	89.54	2.11	89.47	1.27	88.24	4.64	89.41	4.44
EP	O-Robustness	88.58	1.52	88.00	3.21	88.17	0.69	88.59	0.56
	P-Robustness	88.03	2.77	87.13	3.54	86.33	1.47	88.94	0.95
ANP	O-Robustness	-	-	89.18	4.56	88.64	2.74	90.04	1.58
	P-Robustness	-	-	89.06	93.72	88.65	47.47	89.45	71.10
BAU	O-Robustness	88.00	1.81	87.62	5.43	86.91	17.34	88.02	2.63
	P-Robustness	88.52	91.40	88.75	90.40	87.96	78.77	88.84	82.47
SAM	O-Robustness	86.91	2.00	86.96	7.38	86.64	3.92	87.79	2.52
	P-Robustness	86.79	100.0	87.09	78.21	86.98	35.97	87.30	12.23
FST	O-Robustness	88.33	5.37	88.56	0.18	88.32	1.63	87.40	1.57
	P-Robustness	88.16	99.86	87.92	83.06	87.52	68.82	87.47	72.61
BTI	O-Robustness	89.01	0.77	88.22	1.80	88.30	0.53	87.93	2.71
	P-Robustness	88.11	97.73	88.35	26.20	87.43	48.47	87.67	100.0
PAM (Ours)	O-Robustness	88.70	1.22	87.02	1.08	87.62	1.61	87.70	1.81
	P-Robustness	87.03	3.04	86.49	1.94	85.89	3.72	86.44	9.39

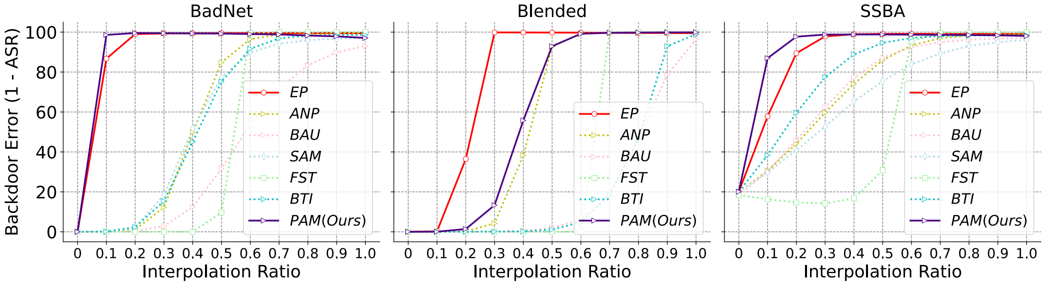

Figure 11: The experimental results of LMC on CIFAR-10. We evaluate the performance on ResNet-18 and set the poisoning rate to 1%.

LMC under Other Datasets We further evaluate the LMC on other two datasets, including the CIFAR-100 (shown in Figure 12) and the Tiny-ImageNet (shown in Figure 13). Observations through the experimental results suggest that our PAM is also effective in introducing high error barriers along the backdoor-connected path across diverse datasets.

LMC with Clean Samples In addition to evaluating the LMC with backdoored examples, we also provide experimental results of the LMC with clean samples, which we refer to as the clean-connected path. As depicted in Figure 14, we observe that diverse purification techniques exhibit almost no clean error barrier along the clean-connected path.

C.3 Additional Results of QRA

In this section, we evaluate the QRA under various attack configurations on the CIFAR-10 dataset. Specifically, we include additional poisoning rates (1% and 10%) in Figure 15, and two model architectures (ResNet-50 and DenseNet-161) in Figure 16. We report the performance against two attacks, namely the BadNet and Blended. Note that for certain defense methods, their ASR after applying RA is low, making it hard to perform QRA evaluations on them. Therefore, we exclude these defense methods from our evaluation and represent them as light gray in the Figure. Experimental

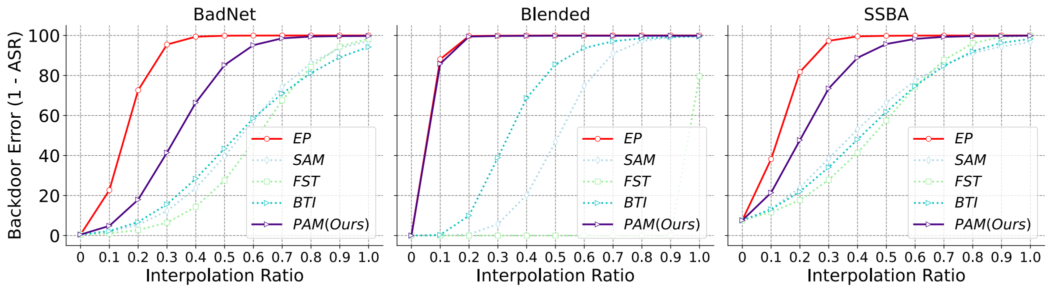

Figure 12: The experimental results of LMC on CIFAR-100. We evaluate the performance on ResNet-18 and set the poisoning rate to 5%.

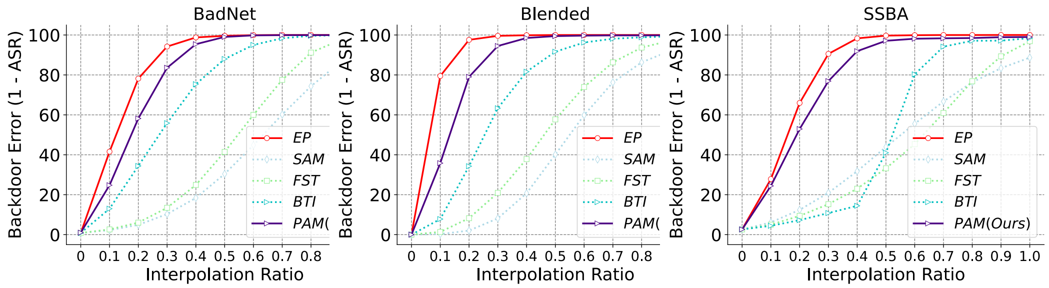

Figure 13: The experimental results of LMC on Tiny-ImageNet. We evaluate the performance on ResNet-18 and set the poisoning rate to 5%.

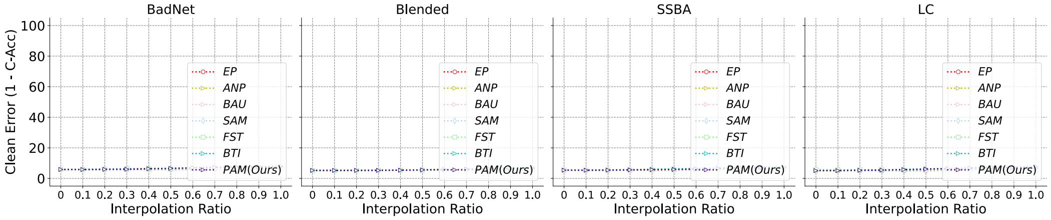

Figure 14: The experimental results of the clean-connected path on CIFAR-10. We evaluate the performance on ResNet-18 and set the poisoning rate to 5%.

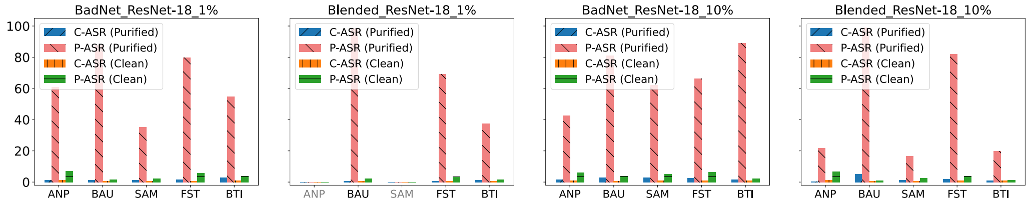

Figure 15: Experimental results of QRA against two poisoning rates (5% and 10%).

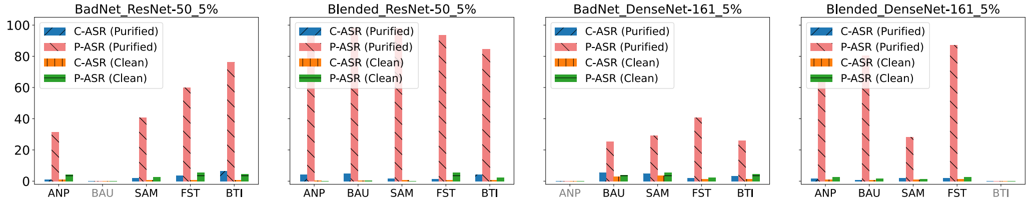

Figure 16: Experimental results of QRA against two model architectures: ResNet-50 and DenseNet-161.

Table 7: The post-purification robustness performance against diverse defense methods. We evaluate the performance on CIFAR-10 with ResNet-18 and set a lower poisoning rate to 1%. We omit the performance of SAM against Blended since it can not achieve successful backdoor purification. The *O-Backdoor* indicates the original performance of backdoor attacks, *O-Robustness* metric represents the purification performance of the defense method, and the *P-Robustness* metric denotes the post robustness after applying RA. All metrics are measured in percentage (%).

Method	Mode	BadNet		Blended		SSBA	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
Attack	O-Backdoor	94.77	100.0	94.90	99.12	94.94	79.98
Clean	O-Robustness	95.07	0.57	95.07	0.90	95.07	0.61
	P-Robustness	94.63	1.50	94.90	2.77	94.47	0.98
EP	O-Robustness	94.36	0.54	93.22	0.52	94.16	0.95
	P-Robustness	93.36	0.68	93.36	0.51	92.86	0.80
ANP	O-Robustness	93.41	0.60	93.84	0.29	94.61	1.13
	P-Robustness	94.32	87.11	94.49	0.82	94.37	48.21
BAU	O-Robustness	92.64	6.89	92.64	3.67	93.18	2.81
	P-Robustness	93.88	95.67	93.92	99.91	94.14	58.04
SAM	O-Robustness	93.45	2.80	-	-	92.68	3.77
	P-Robustness	93.64	41.72	-	-	92.57	55.51
FST	O-Robustness	94.47	1.48	94.09	0.00	94.20	0.13
	P-Robustness	93.53	90.07	93.90	99.90	93.86	80.23
BTI	O-Robustness	92.89	1.44	92.82	1.06	92.90	1.52
	P-Robustness	92.33	99.98	92.50	95.21	91.92	36.84
PAM (Ours)	O-Robustness	92.52	2.91	92.6	0.10	92.71	1.91
	P-Robustness	92.61	1.74	92.91	0.47	91.91	3.45

Table 8: The post-purification robustness performance against diverse defense methods. We evaluate the performance on CIFAR-100 with the pretrained ResNet-18 and set the poisoning rate to 5%. The *O-Backdoor* indicates the original performance of backdoor attacks, *O-Robustness* metric represents the purification performance of the defense method, and the *P-Robustness* metric denotes the post robustness after applying RA. All metrics are measured in percentage (%).

Method	Mode	BadNet		Blended		SSBA	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
Attack	O-Backdoor	78.91	99.51	78.98	100.0	78.59	92.38
Clean	O-Robustness	79.70	0.04	79.70	0.03	79.70	0.04
	P-Robustness	78.92	0.25	79.03	0.39	78.30	1.91
EP	O-Robustness	76.89	0.04	76.95	0.04	76.49	0.05
	P-Robustness	76.82	0.10	76.18	0.07	76.12	1.32
SAM	O-Robustness	76.27	2.85	76.54	0.44	76.34	3.30
	P-Robustness	76.63	95.30	77.16	97.90	75.50	78.35
FST	O-Robustness	73.02	1.71	72.41	0.30	73.54	0.18
	P-Robustness	72.93	93.71	72.28	100.0	72.06	59.90
BTI	O-Robustness	75.55	5.79	75.68	0.49	75.59	1.77
	P-Robustness	75.78	59.29	76.23	70.44	75.07	46.10
PAM (Ours)	O-Robustness	74.97	0.29	76.64	0.19	75.07	0.14
	P-Robustness	75.72	0.76	76.12	0.13	74.75	1.95

results demonstrate the effectiveness of our QRA on waking backdoor behavior under various attack settings.

C.4 Results of Analyzing Sensitivity of ρ

We present the model performance across various ρ values in Table 10. Our observations reveal that as ρ rises, there is a slight decrease in clean accuracy alongside a significant enhancement in robustness against the RA. Additionally, we note that performance is relatively insensitive to ρ when it exceeds 0.3. Given that we primarily monitor C-Acc (with the validation set) in practice, we aim to

Table 9: The post-purification robustness performance against diverse defense methods. We evaluate the performance on Tiny-ImageNet with the pretrained ResNet-18 and set the poisoning rate to 5%. The *O-Backdoor* indicates the original performance of backdoor attacks, *O-Robustness* metric represents the purification performance of the defense method, and the *P-Robustness* metric denotes the post robustness after applying RA. All metrics are measured in percentage (%).

Method	Mode	BadNet		Blended		SSBA	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
Attack	O-Backdoor	72.60	99.01	73.68	99.99	73.02	97.05
Clean	O-Robustness	73.88	0.08	73.88	0.10	73.88	0.05
	P-Robustness	72.99	0.76	73.48	0.60	72.10	2.72
EP	O-Robustness	70.90	0.02	71.11	0.01	70.24	0.01
	P-Robustness	70.59	0.65	70.93	0.09	69.98	1.90
SAM	O-Robustness	70.65	5.92	70.80	4.02	70.18	11.22
	P-Robustness	70.90	84.64	71.30	99.41	70.61	77.85
FST	O-Robustness	65.36	0.63	66.55	1.19	65.85	3.32
	P-Robustness	65.30	93.49	65.58	60.12	64.77	77.58
BTI	O-Robustness	68.43	0.06	68.74	0.55	68.92	1.59
	P-Robustness	68.91	68.29	68.41	38.13	67.98	77.25
PAM (Ours)	O-Robustness	68.78	0.06	67.89	0.14	68.01	4.47
	P-Robustness	68.97	7.81	66.86	0.37	67.39	14.26

Table 10: We demonstrate the performance of PAM with diverse ρ and evaluate the Blended attack on CIFAR-10 with ResNet-18. The O-Robustness metric represents the purification performance of the defense method, and the P-Robustness metric denotes the post robustness after applying RA.

Evaluation Mode	$\rho = 0.1$ (C-Acc / ASR)	$\rho = 0.3$ (C-Acc / ASR)	$\rho = 0.5$ (C-Acc / ASR)	$\rho = 0.7$ (C-Acc / ASR)	$\rho = 0.9$ (C-Acc / ASR)
O-Robustness	94.03 / 6.33	93.64 / 2.07	93.34 / 1.67	92.12 / 0.50	91.99 / 1.00
P-Robustness	93.60 / 33.29	93.61 / 10.06	93.38 / 2.69	92.17 / 2.62	92.54 / 0.30

achieve a favorable trade-off between these two metrics. Therefore, we follow the approach of FST [32] and select ρ to ensure that C-Acc remains above a predefined threshold, such as 92%.

C.5 Additional Discussions on the Frequently Asked Question: Directly Applying SAM with BTI

Given that our proposed method exhibits some similarities to the functionality of SAM, a frequently asked question arises regarding whether utilizing reversed backdoor data from BTI in conjunction with SAM would enhance post-purification robustness. To address this question, we further evaluate the performance of directly integrating SAM with BTI (BTI+SAM) as a robust baseline and present a comparative analysis with PAM in Table 11. Our experimental results indicate that the direct combination of BTI and SAM does not achieve the same level of robustness as PAM. This finding underscores the significance of the backdoor-connected pathway between purified and backdoored models, as delineated by the LMC.

Table 11: The comparison of post-purification robustness performance between BTI+SAM and PAM. We evaluate the performance on CIFAR-10 with ResNet-18 and set the overall poisoning rate to 5%. The *O-Backdoor* indicates the original performance of backdoor attacks, *O-Robustness* metric represents the purification performance of the defense method, and the *P-Robustness* metric denotes the post robustness after applying RA. All metrics are measured in percentage (%).

Method	Mode	BadNet		Blended		SSBA		LC	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
Attack	O-Backdoor	94.04	99.99	94.77	100.0	94.60	96.38	94.86	99.99
BTI+SAM	O-Robustness	91.72	3.70	92.49	42.49	93.03	7.84	93.04	6.67
	P-Robustness	92.48	100.0	92.99	100.0	93.36	94.12	92.87	92.01
PAM (Ours)	O-Robustness	92.11	1.14	93.34	1.67	92.96	1.24	92.32	4.92
	P-Robustness	91.66	3.90	93.38	2.69	92.20	3.31	92.15	8.31

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Section 1

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when the image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: No theory results

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section B

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification:

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section B

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Section B

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section B

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section A

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Section B

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.