Partition-Losses Fine-Tuning: Contamination-Robust Backdoor Unlearning

Anonymous authors
Paper under double-blind review

Abstract

The widespread use of large-scale, weakly curated training data and third-party checkpoints makes training convenient, but also exposes models to poisoning-based backdoor attacks. These attacks embed a hidden trigger-to-target association during training: the infected model behaves normally on clean inputs but predicts an attacker-chosen label whenever the trigger appears, posing risks for security-sensitive deployment and model reuse. Post-training fine-tuning has become a practical default defense as it is computationally efficient and does not require control over the original training pipeline. However, most existing post-training fine-tuning defenses rely exclusively on a clean dataset and ignore suspicious inputs, leaving potentially useful signals unexploited. In this paper, we propose Partition-Losses Fine-Tuning (PL), a simple and architecture-agnostic post-training method that leverages a defender-constructed candidate suspicious pool, either synthesized from publicly known trigger templates or flagged by off-the-shelf detectors. PL minimizes the benign loss on clean data while maximizing the loss on these candidate suspicious samples, directly breaking the trigger-to-target association. Importantly, PL remains effective even when the synthesized trigger pattern differs from that of the true attack or the target class is unknown. Comprehensive experiments show that PL matches or surpasses clean-only fine-tuning methods under the same computational budget while substantially reducing the amount of clean data required. It also remains effective under realistic contamination, hyperparameter variation, and cross-attack settings.

1 Introduction

Deep neural networks (DNNs) have achieved state-of-the-art performance across diverse tasks (Canziani et al., 2016; He et al., 2016; Devlin et al., 2019; Liu et al., 2020; Li et al., 2024). This success is largely driven by the use of large-scale, weakly curated training data and third-party pretraining (Devlin et al., 2019; Radford et al., 2021; Dosovitskiy et al., 2021). At the same time, this reliance introduces security risks: uncurated data can be poisoned, and pretrained checkpoints may already contain hidden vulnerabilities. Poisoning-based backdoor attacks inject a hidden trigger-to-target mapping through the training data. This enables models to behave normally on clean inputs while misclassifying triggered inputs into an attacker-chosen label (Goldblum et al., 2022; Shejwalkar et al., 2022; Carlini et al., 2024). Because the model's accuracy on clean data remains high, these attacks are difficult to detect with standard validation, posing significant challenges to security-sensitive deployment and model reuse (Tran et al., 2018; Wang et al., 2019).

Various defenses have been proposed targeting different stages of the pipeline. Pre-training defenses aim to filter suspected poisoned samples or neutralize poisoned samples, for example, through anomaly detection, trigger localization before training (Tran et al., 2018; Doan et al., 2020). In-training defenses intervene during optimization to inhibit backdoor injection (Li et al., 2021a; Huang et al., 2022). However, both pre-training and in-training defenses require access to and control over the original training pipeline, which is often impractical for large-scale models due to the cost and infrastructure required for training from scratch (Brown et al., 2020; Radford et al., 2021; Dosovitskiy et al., 2021). In many realistic deployment scenarios, practitioners instead rely on fine-tuning third-party pretrained models, making post-training defenses the only

feasible line of defense. Consequently, post-training defenses have emerged as a practical default: they operate directly on an already-trained (and potentially infected) model, require minimal prior knowledge of the training process, are broadly compatible across architectures, and incur relatively low computational overhead (Wu & Wang, 2021; Min et al., 2023).

Most existing post-training defenses rely primarily on a small, trusted clean dataset while avoiding or discarding suspicious inputs (Liu et al., 2018; Li et al., 2021b; Wu & Wang, 2021). Such methods implicitly treat potentially triggered samples as harmful noise rather than as a source of defensive signal, and are therefore constrained to optimizing benign performance alone. As a result, these approaches do not directly address the trigger-to-target association. This limitation is structural: optimizing benign loss on clean samples does not explicitly disrupt the backdoor mechanism and can struggle to substantially reduce attack success rate. A notable recent exception is feature-shift tuning (Min et al., 2023), which introduces an additional regularization term to push the fine-tuned model away from the original backdoored one. In contrast, we address this limitation by explicitly leveraging suspicious inputs to directly weaken the trigger-to-target association, a strategy that is complementary to prior post-training defenses and can be readily combined with them.

Threat Model. We consider a post-training defense setting in which the defender is given a pretrained model f_θ that may contain a backdoor, without access to the original training pipeline or training data. The defender has access to a small clean dataset $\mathcal{D}_b$ from which a candidate suspicious pool $\mathcal{D}_s$ can be constructed, without prior knowledge of the true trigger pattern or target class.

In practice, this pool can be synthesized by applying known poisoning mechanisms to the clean data, without requiring knowledge of the true attack configuration (Gu et al., 2019; Li et al., 2021c; Nguyen & Tran, 2021). Alternatively, suspicious samples may be obtained from untrusted data streams, model-monitoring systems, or incident-response pipelines that flag anomalous behavior (Guo et al., 2020; Hu et al., 2024; Xian et al., 2023; He et al., 2025). Although these samples may be noisy, imperfect, or mismatched to the true attack, they can still provide useful signals for backdoor mitigation.

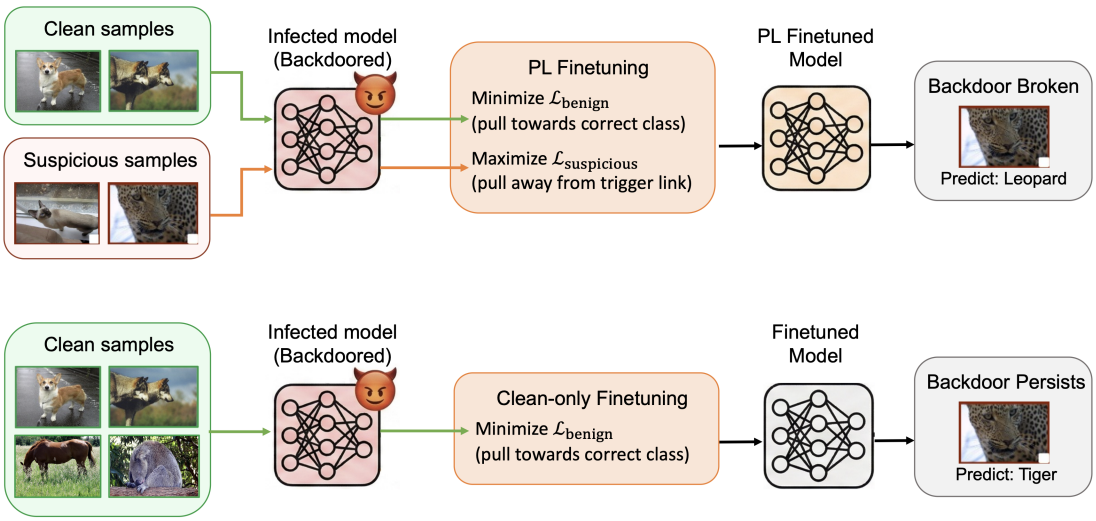

Figure 1: Overview of Partition-Losses Fine-Tuning (PL) (top) compared to clean-only fine-tuning (bottom). Starting from an infected model, PL leverages both clean and candidate triggered samples by minimizing benign loss while maximizing suspicious loss to weaken the trigger-to-target association, thereby suppressing the backdoor. In contrast, clean-only fine-tuning optimizes benign loss alone and may leave the backdoor intact. PL does not require additional data: it reallocates the same fine-tuning budget to explicitly incorporate flagged suspicious samples.

We propose **Partition-Losses Fine-Tuning (PL)**, a simple loss-level modification that explicitly exploits this setting. PL jointly minimizes benign loss on $\mathcal{D}_b$ while maximizing the target-class loss on $\mathcal{D}_s$, directly weakening the trigger-to-target association (see Figure 1). To summarize, our contributions are:

1. **A realistic post-training defense setting.** We formalize a post-training defense scenario in which the defender operates on a pretrained model that may be backdoored and has no access to the original training pipeline. The defender has access to a small mostly benign pool $\mathcal{D}_b$, as in standard fine-tuning based defenses. Crucially, the defender constructs a candidate suspicious pool $\mathcal{D}_s$ rather than relying on externally provided triggered samples. For instance, $\mathcal{D}_s$ can be synthesized by applying public trigger patterns to clean data, or flagged by off-the-shelf detectors. Both pools are allowed to be imperfect, with explicit false-positive and false-negative contamination rates. This setting captures practical deployment without assuming a perfect oracle, and we analyze it both theoretically and empirically.

2. **Partition-Losses Fine-Tuning (PL): a new post-training objective.** We propose Partition-Losses Fine-Tuning (PL), a simple loss-level modification that jointly minimizes the benign loss on $\mathcal{D}_b$ while maximizing the target-class loss on $\mathcal{D}_s$. Unlike prior clean-only fine-tuning defenses, PL explicitly leverages defender-constructed suspicious inputs as a defensive signal to directly weaken the trigger-to-target association without requiring any actual triggered samples from training. We provide a theoretical analysis showing that, under mild conditions on pool contamination, PL provably reduces the surrogate attack success rate more effectively than clean-only fine-tuning.

3. **Robustness to limited data, contamination, and attack mismatch.** Through extensive experiments on multiple datasets and five representative backdoor attacks, we show that PL consistently matches or outperforms state-of-the-art post-training defenses while using 2–2.5$\times$ fewer clean samples. PL remains effective under nonzero contamination in both tuning pools, is stable across a wide range of trade-off parameters, and generalizes across attacks. In particular, PL maintains strong performance even when the suspicious samples used for fine-tuning are generated by a different attack than the one used at test time, demonstrating robustness under cross-attack settings.

4. **Practicality and compatibility.** PL requires no architectural changes, no trigger knowledge, no ground-truth triggered samples, and no access to the original training data. It can be readily combined with existing post-training defenses, such as regularization-based fine-tuning methods. These properties make PL a lightweight and broadly applicable plug-in defense for real-world deployment.

2 Related Work

Backdoor attacks. Backdoor attacks preserve a model's intended behavior while injecting an additional association between a trigger and an adversarial behavior. In targeted attacks, any input stamped with the trigger is mapped to a fixed class. This behavior is learned from a poisoned dataset, where the attacker embeds a trigger in a small fraction of training inputs and relabels them to the target class (Chen et al., 2017; Goldblum et al., 2022; Gu et al., 2019; Li et al., 2021c; Turner et al., 2019; Carlini & Terzis, 2022). Despite growing interest in language models (Chen et al., 2021; Cui et al., 2022; Pan et al., 2022; Liu et al., 2022; He et al., 2025), computer vision is the predominant field of backdoor mechanisms and benchmarks (Wu et al., 2022). In this paper, we focus on image classification, though the proposed defense strategy is agnostic to input modality and model architecture. Evaluation on other tasks is deferred to future work.

Backdoor defenses. Existing defenses span three stages of the pipeline. Pre-training defenses aim to detect or neutralize poisoned samples before training through anomaly detection or trigger localization (Doan et al., 2020; Liu et al., 2017; Tran et al., 2018). In-training defenses intervene during optimization via regularization or representation learning-based screening to inhibit backdoor injection as the model learns (Li et al., 2021a; Huang et al., 2022). These two families require full access to and modification of the training pipelines, which are computationally expensive. In contrast, post-training defenses operate on the trained (infected) model. General strategies include trigger inversion (Wang et al., 2019; 2022; 2023; Xu et al., 2024), neuron or channel pruning (Liu et al., 2018; Wu & Wang, 2021; Zhu et al., 2023), and fine-tuning (Liu et al., 2018; Zheng et al., 2022; Min et al., 2023; Zeng et al., 2022; Zhu et al., 2023). While inversion and pruning can reduce backdoors, they often degrade benign accuracy and transfer poorly across architectures (Min et al., 2023).Therefore, fine-tuning has become a practical default due to its minimal assumptions and broad compatibility. However, most existing post-training fine-tuning defenses rely exclusively on a small trusted clean set, leaving suspicious inputs unused. As a result, they are constrained to optimizing only benign

loss and do not explicitly weaken the trigger-to-target association learned during poisoning. In contrast, our method directly incorporates a defender-constructed candidate suspicious pool $\mathcal{D}_s$ into the optimization objective, in which triggered samples are synthesized through public trigger patterns or optionally surfaced by detectors. This allows PL to disrupt the backdoor mechanism without relying on externally provided poisoned samples.

3 Partition-Losses Fine-Tuning (PL)

We consider an infected model f_θ, where θ denotes the model parameters, and assume the defender has standard fine-tuning access. Crucially, the defender can run forward or backward passes and update θ. The defender has a small, mostly benign pool $\mathcal{D}_b$. In addition, the defender constructs a candidate suspicious pool $\mathcal{D}_s$. Such triggered samples can be obtained either by applying publicly known trigger templates to clean data, or by flagging suspicious inputs with off-the-shelf detectors. Let $\mathcal{P}_{\text{clean}}$ and $\mathcal{P}_{\text{trigger}}$ denote the distribution of clean samples (X, Y) and triggered samples $(\tilde{Z}, T)$ with $T \equiv t$, respectively.

Both pools may be imperfect. The suspicious pool $\mathcal{D}_s$ can include clean samples that are incorrectly flagged as suspicious samples (false positives), while the benign pool $\mathcal{D}_b$ can contain triggered samples that are not flagged (false negatives). We denote the contamination rates as follows: $\epsilon_b \in [0, 1]$ is the false negative rate in $\mathcal{D}_b$, which measures the fraction of triggered samples present in the benign pool; and $\epsilon_s \in [0, 1]$ is the false positive rate in $\mathcal{D}_s$, which measures the fraction of clean samples in the suspicious pool. Therefore, the two tuning pools with contamination can be represented as:

$$\mathcal{D}_b^{(\epsilon_b)} = (1 - \epsilon_b)\, \mathcal{P}_{\text{clean}} + \epsilon_b\, \mathcal{P}_{\text{trigger}}, \qquad \mathcal{D}_s^{(\epsilon_s)} = (1 - \epsilon_s)\, \mathcal{P}_{\text{trigger}} + \epsilon_s\, \mathcal{P}_{\text{clean}}.$$

Proposed method. In contrast to clean-only fine-tuning, which reduces the backdoor indirectly by optimizing on the clean samples, leveraging suspicious samples enables explicit unlearning signals. Specifically, for a candidate triggered input $\tilde{z}$ associated with candidate target class t, we apply gradient ascent on the target-class cross-entropy. This suppresses the target-class logits and redistributes probability mass to other classes. The backdoor association is thus weakened.

We propose Partition-Losses Fine-Tuning (PL), which jointly minimizes the cross-entropy loss on $\mathcal{D}_b$ while maximizing the loss on $\mathcal{D}_s$. Mathematically, we aim to solve the following problem:

$$\min_{\theta}\ \mathbb{E}_{(x,y)\sim\mathcal{D}_b^{(\epsilon_b)}}\left[\mathrm{CE}\left(f_\theta(x), y\right)\right] - \alpha\, \mathbb{E}_{(\tilde{z},t)\sim\mathcal{D}_s^{(\epsilon_s)}}\left[\mathrm{CE}\left(f_\theta(\tilde{z}), t\right)\right], \tag{1}$$

where $\mathrm{CE}(f_\theta(x), y) = -\log f_\theta(x)_y$ is the standard cross-entropy loss, and $f_\theta(x)_y$ is the predicted probability assigned to class y. The hyperparameter $\alpha > 0$ controls the trade-off between benign accuracy and backdoor unlearning: a smaller α prioritizes benign accuracy, while a larger α promotes backdoor forgetting at the risk of clean accuracy degradation.

Equation 1 assumes the defender knows the target class t used to construct $\mathcal{D}_s$. In practice, however, the true target class may be unknown. PL extends naturally to this setting by applying gradient ascent over all non-clean candidate labels, as described in the optimization procedure below.

PL remains computationally efficient, requiring only one additional forward pass on $\mathcal{D}_s$ per iteration, compared to clean-only fine-tuning. Furthermore, PL is complementary to other post-training defenses and can be seamlessly integrated with them, as it only modifies the fine-tuning objective.

Optimization procedure. We keep α fixed throughout fine-tuning and optimize Problem 1 using Adam (Kingma & Ba, 2015). At each iteration, we sample a benign mini-batch $\mathcal{B}_b$ and a suspicious mini-batch $\mathcal{B}_s$, both of size n. We compute the corresponding empirical benign loss

$$\mathcal{L}_{\text{Benign}} = \frac{1}{n} \sum_{(x,y)\in\mathcal{B}_b} \mathrm{CE}\big(f_\theta(x), y\big).$$

The empirical suspicious loss depends on whether the target class is known. When the defender knows the target t, we apply gradient ascent to the target class cross-entropy:

$$\mathcal{L}_{\text{Suspicious}} = \frac{1}{n} \sum_{(\tilde{z},t)\in\mathcal{B}_s} \text{CE}\big(f_\theta(\tilde{z}), t\big).$$

When t is unknown, we instead apply gradient ascent on every class except the sample's clean label y_{clean}:

$$\mathcal{L}^{\text{unknown}}_{\text{Suspicious}} = \frac{1}{n} \sum_{(\tilde{z},y_{\text{clean}})\in\mathcal{B}_s} \sum_{c\neq y_{\text{clean}}} \text{CE}\big(f_\theta(\tilde{z}), c\big).$$

The PL objective is written as $\mathcal{L}_{\text{PL}} = \mathcal{L}_{\text{Benign}} - \alpha\,\mathcal{L}_{\text{Suspicious}}$, with $\mathcal{L}_{\text{Suspicious}}$ replaced by $\mathcal{L}^{\text{unknown}}_{\text{Suspicious}}$ when t is unknown. The model parameters are updated by taking a gradient step on $\mathcal{L}_{\text{PL}}$. Algorithm 1 summarizes the procedure for the known-t case. When t is unknown, we simply substitute $\mathcal{L}_{\text{Suspicious}}$ with $\mathcal{L}^{\text{unknown}}_{\text{Suspicious}}$ in the gradient update.

Algorithm 1: Partition-Losses Fine-Tuning (PL)

Input: Infected model f_θ; benign pool $\mathcal{D}_b^{(\epsilon_b)}$; defender-constructed candidate suspicious pool $\mathcal{D}_s^{(\epsilon_s)}$; target t
Output: Purified model $f_{\theta^{(I)}}$
Parameters: $\alpha > 0$; learning rate η; tuning iterations I; batch size n
for $i = 1$ **to** I **do**
- Sample a mini-batch from benign pool $\mathcal{B}_b = \{(x_k, y_k)\}_{k=1}^n \subseteq \mathcal{D}_b^{(\epsilon_b)}$
- Calculate cross-entropy loss from the benign batch $\mathcal{L}_{\text{Benign}} \leftarrow \frac{1}{n}\sum_{k=1}^n \text{CE}\big(f_\theta(x_k), y_k\big)$
- Sample a mini-batch from the candidate suspicious pool $\mathcal{B}_s = \{(\tilde{z}_j, t)\}_{j=1}^n \subseteq \mathcal{D}_s^{(\epsilon_s)}$
- Calculate cross-entropy loss from the suspicious batch $\mathcal{L}_{\text{Suspicious}} \leftarrow \frac{1}{n}\sum_{j=1}^n \text{CE}\big(f_\theta(\tilde{z}_j), t\big)$
- Form PL objective $\mathcal{L}_{\text{PL}} \leftarrow \mathcal{L}_{\text{Benign}} - \alpha\,\mathcal{L}_{\text{Suspicious}}$
- Update model parameters $\theta^{(i+1)} \leftarrow \theta^{(i)} - \eta\,\nabla_{\theta^{(i)}}\mathcal{L}_{\text{PL}}$

return $f_{\theta^{(I)}}$

Theoretical analysis. Next, we provide theoretical support for the PL objective: we characterize sufficient conditions under which PL provably improves over clean-only fine-tuning. To isolate the effect by which suspicious samples aid backdoor unlearning, our analysis is local and surrogate-based. The proof is provided in Appendix A.

Assumption 1. *Let $f_\theta : \mathcal{X} \to \Delta^C$ be a classifier with parameters θ, where Δ^C denotes the probability simplex over C classes. Let $(X, Y) \sim \mathcal{P}_{clean}$ denote a clean input–label pair, and $(\tilde{Z}, T) \sim \mathcal{P}_{trigger}$ a triggered input paired with the fixed target label $T \equiv t$. For contamination levels $\epsilon_b, \epsilon_s \in [0, 1]$, define the (mixture) tuning distributions $\mathcal{D}_b^{\epsilon_b} = (1-\epsilon_b)\,\mathcal{P}_{clean} + \epsilon_b\,\mathcal{P}_{trigger}, \mathcal{D}_s^{\epsilon_s} = (1-\epsilon_s)\,\mathcal{P}_{trigger} + \epsilon_s\,\mathcal{P}_{clean}$. The PL objective is*

$$\mathcal{L}_{\text{PL}}(\theta) = \mathbb{E}_{(X,Y)\sim\mathcal{D}_b^{\epsilon_b}}\Big[\text{CE}\left(f_\theta(X), Y\right)\Big] - \alpha\,\mathbb{E}_{(\tilde{Z},T)\sim\mathcal{D}_s^{\epsilon_s}}\Big[\text{CE}\left(f_\theta(\tilde{Z}), t\right)\Big], \quad \alpha > 0.$$

The (true) attack success rate is $\text{ASR}(f_\theta) = \Pr_{(\tilde{Z},T)\sim\mathcal{P}_{trigger}}[\arg\max_c f_\theta(\tilde{Z})_c = t]$. Since ASR is non-differentiable, we analyze the surrogate $\widetilde{\text{ASR}}$:

$$\widetilde{\text{ASR}}(f_\theta) = \mathbb{E}_{(\tilde{Z},T)\sim\mathcal{P}_{trigger}}[\log f_\theta(\tilde{Z})_t] = -\,\mathbb{E}_{(\tilde{Z},T)\sim\mathcal{P}_{trigger}}[\text{CE}(f_\theta(\tilde{Z}), t)].$$

Theorem 1 (Robustness under imperfect pools via surrogate ASR)**.** *Suppose Assumption 1 holds. Let $g_{\text{trig}} := \mathbb{E}_{(\tilde{Z},T)\sim\mathcal{P}_{\text{trigger}}}\Big[\nabla_\theta\,\text{CE}\left(f_\theta(\tilde{Z}), t\right)\Big]$, $g_{\text{clean}} := \mathbb{E}_{(X,Y)\sim\mathcal{P}_{\text{clean}}}\Big[\nabla_\theta\,\text{CE}\left(f_\theta(X), t\right)\Big]$. Note that g_{clean} is evaluated using label t so that it captures the gradient component aligned with the target-class direction, making it directly comparable to g_{trig}. Suppose the trigger and clean gradients have comparable norms: for some $\rho \geq 1$,*

$$\tfrac{1}{\rho}\|g_{\text{clean}}\| \leq \|g_{\text{trig}}\| \leq \rho\|g_{\text{clean}}\|.$$

Then:

1. ***(Relative improvement)*** *If $\epsilon_s < \frac{1}{1+\rho^3}$, the expected stochastic gradient direction strictly decreases the surrogate attack success* $\widetilde{\mathrm{ASR}}(f_\theta)$ more than *clean-only fine-tuning, for any $\alpha > 0$.*

2. ***(Absolute decrease)*** *Moreover, a sufficient condition for $\mathcal{L}_{\mathrm{PL}}$ itself to strictly decrease the surrogate attack success is*
$$\alpha \;>\; \frac{\max\{0,\, \langle g_{\mathrm{trig}}, \nabla\mathcal{L}_b\rangle\}}{(1-\epsilon_s)\|g_{\mathrm{trig}}\|^2 + \epsilon_s\langle g_{\mathrm{trig}}, g_{\mathrm{clean}}\rangle},$$
where $\nabla\mathcal{L}_b = \mathbb{E}_{(X,Y)\sim\mathcal{D}_b^{\epsilon_b}}\big[\nabla_\theta \,\mathrm{CE}(f_\theta(X), Y)\big]$. In particular, when $\epsilon_s < \frac{1}{1+\rho^3}$, there exists $\alpha > 0$ satisfying the above inequality.

Theorem 1 formalizes the partitioning strategy. The comparable-norms assumption is mild, requiring only the triggered and clean gradients to lie within a bounded ratio ρ of each other. Under this assumption, when the candidate suspicious pool satisfies $\epsilon_s < \frac{1}{1+\rho^3}$, the expected PL update decreases the surrogate ASR more than clean-only fine-tuning for any $\alpha > 0$ (part 1). With an appropriate choice of α, PL training is guaranteed to strictly decrease the surrogate ASR (part 2).

4 Experiments

We evaluate PL under the ways how the benign pool $\mathcal{D}_b$ and the suspicious pool $\mathcal{D}_s$ are constructed in practice, progressively relaxing the idealized assumptions of Section 3. Our experiments address four questions:

Q1. How effective is PL under limited tuning budgets, compared to clean-only baselines? This is the most controlled setting where $\mathcal{D}_s$ is defender-synthesized from clean data with matched trigger and known target. (Section 4.1)

Q2. How robust is PL when the clean and suspicious pools are imperfectly separated? This relaxes the perfect-partition assumption. (Section 4.2)

Q3. How robust is PL under cross-attack generalization, when the defender's synthetic trigger does not match the true attack, or the target class is unknown? (Section 4.3)

Q4. How sensitive is PL to parameter α and the amount of fine-tuning data? (Section 4.4)

Datasets and Models. We evaluate PL on three widely used benchmark datasets in the backdoor learning literature: CIFAR-10 (Krizhevsky et al., 2009), GTSRB (Stallkamp et al., 2012), and Tiny-ImageNet (Chrabaszcz et al., 2017). Following previous works (Min et al., 2023; Huang et al., 2022; Liu et al., 2018; Nguyen & Tran, 2021; Wu et al., 2022), we use ResNet-18 (He et al., 2016) on CIFAR-10, GTSRB, and CelebA, and Swin Transformer (Liu et al., 2021) on Tiny-ImageNet and ImageNet-12.

Attack Settings. To demonstrate the defense effectiveness of PL, we consider six representative dirty-label backdoor attacks from recent works: 1) BadNet (Gu et al., 2019), 2) Blended Attack (Chen et al., 2017), 3) SSBA (Li et al., 2021c), 4) WaNet (Nguyen & Tran, 2021), 5) Adaptive Blend (Qi et al., 2023), and 6) TITIM-SIG (Lin et al., 2025). The target label is set to $t = 0$ for all attacks, and the poison rate is 15%. A more detailed experimental setup can be found in Appendix B.

Defense Baselines. We compare PL with four tuning-based defenses: vanilla fine-tuning (FT) (Min et al., 2023), fine-tuning with sharpness-aware minimization (FT+SAM) (Zhu et al., 2023), implicit backdoor adversarial unlearning (I-BAU) (Zeng et al., 2022), and feature shift tuning (FST) (Min et al., 2023). See Appendix B for detailed experimental settings.

4.1 Effectiveness under limited Clean Data

We begin the evaluation of PL with the controlled setting where the defender synthesizes $\mathcal{D}_s$ by applying publicly known trigger templates to clean data. In this analysis, we match the template to the true attack, producing a perfect partition ($\epsilon_b = \epsilon_s = 0$), which removes confounding factors such as partition noise

and attack mismatch. This setup provides a clean benchmark for evaluating PL's data efficiency against clean-only baselines. We compare PL with $\alpha = 0.2$ against four clean-only fine-tuning baselines, FST, FT, I-BAU and FT+SAM, across multiple datasets and attacks. Performance is measured using attack success rate (ASR) and clean accuracy (C-ACC). We evaluate two tuning-budget regimes: (i) PL-5: PL uses 5% of the training data as the tuning set, while clean-only baselines use 10%; (ii) PL-2: PL uses 2% of the training data as the tuning set, while clean-only baselines use 5%.

Let N be the total number of training samples and set the poison rate $\rho = 0.15$. The tuning set consists of a benign pool $D_b^{\epsilon_b}$ and a candidate suspicious pool $D_s^{\epsilon_s}$ with $\epsilon_b = \epsilon_s = 0$. Since clean-only baselines exclude all candidate triggered samples, their effective tuning size shrinks by a factor of $(1 - \rho)$. Consequently, in total tuning data, baselines use $8.5\%N$ versus $5\%N$ at PL-5, and $4.25\%N$ versus $2\%N$ at PL-2. In clean samples, baselines use approximately $2\times$ more clean data at PL-5 and $2.5\times$ more clean data at PL-2.

Despite this substantial data disadvantage, PL consistently achieves a strong safety–utility balance, matching or outperforming clean-only baselines. Under the PL-5 regime (Table 1a), PL achieves near-zero ASR (grand mean: 0.004) across all dataset-attack combinations, while maintaining competitive clean accuracy (grand mean: 0.784). This behavior is consistent across datasets. On CIFAR-10, PL achieves the lowest ASR among all baselines, with only a marginal clean-accuracy drop relative to the strongest clean-only baseline. On GTSRB, PL eliminates backdoor behavior (ASR=0) while preserving competitive clean performance. On the more challenging Tiny-ImageNet and ImageNet-12, PL similarly suppresses the backdoor without sacrificing utility. In contrast, clean-only methods exhibit a clear trade-off: FT and FT+SAM preserve clean accuracy, but leave the model highly vulnerable, whereas I-BAU and FST reduce ASR only at the cost of severe utility degradation.

The advantage of PL persists under the more restrictive PL-2 regime (Table 1b), further highlighting its data efficiency. Across all dataset-attack combinations, PL again achieves the lowest ASR (grand mean: 0.009) without compromising clean accuracy (grand mean: 0.768). On the CelebA and ImageNet-12, PL maintains near-zero ASR while preserving competitive clean accuracy, demonstrating PL's scalability to more complex classification tasks. For GTSRB, we run each configuration over 5 seeds and report the mean in the main text; standard deviation are in Table 5 of Appendix C. PL achieves zero mean ASR across all attacks, confirming its stability. PL is also compatible with existing post-training defenses; results combining PL with FST are reported in Table 6 of Appendix D.

Overall, the consistency across tuning budgets shows that PL achieves a stable safety-utility balance while using substantially less tuning data than clean-only baselines, which makes PL suitable for deployment scenarios where trusted data is scarce.

4.2 Robustness to Partition Contamination

In practice, a perfectly clean partition is rarely available: the benign pool may contain unflagged triggered samples, and a suspicious pool constructed by off-the-shelf detectors inevitably admits false positives and false negatives (Northcutt et al., 2021; Dong et al., 2021; Hayase et al., 2021). We therefore evaluate PL under partition contamination, considering three cases: contamination only in the benign pool ($\epsilon_b = 0.1, \epsilon_s = 0$), the suspicious pool ($\epsilon_b = 0, \epsilon_s = 0.1$), and both ($\epsilon_b = \epsilon_s = 0.1$).

Table 2 summarizes results on CIFAR-10, GTSRB, and Tiny-ImageNet under the PL-2 regime. The same trends extend to CelebA, and ImageNet-12; full results across all five datasets, along with the combined-defense results (FST+PL) and the PL-5 regime, are provided in Appendix E.

Contamination in the benign pool ($\epsilon_b = 0.1,\ \epsilon_s = 0$): When triggered samples leak into the benign pool, the tuning data for clean-only methods is no longer purely clean, violating the core assumption. Under this setting, clean-only baselines fail to maintain a stable safety-utility balance (Table 2a). FT+SAM preserves high accuracy but leaves the model highly vulnerable to the backdoor. I-BAU reduces ASR in some cases but suffers severe utility degradation. Moreover, the clean-only method's performance becomes dataset-dependent: FT and FST appear effective on CIFAR-10 but collapse on Tiny-ImageNet. In contrast, PL suppresses the attack across all three datasets, achieving a mean ASR of 0.046, 0.270, and 0.000 on CIFAR-10, GTSRB, and Tiny-ImageNet respectively while preserving competitive clean accuracy (0.780).

Table 1: Performance of PL compared with four baselines under perfect partition ($\epsilon_b = \epsilon_s = 0$). Results are reported across five datasets and six attacks. Each entry shows attack success rate (ASR, lower is better) and clean accuracy (C-ACC, higher is better). Bold values indicate the lowest ASR in each row. Panel (a) shows the PL-5 regime, and panel (b) shows the PL-2 regime. GTSRB results are averaged over 5 seeds; the corresponding variability statistics are reported in the appendix. FST+PL is evaluated separately in Appendix D.

		FST		FT		I-BAU		FT+SAM		PL (Ours)	
Data	Attack	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC
(a) PL-5 regime (PL uses 5%, baselines use 10%)											
CIFAR-10	Blended	0.018	0.791	0.128	0.811	0.547	0.187	0.552	0.830	**0.006**	0.795
	Adaptive Blend	0.050	0.799	0.157	0.813	0.606	0.154	0.389	0.820	**0.012**	0.785
	SSBA	0.039	0.780	0.041	0.806	0.795	0.147	0.079	0.810	**0.013**	0.774
	BadNet	0.049	0.805	0.253	0.825	0.889	0.131	0.906	0.826	**0.021**	0.774
	WaNet	0.035	0.797	0.046	0.824	1.000	0.100	0.130	0.819	**0.008**	0.771
	Mean	0.038	0.794	0.125	0.816	0.767	0.144	0.411	0.821	**0.012**	0.780
GTSRB	Blended	**0.000**	0.880	0.001	0.896	**0.000**	0.142	0.013	0.914	**0.000**	0.873
	Adaptive Blend	**0.000**	0.869	0.026	0.890	**0.000**	0.191	0.419	0.910	**0.000**	0.859
	SSBA	**0.000**	0.874	0.003	0.895	0.007	0.197	0.005	0.905	**0.000**	0.887
	BadNet	**0.000**	0.908	**0.000**	0.922	0.006	0.221	**0.000**	0.918	**0.000**	0.896
	WaNet	0.001	0.895	0.015	0.906	0.006	0.090	0.076	0.904	**0.000**	0.882
	Mean	**0.000**	0.885	0.009	0.902	0.004	0.168	0.103	0.910	**0.000**	0.879
Tiny-ImageNet	Blended	**0.000**	0.412	0.996	0.717	**0.000**	0.005	0.992	0.757	**0.000**	0.697
	Adaptive Blend	**0.000**	0.412	0.996	0.717	**0.000**	0.004	0.992	0.757	**0.000**	0.697
	SSBA	**0.000**	0.387	0.972	0.703	**0.000**	0.006	0.961	0.760	**0.000**	0.727
	BadNet	**0.000**	0.394	0.986	0.719	**0.000**	0.005	0.993	0.766	**0.000**	0.720
	WaNet	0.005	0.399	0.468	0.702	**0.000**	0.005	0.424	0.757	**0.000**	0.626
	Mean	0.001	0.401	0.884	0.712	**0.000**	0.005	0.872	0.759	**0.000**	0.693
Grand Mean (a)		0.013	0.693	0.339	0.810	0.257	0.106	0.462	0.830	**0.004**	0.784
(b) PL-2 regime (PL uses 2%, baselines use 5%)											
CIFAR-10	Blended	0.013	0.795	0.143	0.810	0.635	0.213	0.649	0.820	**0.006**	0.789
	Adaptive Blend	0.049	0.794	0.159	0.806	0.054	0.131	0.357	0.817	**0.017**	0.790
	SSBA	0.045	0.779	0.045	0.798	0.558	0.234	0.114	0.796	**0.015**	0.768
	BadNet	0.036	0.795	0.237	0.817	0.571	0.230	0.917	0.820	**0.025**	0.770
	WaNet	0.026	0.797	0.042	0.809	0.983	0.111	0.147	0.810	**0.005**	0.784
	TITIM-SIG	0.013	0.800	0.040	0.815	0.467	0.225	0.056	0.823	**0.003**	0.798
	Mean	0.030	0.793	0.111	0.809	0.545	0.191	0.373	0.814	**0.011**	0.783
Tiny-ImageNet	Blended	**0.000**	0.295	0.999	0.721	0.051	0.579	0.752	0.740	**0.000**	0.660
	Adaptive Blend	**0.000**	0.295	0.999	0.721	0.051	0.579	0.752	0.740	**0.000**	0.663
	SSBA	**0.000**	0.309	0.954	0.706	**0.000**	0.005	0.944	0.745	**0.000**	0.650
	BadNet	**0.000**	0.285	0.965	0.720	**0.000**	0.005	0.990	0.748	**0.000**	0.656
	WaNet	0.003	0.290	0.946	0.715	**0.000**	0.005	0.710	0.744	**0.000**	0.618
	TITIM-SIG	0.001	0.229	0.444	0.737	–	–	0.478	0.753	**0.000**	0.659
	Mean	0.001	0.284	0.885	0.717	0.026	0.235	0.771	0.745	**0.000**	0.651
GTSRB	Blended	**0.000**	0.863	0.018	0.886	0.005	0.184	0.120	0.906	**0.000**	0.843
	Adaptive Blend	**0.000**	0.856	0.001	0.888	**0.000**	0.171	0.002	0.909	**0.000**	0.824
	TITIM-SIG	**0.000**	0.841	**0.000**	0.858	0.001	0.128	**0.000**	0.910	**0.000**	0.826
	SSBA	**0.000**	0.865	0.004	0.889	0.001	0.253	0.004	0.900	**0.000**	0.845
	Visible	**0.000**	0.879	**0.000**	0.905	0.026	0.242	0.003	0.909	**0.000**	0.868
	WaNet	**0.000**	0.891	0.004	0.907	**0.000**	0.069	0.027	0.906	**0.000**	0.857
	Mean	**0.000**	0.866	0.004	0.889	0.006	0.175	0.026	0.907	**0.000**	0.844
CelebA	Blended	0.121	0.733	0.164	0.741	0.355	0.522	0.224	0.793	**0.060**	0.682
	Adaptive Blend	0.098	0.743	0.138	0.747	0.276	0.514	0.244	0.792	**0.035**	0.669
	TITIM-SIG	0.050	0.786	0.068	0.787	0.186	0.574	0.134	0.792	**0.035**	0.731
	SSBA	0.045	0.789	0.095	0.790	0.081	0.597	0.359	0.791	**0.034**	0.762
	Visible	0.033	0.784	0.079	0.792	0.119	0.643	0.163	0.793	**0.007**	0.615
	WaNet	0.063	0.785	0.131	0.790	0.742	0.354	0.362	0.793	**0.025**	0.702
	Mean	0.069	0.770	0.113	0.775	0.293	0.534	0.248	0.792	**0.032**	0.694
ImageNet-12	Blended	0.006	0.890	0.984	0.892	0.976	0.900	0.704	0.915	**0.000**	0.860
	Adaptive Blend	**0.000**	0.875	**0.000**	0.885	**0.000**	0.888	**0.000**	0.900	**0.000**	0.872
	TITIM-SIG	0.002	0.875	0.435	0.903	0.371	0.910	0.344	0.920	**0.000**	0.885
	SSBA	0.006	0.897	0.533	0.907	0.262	0.915	0.385	0.913	**0.000**	0.885
	Visible	**0.000**	0.880	0.931	0.890	0.918	0.905	0.887	0.912	**0.000**	0.892
	WaNet	0.004	0.893	0.009	0.880	0.011	0.908	0.014	0.902	**0.000**	0.857
	Mean	0.003	0.885	0.482	0.893	0.423	0.904	0.389	0.910	**0.000**	0.875
Grand Mean (b)		0.021	0.722	0.339	0.816	0.260	0.406	0.386	0.833	**0.009**	0.768

Contamination in the suspicious pool ($\epsilon_b = 0,\ \epsilon_s = 0.1$): When clean samples enter the suspicious pool, PL's suspicious signal is diluted. This setting does not directly violate the core assumption of any method. It just weakens the strength of the suspicious signal used for PL regularization. Despite this challenge, PL

continues to effectively suppress the attack across datasets (Table 2b), achieving near-zero mean ASRs on all three datasets while clean-only baselines exhibit high ASRs on Tiny-ImageNet in particular. PL also maintains benign performance comparable to the clean-only baselines. Again, PL preserves the strongest safety-utility tradeoff across datasets.

Contamination in both benign and suspicious pools ($\epsilon_b = 0.1,\ \epsilon_s = 0.1$): This is the most challenging scenario: clean samples contaminate the suspicious pool and triggered samples remain in the benign pool. Clean-only baselines operate on fewer and contaminated clean samples, and PL must handle conflicting signals in both pools. Despite these compounded difficulties, PL maintains mean ASR of 0.025, 0.023, and 0.000 on CIFAR-10, GTSRB, and Tiny-ImageNet respectively (Table 2c). In contrast, clean-only baselines either fail to suppress the attack or sacrifice substantial utility. Across all three contamination settings under the PL-2 regime, PL consistently achieves the lowest ASR with competitive clean accuracy, despite using less than half the tuning data of the baselines (2%N vs. 4.25%N). Extended results on CelebA and ImageNet-12, combined-defense (FST+PL) comparison, and PL-5 regimes are provided in Appendix E.

4.3 Robustness to Cross-Attack Generalization and Unknown Target Classes

In practical deployments, the defender may know neither the true trigger pattern used by the attack, nor its target class. In this section, we evaluate PL under two such variants: (i) *cross-attack generalization*, where the defender's trigger pattern differs from the true attack; and (ii) *unknown target class*, where the defender does not know which label the attack chose.

Cross-attack generalization under known target. We first assess PL's cross-attack robustness under perfect partition ($\epsilon_b = \epsilon_s = 0$) across three datasets (CIFAR-10, GTSRB, and Tiny-ImageNet), with the target label used to build $\mathcal{D}_s$ matched to the true target. In Table 3, rows indicate the attack used to synthesize $\mathcal{D}_s$ (or the attack presumed by the detector) for fine-tuning, and columns indicate the attack used for evaluation. Diagonal entries are matched attacks, and off-diagonal entries quantify cross-attack generalization. As shown in Table 3, PL substantially suppresses ASR across attack families while maintaining competitive clean accuracy, indicating that it does not require the constructed suspicious pool to use the exact trigger of the true attack. However, cross-attack transfer is not uniform across attack pairs. Some visually distinctive attacks are more challenging when $\mathcal{D}_s$ is constructed from a dissimilar trigger family. We therefore interpret these results as evidence of broad, but not unlimited, trigger-agnostic transfer.

We further evaluate this known-target cross-tuning setting under the same three contamination settings as in Section 4.2. Results in Appendix F show that PL remains reasonably stable under moderate partition noise, although ASR suppression can be weakened when the clean and suspicious pools are imperfectly separated.

Cross-attack generalization under unknown target class. The known-target setting above still assumes that the defender knows the attacker's target label, which may not always be true. We now remove this assumption. Concretely, the base model is trained under attack A with a fixed target class (an all-to-one attack with target class 0). As the defender, we generate triggered samples using attack B, which may differ from A. When the target is unknown, we treat every class other than the sample's clean label as a candidate target. Specifically, for each triggered sample $\tilde{z}$ with clean label y_{clean}, PL applies gradient ascent on $\text{CE}(f_\theta(\tilde{z}), c)$ for every $c \neq y_{\text{clean}}$, suppressing shortcut mappings to any non-clean class. Compared with the known-target setting, this is more challenging: the defensive signal is spread across $C-1$ candidate labels rather than concentrated on the true target. Nevertheless, Table 4 reports results across all five datasets. The off-diagonal entries correspond to the combination of (i) cross-attack generalization and (ii) unknown target class. Overall, PL still reduces its dependence on target-label knowledge in many cases, particularly when the suspicious-pool attack is aligned with or related to the evaluation attack.

4.4 Ablation Studies for PL

In this section, we conduct ablations on the trade-off parameter α and fine-tuning data size.

Sensitivity to α. We evaluate $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ under both perfect partition and realistic dual-contamination settings. As shown in Figure 2, both ASR and C-ACC are largely flat across α under perfect partition. ASR stays near zero and clean accuracy varies only marginally. Under contamination, the expected

Table 2: PL vs. clean-only baselines under three partition-contamination regimes on the PL-2 regime. Results are shown for CIFAR-10, GTSRB, and Tiny-ImageNet. Bold indicates lowest ASR per row. GTSRB values are means over 5 seeds; standard deviations are reported in Table 8. Full results across all five datasets, and FST+PL comparisons under both-pool contamination, are in Appendix E.

		FST		FT		I-BAU		FT+SAM		PL (Ours)	
Data	Attack	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC
(a) Benign pool contamination ($\epsilon_b = 0.1$, $\epsilon_s = 0$)											
CIFAR-10	Blended	0.225	0.787	0.336	0.809	0.344	0.264	0.885	0.818	**0.061**	0.789
	Adaptive Blend	0.245	0.797	0.339	0.808	0.562	0.159	0.879	0.811	**0.091**	0.787
	SSBA	0.075	0.769	0.063	0.801	0.694	0.160	0.221	0.794	**0.018**	0.773
	BadNet	0.205	0.793	0.349	0.815	0.808	0.232	0.957	0.823	**0.057**	0.773
	WaNet	0.092	0.798	0.128	0.812	0.999	0.102	0.507	0.802	**0.033**	0.780
	TITIM-SIG	0.087	0.799	0.110	0.807	0.466	0.273	0.192	0.814	**0.018**	0.779
	Mean	0.155	0.791	0.221	0.809	0.646	0.198	0.607	0.810	**0.046**	0.780
GTSRB	Blended	0.363	0.875	0.481	0.890	**0.000**	0.127	0.939	0.907	0.324	0.839
	Adaptive Blend	0.382	0.853	0.505	0.878	**0.000**	0.157	0.927	0.900	0.370	0.822
	SSBA	0.370	0.882	0.536	0.900	**0.013**	0.189	0.930	0.900	0.298	0.860
	BadNet	0.318	0.880	0.524	0.900	**0.091**	0.114	0.936	0.905	0.299	0.868
	WaNet	0.107	0.875	0.181	0.897	0.998	0.006	0.625	0.899	**0.058**	0.844
	Mean	0.308	0.873	0.445	0.893	**0.220**	0.119	0.871	0.902	0.270	0.847
Tiny-ImageNet	Blended	1.000	0.320	1.000	0.722	1.000	0.014	1.000	0.744	**0.000**	0.677
	Adaptive Blend	1.000	0.320	1.000	0.722	1.000	0.014	1.000	0.744	**0.000**	0.665
	SSBA	0.994	0.307	0.998	0.718	0.996	0.428	0.997	0.749	**0.000**	0.654
	BadNet	0.993	0.326	0.995	0.724	0.990	0.013	0.995	0.751	**0.000**	0.651
	TITIM-SIG	0.997	0.315	0.998	0.719	0.993	0.365	0.998	0.746	**0.000**	0.645
	WaNet	0.991	0.333	0.992	0.702	**0.000**	0.005	0.001	0.726	**0.000**	0.625
	Mean	0.996	0.320	0.997	0.718	0.830	0.140	0.832	0.743	**0.000**	0.653
(b) Suspicious pool contamination ($\epsilon_b = 0$, $\epsilon_s = 0.1$)											
CIFAR-10	Blended	0.033	0.805	0.218	0.818	0.660	0.200	0.629	0.818	**0.000**	0.757
	Adaptive Blend	0.088	0.801	0.252	0.814	0.349	0.250	0.514	0.818	**0.001**	0.758
	SSBA	0.035	0.784	0.054	0.802	0.373	0.306	0.147	0.797	**0.006**	0.745
	BadNet	0.073	0.808	0.270	0.819	0.643	0.325	0.637	0.822	**0.007**	0.740
	WaNet	0.025	0.799	0.051	0.812	1.000	0.102	0.196	0.807	**0.004**	0.759
	TITIM-SIG	0.018	0.802	0.035	0.810	0.298	0.315	0.110	0.815	**0.003**	0.750
	Mean	0.045	0.800	0.147	0.813	0.554	0.250	0.372	0.813	**0.004**	0.752
GTSRB	Blended	**0.000**	0.881	**0.000**	0.897	**0.000**	0.159	**0.000**	0.905	0.001	0.748
	Adaptive Blend	**0.000**	0.859	**0.000**	0.892	**0.000**	0.142	0.001	0.897	**0.000**	0.697
	SSBA	**0.000**	0.866	**0.000**	0.893	0.001	0.167	**0.000**	0.892	**0.000**	0.631
	BadNet	**0.000**	0.889	0.013	0.907	**0.000**	0.286	0.510	0.904	0.004	0.701
	WaNet	0.001	0.882	0.035	0.896	**0.000**	0.034	0.170	0.895	**0.000**	0.731
	Mean	**0.000**	0.875	0.010	0.897	**0.000**	0.158	0.136	0.899	0.001	0.702
Tiny-ImageNet	Blended	0.000	0.282	0.999	0.714	0.000	0.006	0.999	0.743	**0.000**	0.476
	Adaptive Blend	0.000	0.282	0.999	0.714	0.000	0.006	0.999	0.743	**0.000**	0.496
	SSBA	0.000	0.289	0.976	0.717	0.000	0.005	0.966	0.744	**0.000**	0.547
	BadNet	0.000	0.287	0.985	0.714	0.000	0.005	0.992	0.751	**0.000**	0.530
	TITIM-SIG	0.000	0.285	0.990	0.716	0.000	0.005	0.995	0.747	**0.000**	0.520
	WaNet	0.000	0.287	0.981	0.727	**0.000**	0.005	0.918	0.743	**0.000**	0.527
	Mean	0.000	0.285	0.988	0.717	0.000	0.005	0.978	0.745	**0.000**	0.516
(c) Both pools contaminated ($\epsilon_b = 0.1$, $\epsilon_s = 0.1$)											
CIFAR-10	Blended	0.333	0.801	0.475	0.819	0.751	0.150	0.888	0.821	**0.049**	0.716
	Adaptive Blend	0.282	0.796	0.425	0.812	0.217	0.218	0.873	0.814	**0.033**	0.740
	SSBA	0.088	0.783	0.105	0.797	0.418	0.258	0.273	0.796	**0.012**	0.739
	BadNet	0.323	0.798	0.539	0.816	0.772	0.181	0.955	0.822	**0.004**	0.722
	WaNet	0.126	0.794	0.179	0.812	1.000	0.100	0.493	0.800	**0.026**	0.746
	TITIM-SIG	0.265	0.804	0.315	0.816	0.715	0.139	0.750	0.815	**0.024**	0.732
	Mean	0.236	0.796	0.340	0.813	0.645	0.174	0.722	0.811	**0.025**	0.733
GTSRB	Blended	0.468	0.878	0.604	0.898	0.198	0.149	0.939	0.903	**0.016**	0.742
	Adaptive Blend	0.312	0.868	0.416	0.896	**0.006**	0.144	0.898	0.905	0.012	0.684
	SSBA	0.376	0.876	0.602	0.896	**0.003**	0.244	0.954	0.896	0.053	0.726
	BadNet	0.344	0.881	0.607	0.897	0.609	0.205	0.939	0.902	**0.006**	0.695
	TITIM-SIG	0.355	0.882	0.486	0.900	0.081	0.092	0.846	0.909	**0.047**	0.735
	WaNet	0.040	0.891	0.067	0.902	0.129	0.037	0.148	0.900	**0.003**	0.772
	Mean	0.316	0.879	0.464	0.898	0.171	0.145	0.787	0.903	**0.023**	0.726
Tiny-ImageNet	Blended	1.000	0.307	1.000	0.704	0.984	0.360	1.000	0.738	**0.000**	0.512
	Adaptive Blend	1.000	0.307	1.000	0.704	0.984	0.360	1.000	0.738	**0.000**	0.493
	SSBA	0.998	0.284	0.999	0.722	0.999	0.393	0.999	0.746	**0.000**	0.557
	BadNet	0.995	0.297	0.995	0.716	0.976	0.011	0.994	0.739	**0.000**	0.537
	TITIM-SIG	0.995	0.249	0.982	0.730	–	–	0.994	0.741	**0.000**	0.556
	WaNet	0.991	0.286	0.997	0.723	1.000	0.005	0.995	0.744	**0.002**	0.462
	Mean	0.997	0.288	0.996	0.717	0.989	0.226	0.997	0.741	**0.000**	0.520

Table 3: Cross-tuning evaluation of PL (**true-candidate**) under perfect partition ($\epsilon_b = \epsilon_s = 0$) with Train_prop=5%. Columns denote the evaluation, or base attack A; rows denote the fine-tuning attack B used to construct the suspicious pool. All results are reported under the known-target setting, where the suspicious pool target class matches the true backdoor target class. Each triggered sample is paired with the known true target label.

		Evaluation Attack (A)									
		Blended		Adaptive Blend		SSBA		Visible		WaNet	
Data	Fine-tune (B)	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR
CIFAR-10	Blended	0.784	0.004	0.786	0.004	0.769	0.010	0.780	0.184	0.804	0.010
	Adaptive Blend	0.784	0.004	0.786	0.004	0.769	0.010	0.780	0.184	0.804	0.010
	SSBA	0.793	0.004	0.808	0.016	0.774	0.013	0.786	0.067	0.810	0.012
	Visible	0.784	0.004	0.795	0.008	0.765	0.008	0.774	0.021	0.800	0.006
	WaNet	0.793	0.003	0.800	0.006	0.780	0.011	0.789	0.073	0.801	0.007
GTSRB	Blended	0.884	0.000	0.880	0.000	0.882	0.000	0.882	0.000	0.878	0.000
	Adaptive Blend	0.884	0.000	0.880	0.000	0.882	0.000	0.882	0.000	0.878	0.000
	SSBA	0.900	0.000	0.899	0.000	0.887	0.000	0.898	0.000	0.901	0.000
	Visible	0.887	0.000	0.894	0.000	0.881	0.000	0.896	0.000	0.888	0.000
	WaNet	0.896	0.000	0.906	0.000	0.901	0.000	0.906	0.000	0.908	0.000
Tiny-ImageNet	Blended	0.697	0.000	0.697	0.000	0.697	0.000	0.692	0.000	0.636	0.001
	Adaptive Blend	0.697	0.000	0.743	0.000	0.697	0.000	0.692	0.000	0.734	0.000
	SSBA	0.688	0.000	0.637	0.000	0.649	0.000	0.682	0.000	0.707	0.004
	Visible	0.633	0.001	0.627	0.008	0.589	0.009	0.701	0.000	0.670	0.000
	WaNet	0.602	0.005	0.677	0.000	0.620	0.000	0.623	0.000	0.729	0.000

Table 4: Cross-tuning evaluation of PL (**all-candidate**) under perfect partition ($\epsilon_b = \epsilon_s = 0$) with Train_prop=5%. Columns denote evaluation (base) attack A; rows denote fine-tuning attack B. This table evaluates the unknown-target setting, where the defender does not know the true backdoor target class. For each triggered sample with ground-truth clean label y_{clean}, PL pairs the sample with all candidate target labels $c \neq y_{\text{clean}}$ and applies gradient ascent over all such pairs. Each cell reports C-ACC and ASR. Extended results on CelebA and ImageNet-12 are also in Appendix G.

		Evaluation Attack (A)											
		Blended		Adaptive Blend		SSBA		Visible		WaNet		TITIM-SIG	
Data	Fine-tune (B)	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR
CIFAR-10	Blended	0.797	0.033	0.797	0.026	0.783	0.049	0.797	0.580	0.816	0.040	0.795	0.111
	Adaptive Blend	0.797	0.033	0.797	0.026	0.783	0.049	0.797	0.580	0.816	0.040	0.795	0.111
	SSBA	0.799	0.016	0.813	0.039	0.783	0.029	0.802	0.678	0.815	0.032	0.805	0.121
	Visible	0.787	0.033	0.809	0.026	0.787	0.046	0.804	0.092	0.812	0.022	0.795	0.087
	WaNet	0.802	0.026	0.818	0.011	0.789	0.042	0.802	0.167	0.805	0.012	0.799	0.101
	TITIM-SIG	0.796	0.035	0.805	0.037	0.786	0.051	0.803	0.548	0.815	0.029	0.793	0.044
GTSRB	Blended	0.880	0.000	0.881	0.000	0.880	0.000	0.888	0.000	0.880	0.000	0.898	0.000
	Adaptive Blend	0.880	0.000	0.881	0.000	0.880	0.000	0.888	0.000	0.880	0.000	0.898	0.000
	SSBA	0.897	0.060	0.894	0.002	0.872	0.001	0.894	0.000	0.899	0.000	0.901	0.000
	Visible	0.899	0.001	0.900	0.000	0.885	0.000	0.897	0.000	0.898	0.000	0.904	0.000
	WaNet	0.897	0.000	0.902	0.001	0.890	0.001	0.897	0.022	0.905	0.000	0.910	0.000
	TITIM-SIG	0.889	0.001	0.884	0.000	0.876	0.000	0.874	0.000	0.886	0.000	0.868	0.000
Tiny-ImageNet	Blended	0.717	0.000	0.647	0.000	0.706	0.000	0.695	0.294	0.695	0.001	0.699	0.596
	Adaptive Blend	0.717	0.000	0.695	0.003	0.706	0.000	0.695	0.294	0.734	0.157	0.732	0.736
	SSBA	0.687	0.002	0.672	0.000	0.686	0.000	0.685	0.000	0.633	0.001	0.660	0.228
	Visible	0.609	0.118	0.651	0.003	0.633	0.001	0.698	0.000	0.650	0.068	0.635	0.726
	WaNet	0.634	0.619	0.711	0.004	0.632	0.984	0.650	0.971	0.735	0.000	0.676	0.586
	TITIM-SIG	0.629	0.034	0.661	0.003	0.694	0.047	0.656	0.083	0.684	0.561	0.729	0.000

trade-off emerges. As α becomes larger, ASR is suppressed at the cost of clean accuracy. Across both settings, $\alpha = 0.2$ achieves the best safety-utility balance, which is why we use it as a default value.

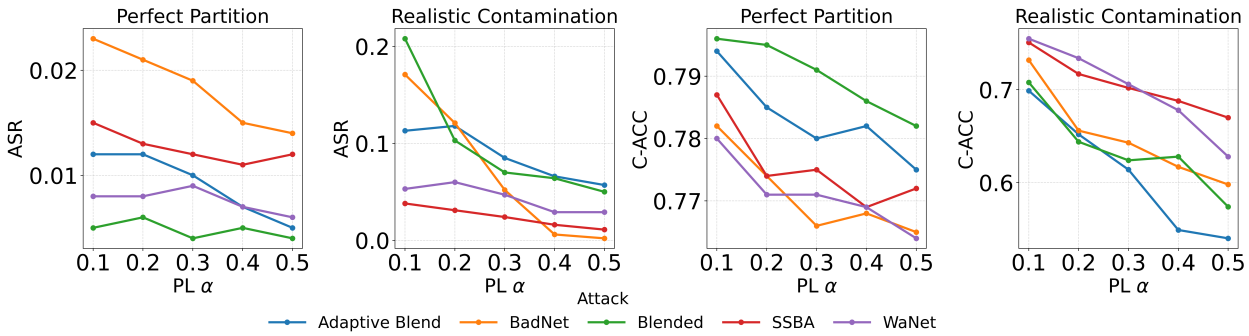

Figure 2: Sensitivity to α on CIFAR-10 under perfect partition and realistic contamination across five attacks. Increasing α consistently reduces ASR at a modest cost to C-ACC across five attacks.

Stability under varying tuning-data size. We also vary the fine-tuning fraction over $\{2\%, 5\%, 10\%\}$ under both perfect and contaminated partitions. Figure 3 shows that across all budgets, PL consistently achieves the lowest ASR while preserving clean accuracy close to the strongest clean-only baselines.

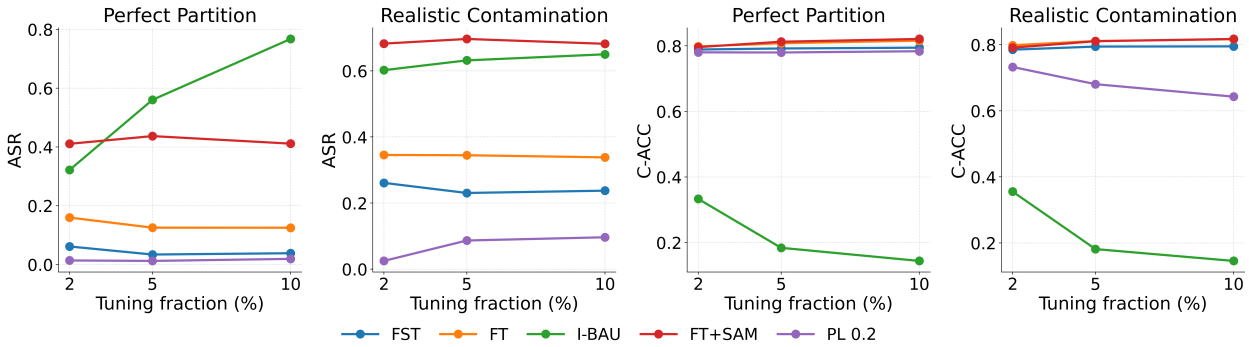

Figure 3: Sensitivity of PL and baselines to fine-tuning fraction on CIFAR-10 under perfect partition and realistic contamination. PL consistently achieves the lowest ASR while maintaining C-ACC close to the best clean-only baselines, showing its data efficiency across budgets.

These ablations confirmeither synthesized from public trigger templates or flagged that PL's performance is stable across a range of hyper parameter and data-availability settings, with $\alpha = 0.2$ providing a robust default choice. Additional results on higher contamination rates (5%–20%) under FN-only, FP-only, and combined settings are provided in Appendix H.1.

5 Conclusion

This paper proposes Partition-Losses Fine-Tuning (PL), a simple, lightweight post-training defense for mitigating poisoning-based backdoor attacks. PL relies on a small benign pool and additionally constructs a candidate suspicious pool either synthesized from public trigger templates or flagged by off-the-shelf detectors. This let PL exploit suspicious samples that prior clean-only defenses leave unused. Across multiple datasets and attacks, PL matches or outperforms state-of-the-art defenses while using substantially less clean data, and remains effective under pool contamination, trigger mismatch, and unknown target classes. PL is simple, model-agnostic, and data-efficient, making it well-suited for real-world deployment. In the broader context, PL provides a step toward building more trustworthy and resilient machine learning systems, with potential benefits for safety-critical applications such as healthcare, autonomous driving, and cybersecurity.

References

Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 101–105. IEEE, 2019. 17

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 *Adv. Neural Inform. Process. Syst.*, volume 33, pp. 1877–1901, 2020. 1

Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016. 1

Nicholas Carlini and Andreas Terzis. Poisoning and backdooring contrastive learning. In *Int. Conf. Learn. Represent.*, 2022. 3

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. In *IEEE Symp. Security Privacy*, pp. 407–425, 2024. 1

Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Ann. Comput. Security Appl. Conf.*, pp. 554–569, 2021. 3

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. 3, 6, 17

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. 6

Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. In *Adv. Neural Inform. Process. Syst.*, volume 35, pp. 5009–5023, 2022. 3

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North Am. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, pp. 4171–4186, 2019. 1

Bao Gia Doan, Ehsan Abbasnejad, and Damith C Ranasinghe. Februus: Input purification defense against trojan attacks on deep neural network systems. In *Ann. Comput. Security Appl. Conf.*, pp. 897–912, 2020. 1, 3

Yinpeng Dong, Xiao Yang, Zhijie Deng, Tianyu Pang, Zihao Xiao, Hang Su, and Jun Zhu. Black-box detection of backdoor attacks with limited information and data. In *Int. Conf. Comput. Vis.*, pp. 16482–16491, 2021. 7

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2021. 1

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 Trans. Pattern Anal. Mach. Intell.*, 45(2):1563–1580, 2022. 1, 3

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019. 2, 3, 6, 17

Wenbo Guo, Lun Wang, Yan Xu, Xinyu Xing, Min Du, and Dawn Song. Towards inspecting and eliminating trojan backdoors in deep neural networks. In *IEEE Int. Conf. Data Mining*, pp. 162–171, 2020. 2

Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. Spectre: defending against backdoor attacks using robust statistics. In *Int. Conf. Mach. Learn.*, volume 139, pp. 4129–4139, 18–24 Jul 2021. 7

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2016. 1, 6

Xianwen He, Xinglin Li, Yao Li, and Minhao Cheng. Defense against syntactic textual backdoor attacks with token substitution. *IEEE Trans. Inf. Forensics Security*, 2025. 2, 3

Mengxuan Hu, Zihan Guan, Junfeng Guo, Zhongliang Zhou, Jielu Zhang, and Sheng Li. Bbcal: Black-box backdoor detection under the causality lens. *Trans. Mach. Learn Res.*, 2024. 2

Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor defense via decoupling the training process. In *Int. Conf. Learn. Represent.*, 2022. 1, 3, 6, 17

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learn. Represent.*, 2015. 4

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 6

Yao Li, Sarah C. Van Alsten, Dong Neuck Lee, Taebin Kim, Benjamin C. Calhoun, Charles M. Perou, Sara E. Wobker, J. S. Marron, Katherine A. Hoadley, and Melissa A. Troester. Visual intratumor heterogeneity and breast tumor progression. *Cancers*, 16(13), 2024. ISSN 2072-6694. 1

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In *Adv. Neural Inform. Process. Syst.*, 2021a. 1, 3

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *Int. Conf. Learn. Represent.*, 2021b. 2

Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Int. Conf. Comput. Vis.*, pp. 16463–16472, October 2021c. 2, 3, 6, 17

Chenhao Lin, Chenyang Zhao, Shiwei Wang, Longtian Wang, Chao Shen, and Zhengyu Zhao. Revisiting {Training-Inference} trigger intensity in backdoor attacks. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 6359–6378, 2025. 6, 17

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Int. Symp. Res. Attacks Intrusions Defenses*, pp. 273–294, 2018. 2, 3, 6, 17

Li Liu, Gang Feng, Denis Beautemps, and Xiao-Ping Zhang. Re-synchronization using the hand preceding model for multi-modal fusion in automatic continuous cued speech recognition. *IEEE Trans. Multimedia*, 23:292–305, 2020. 1

Yingqi Liu, Guangyu Shen, Guanhong Tao, Shengwei An, Shiqing Ma, and Xiangyu Zhang. Piccolo: Exposing complex backdoors in nlp transformer models. In *IEEE Symp. Security Privacy*, pp. 2025–2042, 2022. 3

Yuntao Liu, Yang Xie, and Ankur Srivastava. Neural trojans. In *IEEE Int. Conf. Comput. Design*, pp. 45–48, 2017. 3

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Int. Conf. Comput. Vis.*, pp. 10012–10022, 2021. 6

Rui Min, Zeyu Qin, Li Shen, and Minhao Cheng. Towards stable backdoor purification through feature shift tuning. In *Adv. Neural Inform. Process. Syst.*, volume 36, pp. 75286–75306, 2023. 2, 3, 6, 17, 18

Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *Int. Conf. Learn. Represent.*, 2021. 2, 6, 17

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Adv. Neural Inform. Process. Syst.*, 2021. 7

Xudong Pan, Mi Zhang, Beina Sheng, Jiaming Zhu, and Min Yang. Hidden trigger backdoor attack on nlp models via linguistic style manipulation. In *USENIX Security Symp.*, pp. 3611–3628, 2022. 3

Xiangyu Qi, Tinghao Xie, Yiming Li, Saeed Mahloujifar, and Prateek Mittal. Revisiting the assumption of latent separability for backdoor defenses. In *Int. Conf. Learn. Represent.*, 2023. 6, 17

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Int. Conf. Mach. Learn.*, pp. 8748–8763, 2021. 1

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 *IEEE Symp. Security Privacy*, pp. 1354–1371, 2022. 1

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Netw.*, 32:323–332, 2012. 6

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Adv. Neural Inform. Process. Syst.*, volume 31, 2018. 1, 3

Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019. 3

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE Symp. Security Privacy*, pp. 707–723, 2019. 1, 3

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. 3

Zhenting Wang, Kai Mei, Juan Zhai, and Shiqing Ma. UNICORN: A unified backdoor trigger inversion framework. In *Int. Conf. Learn. Represent.*, 2023. 3

Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoorbench: A comprehensive benchmark of backdoor learning. In *Adv. Neural Inform. Process. Syst.*, volume 35, pp. 10546–10559, 2022. 3, 6, 17

Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *Adv. Neural Inform. Process. Syst.*, volume 34, pp. 16913–16925, 2021. 2, 3

Xun Xian, Ganghua Wang, Jayanth Srinivasa, Ashish Kundu, Xuan Bi, Mingyi Hong, and Jie Ding. A unified detection framework for inference-stage backdoor defenses. In *Adv. Neural Inform. Process. Syst.*, volume 36, pp. 7867–7894, 2023. 2

Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, and Kui Ren. Towards reliable and efficient backdoor trigger inversion via decoupling benign features. In *Int. Conf. Learn. Represent.*, 2024. 3

Yi Zeng, Si Chen, Won Park, Zhuoqing Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *Int. Conf. Learn. Represent.*, 2022. 3, 6, 18

Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Data-free backdoor removal based on channel lipschitzness. In *Eur. Conf. Comput. Vis.*, pp. 175–191, 2022. 3

Mingli Zhu, Shaokui Wei, Li Shen, Yanbo Fan, and Baoyuan Wu. Enhancing fine-tuning based backdoor defense with sharpness-aware minimization. In *Int. Conf. Comput. Vis.*, pp. 4466–4477, 2023. 3, 6, 18

A Proof of Theorem 1

Proof. The expected gradient of $\mathcal{L}_{\mathrm{PL}}$ is

$$\nabla_\theta \mathcal{L}_{\mathrm{PL}} = \underbrace{\mathbb{E}_{(X,Y)\sim\mathcal{D}_b^{\epsilon_b}}\big[\nabla_\theta \operatorname{CE}(f_\theta(X), Y)\big]}_{=:\nabla\mathcal{L}_b} - \alpha \underbrace{\mathbb{E}_{(\tilde{Z},T)\sim\mathcal{D}_s^{\epsilon_s}}\big[\nabla_\theta \operatorname{CE}(f_\theta(\tilde{Z}), t)\big]}_{=:\nabla\mathcal{L}_m}. \tag{2}$$

Note that

$$\begin{aligned}\nabla\mathcal{L}_m &= (1-\epsilon_s)\mathbb{E}_{\mathcal{P}_{\mathrm{trigger}}}\big[\nabla \operatorname{CE}(f_\theta(\tilde{Z}), t)\big] + \epsilon_s \mathbb{E}_{\mathcal{P}_{\mathrm{clean}}}\big[\nabla \operatorname{CE}(f_\theta(X), t)\big] \\ &= (1-\epsilon_s) g_{\mathrm{trigger}} + \epsilon_s g_{\mathrm{clean}}.\end{aligned} \tag{3}$$

For a small step size $\eta > 0$, the parameter update is $\theta^+ = \theta - \eta\nabla_\theta L(\theta)$. The change in surrogate ASR is $\Delta\widetilde{\mathrm{ASR}} = \widetilde{\mathrm{ASR}}(f_{\theta^+}) - \widetilde{\mathrm{ASR}}(f_\theta)$. By a first-order Taylor expansion of $\widetilde{\mathrm{ASR}}$ around θ,

$$\Delta\widetilde{\mathrm{ASR}} \approx \big\langle \nabla_\theta \widetilde{\mathrm{ASR}}(f_\theta),\, \theta^+ - \theta \big\rangle.$$

Substituting the update rule gives

$$\Delta\widetilde{\mathrm{ASR}} \approx -\eta \big\langle \nabla_\theta \widetilde{\mathrm{ASR}}(f_\theta),\, \nabla_\theta L(\theta) \big\rangle. \tag{4}$$

Since $\widetilde{\mathrm{ASR}}(f_\theta) = -\mathbb{E}_{\mathcal{P}_{\mathrm{trigger}}}\big[\operatorname{CE}(f_\theta(\tilde{Z}), t)\big]$, we have $\nabla_\theta \widetilde{\mathrm{ASR}}(f_\theta) = -g_{\mathrm{trig}}$. Equation 4 simplifies to

$$\Delta\widetilde{\mathrm{ASR}} \approx \eta \big\langle g_{\mathrm{trig}},\, \nabla_\theta L(\theta) \big\rangle.$$

Substituting Equation 2 and Equation 3 gives

$$\Delta\widetilde{\mathrm{ASR}} \approx \eta\Big(\langle g_{\mathrm{trig}}, \nabla\mathcal{L}_b\rangle - \alpha(1-\epsilon_s)\|g_{\mathrm{trig}}\|^2 - \alpha\,\epsilon_s \langle g_{\mathrm{trig}}, g_{\mathrm{clean}}\rangle \Big).$$

By comparison, clean-only fine-tuning corresponds to $\alpha = 0$, in which case

$$\Delta\widetilde{\mathrm{ASR}}_{\mathrm{clean}} \approx \eta \langle g_{\mathrm{trig}}, \nabla\mathcal{L}_b\rangle.$$

Therefore, the difference between PL and clean-only fine-tuning is

$$\Delta\widetilde{\mathrm{ASR}} - \Delta\widetilde{\mathrm{ASR}}_{\mathrm{clean}} \approx -\eta\alpha\Big((1-\epsilon_s)\|g_{\mathrm{trig}}\|^2 + \epsilon_s \langle g_{\mathrm{trig}}, g_{\mathrm{clean}}\rangle \Big). \tag{5}$$

Under the comparable-norms assumption $\frac{1}{\rho}\|g_{\mathrm{clean}}\| \le \|g_{\mathrm{trig}}\| \le \rho\|g_{\mathrm{clean}}\|$ and by the Cauchy-Schwarz inequality,

$$\begin{aligned}\langle g_{\mathrm{trig}}, g_{\mathrm{clean}}\rangle &\ge -\|g_{\mathrm{trig}}\|\,\|g_{\mathrm{clean}}\| \\ &\ge -\rho\|g_{\mathrm{clean}}\|^2.\end{aligned}$$

Also, the lower side of the assumption gives

$$\|g_{\mathrm{trig}}\|^2 \ge \frac{1}{\rho^2}\|g_{\mathrm{clean}}\|^2.$$

Therefore the bracketed term in Equation 5 admits the lower bound

$$\begin{aligned}&(1-\epsilon_s)\|g_{\mathrm{trig}}\|^2 + \epsilon_s\langle g_{\mathrm{trig}}, g_{\mathrm{clean}}\rangle \\ &\quad\ge (1-\epsilon_s)\frac{1}{\rho^2}\|g_{\mathrm{clean}}\|^2 - \epsilon_s\rho\|g_{\mathrm{clean}}\|^2 \\ &\quad= \left(\frac{1-\epsilon_s}{\rho^2} - \epsilon_s\rho\right)\|g_{\mathrm{clean}}\|^2.\end{aligned}$$

This lower bound is positive whenever

$$\frac{1-\epsilon_s}{\rho^2} - \epsilon_s \rho > 0,$$

equivalently,

$$1 - \epsilon_s > \epsilon_s \rho^3.$$

Hence it suffices that

$$\epsilon_s < \frac{1}{1+\rho^3}.$$

This shows that the PL update always reduces the surrogate ASR *more* than clean-only fine-tuning whenever $\epsilon_s < \frac{1}{1+\rho^3}$, for any $\alpha > 0$.

Finally, to obtain an *absolute* decrease guarantee, note from

$$\Delta\widetilde{\mathrm{ASR}} \approx \eta\Big(\langle g_{\mathrm{trig}}, \nabla\mathcal{L}_b\rangle - \alpha(1-\epsilon_s)\|g_{\mathrm{trig}}\|^2 - \alpha\epsilon_s\langle g_{\mathrm{trig}}, g_{\mathrm{clean}}\rangle\Big)$$

that a sufficient condition for $\Delta\widetilde{\mathrm{ASR}} < 0$ is

$$\alpha \;>\; \frac{\max\{0,\, \langle g_{\mathrm{trig}}, \nabla\mathcal{L}_b\rangle\}}{(1-\epsilon_s)\|g_{\mathrm{trig}}\|^2 + \epsilon_s\langle g_{\mathrm{trig}}, g_{\mathrm{clean}}\rangle}.$$

In particular, when $\epsilon_s < \frac{1}{1+\rho^3}$, the denominator is strictly positive by the argument above, so such an α always exists. □

B Experimental Setup

Datasets and Models. Following previous works Min et al. (2023); Huang et al. (2022); Liu et al. (2018); Nguyen & Tran (2021); Wu et al. (2022), we evaluate on five widely used benchmark datasets in the backdoor learning literature: CIFAR-10, GTSRB, Tiny-ImageNet, CelebA and ImageNet-12. CIFAR-10 and GTSRB contain images of 32×32 resolution of 10 and 43 categories. CelebA (CelebFaces Attributes) consists of face images originally of 178×218 pixels, which we resize to 128×128. We use ResNet-18 to build a backdoor model for those three datasets (CIFAR-10, GTSRB, and CelebA). Tiny-ImageNet contains images of 64×64 resolution and 100 categories, and ImageNet-12 is a 12-class subset of ImageNet. Both datasets are resized to 224×224. For these two higher-resolution datasets, we used a pre-trained SwinTransformer provided by PyTorch as the backdoor.

Attack Settings. We conducted all the experiments with 4× NVIDIA RTX A6000 GPUs (48 GiB each). We implement six representative backdoor attacks from recent works: BadNet, Blended, SSBA, WaNet, Adaptive Blend, and TITIM-SIG. For BadNet Gu et al. (2019), we use a white square as the backdoor trigger and stamp the pattern at the lower right corner of the image; for Blended Chen et al. (2017), we adopt the uniform noise as the trigger and set the blend ratio as 0.1 for both the training and inference phases; for WaNet Nguyen & Tran (2021), we set the size of the backward warping field as four and the strength of the wrapping field as 0.5; SSBA Li et al. (2021c) is a dynamic invisible-trigger attack. We implement the complete SSBA encoder–decoder pipeline: we train the SSBA steganographic trigger, where a learnable encoder injects a binary fingerprint into each input image, and a decoder is jointly trained to recover this fingerprint at a poisoning rate of 0.15; Adaptive Blend Qi et al. (2023) is an enhanced variant of the classical blended attack. It consists of input-dependent partial blending and probabilistic label flipping. During poisoning, the image is divided into a grid of patches; only half of the patches are randomly selected for blending with the trigger at the poisoning rate of 0.15. The remaining patches stay unchanged, creating a spatially sparse and input-dependent backdoor pattern. TITIM-SIG Lin et al. (2025), which builds upon the SIG attack Barni et al. (2019), uses an additive horizontal sinusoidal wave as the trigger. Specifically, the trigger added to each image channel is defined as

$$p_{c,r,j} = \frac{\delta}{255}\sin\left(\frac{2\pi f j}{W}\right), \quad c = 1,\ldots,C,\; r = 1,\ldots,H,\; j = 0,\ldots,W-1,$$

Table 5: GTSRB variability statistics under the PL-2 regime with perfect partition ($\epsilon_b = \epsilon_s = 0$). Each cell reports mean±std over 5 seeds for attack success rate (ASR, lower is better) and clean accuracy (C-ACC, higher is better). This table corresponds to the GTSRB block of Table 1 in the main text, which reports means only for readability.

	FST		FT		I-BAU		FT+SAM		FST+PL		PL (Ours)	
Attack	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC
Blended	$0.000_{\pm 0.000}$	$0.863_{\pm 0.011}$	$0.018_{\pm 0.036}$	$0.886_{\pm 0.006}$	$0.005_{\pm 0.011}$	$0.184_{\pm 0.027}$	$0.120_{\pm 0.255}$	$0.906_{\pm 0.002}$	$0.000_{\pm 0.000}$	$0.804_{\pm 0.015}$	$0.000_{\pm 0.000}$	$0.843_{\pm 0.013}$
Adaptive Blend	$0.000_{\pm 0.000}$	$0.856_{\pm 0.016}$	$0.001_{\pm 0.001}$	$0.888_{\pm 0.012}$	$0.000_{\pm 0.000}$	$0.171_{\pm 0.027}$	$0.002_{\pm 0.003}$	$0.909_{\pm 0.005}$	$0.000_{\pm 0.000}$	$0.794_{\pm 0.010}$	$0.000_{\pm 0.000}$	$0.824_{\pm 0.012}$
TITIM-SIG	$0.000_{\pm 0.000}$	$0.841_{\pm 0.018}$	$0.000_{\pm 0.000}$	$0.858_{\pm 0.016}$	$0.001_{\pm 0.003}$	$0.128_{\pm 0.019}$	$0.000_{\pm 0.001}$	$0.910_{\pm 0.008}$	$0.000_{\pm 0.000}$	$0.800_{\pm 0.020}$	$0.000_{\pm 0.000}$	$0.826_{\pm 0.016}$
SSBA	$0.000_{\pm 0.001}$	$0.865_{\pm 0.004}$	$0.004_{\pm 0.003}$	$0.889_{\pm 0.004}$	$0.001_{\pm 0.001}$	$0.253_{\pm 0.057}$	$0.004_{\pm 0.005}$	$0.900_{\pm 0.006}$	$0.000_{\pm 0.000}$	$0.804_{\pm 0.011}$	$0.000_{\pm 0.000}$	$0.845_{\pm 0.007}$
Visible	$0.000_{\pm 0.000}$	$0.879_{\pm 0.003}$	$0.000_{\pm 0.000}$	$0.905_{\pm 0.004}$	$0.026_{\pm 0.046}$	$0.242_{\pm 0.021}$	$0.003_{\pm 0.005}$	$0.909_{\pm 0.004}$	$0.000_{\pm 0.000}$	$0.815_{\pm 0.014}$	$0.000_{\pm 0.000}$	$0.868_{\pm 0.015}$
WaNet	$0.000_{\pm 0.000}$	$0.891_{\pm 0.012}$	$0.004_{\pm 0.008}$	$0.907_{\pm 0.006}$	$0.000_{\pm 0.001}$	$0.069_{\pm 0.030}$	$0.027_{\pm 0.060}$	$0.906_{\pm 0.007}$	$0.000_{\pm 0.000}$	$0.823_{\pm 0.017}$	$0.000_{\pm 0.000}$	$0.857_{\pm 0.012}$
Mean	0.000	0.866	0.004	0.889	0.006	0.175	0.026	0.907	0.000	0.807	0.000	0.844

where c, r, and j denote the channel, row and horizontal pixel coordinate, respectively; W is the image width; f is the spatial frequency; and δ controls the trigger amplitude. We set the spatial frequency to $f = 6$ cycles per image width. To implement TITIM-style mixed-intensity poisoning, we set the base amplitude to $\delta = 10$ and sample the training-time trigger amplitude uniformly from $[10, 60]$ for each poisoned sample; during inference, we use the fixed amplitude $\delta = 10$.

All backdoor models are trained for 50 epochs with an initial learning rate of 0.0001. Then, we fine-tune each model for 10 epochs.

Defenses Settings. We compare four tuning-based defenses, namely FT-Init, FT+SAM, FST, and FST+PL, and one extra state-of-the-art defense strategy, I-BAU. For all defense settings, we set the batch size to 128 on CIFAR-10, GTSRB, and CelebA and set the batch size as 32 on Tiny-ImageNet and ImageNet-12 due to the memory limit. I-BAU utilizes the implicit hypergradients to account for the interdependence between inner and outer optimization problems Zeng et al. (2022). FT-init randomly re-initializes the linear head and fine-tunes the entire model Min et al. (2023). FT+SAM replace the optimizer in FT-Init with SAM Min et al. (2023); Zhu et al. (2023). FST shifts features by encouraging the discrepancy between the tuned classifier weight and the original backdoored classifier weight Min et al. (2023). FST+PL combines the feature-shift regularization in FST with our partition-loss objective. Unlike standard clean-only fine-tuning defense, which assume access to a small clean fine-tuning set, FST+PL additionally uses a defender-constructed suspicious pool. Specifically, it preserves benign utility by minimizing the classification loss on the clean fine-tuning set, while suppressing the backdoor behavior by maximizing the classification loss on the suspicious pool.

C Variability of GTSRB Results

Under perfect partition and for GTSRB, we run each combination over 5 seeds. Table 1 in the main text reports the mean ASR and C-ACC values to improve readability. Here in the appendix, Table 5 reports the corresponding variability statistics for the PL-2 regime.

D Compatibility of PL with Feature Shift Tuning under Perfect partition

To evaluate whether PL can be integrated with existing post-training defenses, we combine PL with Feature Shift Tuning (FST), denoted as FST+PL. FST+PL combines the feature-shift regularization in the FST with the partition-loss objective: it minimizes the classification loss on the clean fine-tuning set while maximizing the classification loss on the defender-constructed suspicious pool. Because FST+PL results are available only for a subset of settings, we report them separately rather than including them in the main comparison table. As shown in Table 6, FST+PL achieves comparable ASR to PL in the evaluated settings, suggesting that PL can be combined with existing post-training defenses.

E Performance under Contaminated Partition of Tuning Sets.

This Appendixprovides the complete results for the partition-contamination experiments summarized on CIFAR-10 in Section 4.2 of the main text. It has three components: (i) extended PL-2 results on CelebA and

Table 6: Compatibility evaluation of FST+PL under the PL-2 regime and perfect partition. FST+PL combines Feature Shift Tuning with the partition-loss objective. Because FST+PL results are available only for a subset of settings, we report them separately from the main comparison table. Each cell reports ASR and C-ACC.

		FST+PL		PL (Ours)	
Data	Attack	ASR	C-ACC	ASR	C-ACC
CIFAR-10	TITIM-SIG	**0.002**	0.792	0.003	0.798
GTSRB	Blended	**0.000**	0.804	**0.000**	0.843
	Adaptive Blend	**0.000**	0.794	**0.000**	0.824
	TITIM-SIG	**0.000**	0.800	**0.000**	0.826
	SSBA	**0.000**	0.804	**0.000**	0.845
	Visible	**0.000**	0.815	**0.000**	0.868
	WaNet	**0.000**	0.823	**0.000**	0.857
CelebA	Blended	0.066	0.663	**0.060**	0.682
	Adaptive Blend	0.044	0.705	**0.035**	0.669
	TITIM-SIG	**0.030**	0.730	0.035	0.731
	SSBA	**0.032**	0.766	0.034	0.762
	Visible	**0.001**	0.588	0.007	0.615
	WaNet	**0.011**	0.712	0.025	0.702
ImageNet-12	Blended	**0.000**	0.860	**0.000**	0.860
	Adaptive Blend	**0.000**	0.857	**0.000**	0.872
	TITIM-SIG	**0.000**	0.875	**0.000**	0.885
	SSBA	**0.000**	0.868	**0.000**	0.885
	Visible	**0.000**	0.888	**0.000**	0.892
	WaNet	**0.000**	0.863	**0.000**	0.857
Mean over available settings		**0.010**	0.790	0.010	0.804

ImageNet- (Appendix E.1); (ii) combined-defense (FST+PL) comparisons under both-pool contamination (Appendix E.2); and (iii) full PL-5 regime results across all five datasets (Appendix E.3).

E.1 Extended PL-2 Results on CelebA and ImageNet-12

Table 7 reports PL-2 results on CelebA and ImageNet-12 across three contamination cases, complementing the three-dataset results in Table 2 of the main text. The qualitative pattern is preserved: PL achieves the lowest ASR in every dataset $\times$ contamination combination. On those two more complex Tiny-ImageNet and ImageNet-12, clean-only baselines fail catastrophically under benign-pool contamination (ASR > 0.98 across most attacks). In contrast, PL trades a modest amount of clean accuracy for a large reduction in ASR.

E.2 Combined-Defense Results: FST+PL under Both-Pool Contamination

Table 8 reports the FST+PL combined defense results under both-pool contamination ($\epsilon_b = \epsilon_s = 0.1$) under the PL-2 regime, which were deferred from Table 2c of the main text. FST+PL combines PL's suspicious-pool regularization with FST's feature-shift tuning. On both CIFAR-10 and GTSRB, FST+PL achieves ASR comparable to PL alone. This confirms that PL can be integrated with existing post-training defenses without loss of effectiveness.

E.3 PL-5 Regime

Table 9 reports the same three contamination cases under the PL-5 regime (PL uses 5% of the training data as the tuning set; baselines uses 10%). The qualitative pattern from PL-2 is preserved; PL achieves the lowest grand-mean ASR across all the three contamination cases (0.063, 0.002 0.034 for benign-, suspicious-,

and both-pool contamination respectively). In contrast, clean-only baselines either leave the backdoor active or substantially sacrifice the clean accuracy.

THe largeer tuning budge under PL-5 modestly improves clean accuracy across all method but does not change the relative ordering. PL remains the monly method achieving both low ASR and competitive clean accuracy across all contamination cases.

F Performance under cross-tune setting with Imperfect Partitions.

In this section, we further evaluate PL under the cross-tuning setting with imperfectly constructed clean and suspicious pools. Here, cross-tuning refers to the case where the attack used to construct the suspicious pool $\mathcal{D}_s$ may differ from the true attack used to poison the victim model. Importantly, this section focuses on the known-target setting: the target class used when constructing the $\mathcal{D}_s$ matches the true target class of the original backdoor attack. The unknown-target setting is evaluate separately in Section 4.3.

In practice, an imperfect partition between clean and suspicious pools may occur when clean and suspicious pools are constructed using automated monitoring or anomaly detection tools. To assess robustness under such imperfect partitioning, we evaluate PL with $\epsilon_b = 0.1, \epsilon_s = 0$ (clean-pool contamination, Table 10), $\epsilon_b = 0, \epsilon_s = 0.1$ (suspicious-pool contamination, Table 11), and $\epsilon_b = \epsilon_s = 0.1$ (both pools contaminated, Table 12).

We emphasize that these imperfect-partition experiments serve as sensitivity analyses. In the cross-tuning scenarios discussed in the main text, suspicious samples are generated by defenders and are thus quality-controlled. The imperfect cases demonstrate that while ASR suppression may weaken under moderate contamination, PL does not exhibit systematic collapse and maintains competitive clean accuracy across datasets.

G Cross-Tuning Results under Unknown Target

This section provides additional cross-tuning results under the unknown-target setting, complementing Table 4 in Section 4.3. We report (i) the per-seed standard deviations for GTSRB (5 seeds), which were suppressed in the main table for visual consistency (Table 13); and (ii) extended evaluation on CelebA and ImageNet-12 (Table 14). The experimental setup is identical across all tables: perfect partition ($\epsilon_b = \epsilon_s = 0$) with Train_prop=5%. Columns denote evaluation attack A and rows denote fine-tuning attack B. For each triggered sample with clean label y_{clean}, PL pairs the sample with all candidate target labels $c \neq y_{\text{clean}}$ and applies gradient ascent over all such pairs. Overall, PL transfers most effectively when the suspicious-poll and evaluation attacks share visual or structural similarity, consistent with the general pattern in Section 4.3.

H Additional ablation studies of PL

H.1 Sensitivity Analysis on Partition Contamination

We evaluate both the proposed Partition-Loss fine-tuning (PL) and the baseline fine-tuning method under controlled benign contamination (False Negative, FN only), suspicious contamination (False Positive, FP only), and combined (FN+FP) contamination at the rate of 5%, 10%, 15%, 20%. These results (Figure 4, Figure 5, Figure 6, Figure 7, Figure 8, Figure 9) complement the main text and demonstrate the expected performance degradation as contamination increases, while also showing that PL remains competitive under all tested regimes.

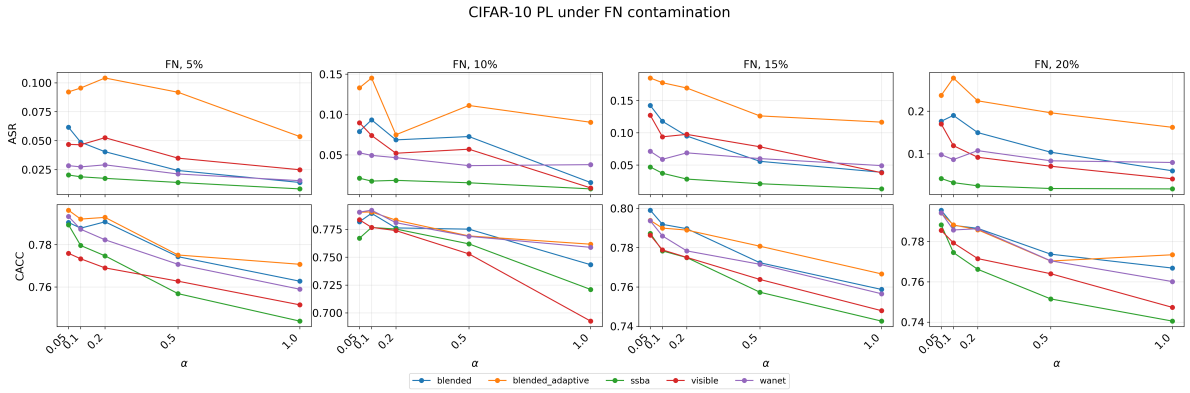

Figure 4: **PL under benign contamination (FN only).**

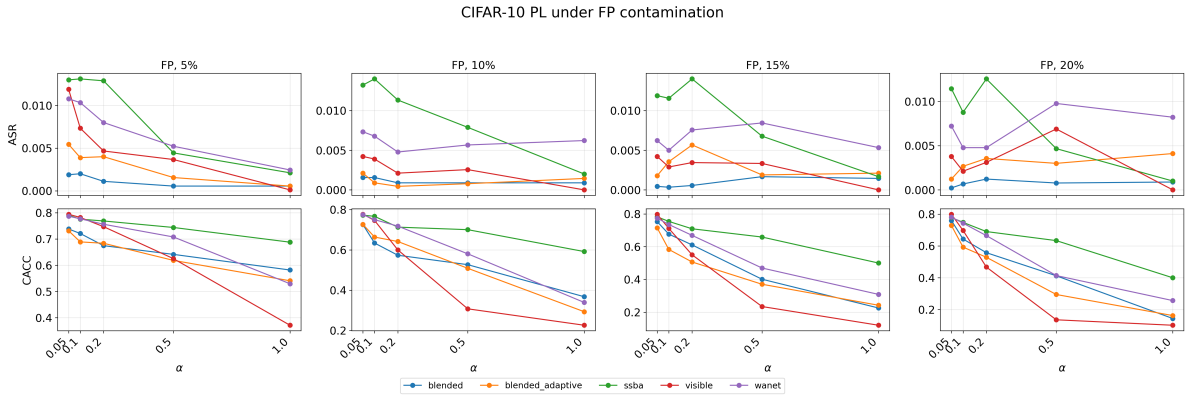

Figure 5: **PL under suspicious contamination (FP only).**

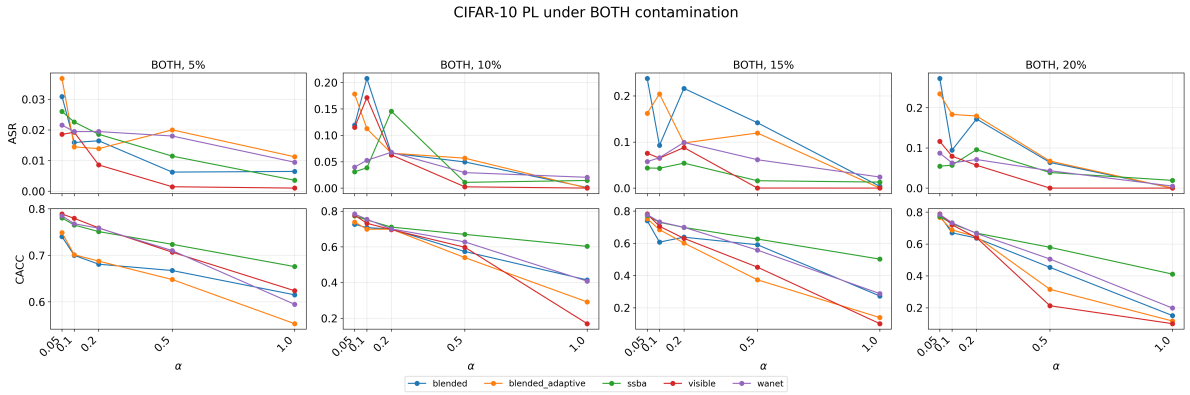

Figure 6: **PL under combined contamination (FN+FP).**

Table 7: Performance under benign-pool contamination (a) and suspicious-pool contamination (b) on the PL-2 regime. Bold values indicate the lowest ASR per row. PL consistently achieves the best safety-utility trade-off across both settings.

Data	Attack	FST		FT		I-BAU		FT+SAM		PL (Ours)	
		ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC
(a) Benign pool contamination $\epsilon_b = 0.1,\ \epsilon_s = 0$											
CIFAR-10	Blended	0.225	0.787	0.336	0.809	0.344	0.264	0.885	0.818	**0.061**	0.789
	Adaptive Blend	0.245	0.797	0.339	0.808	0.562	0.159	0.879	0.811	**0.091**	0.787
	SSBA	0.075	0.769	0.063	0.801	0.694	0.160	0.221	0.794	**0.018**	0.773
	BadNet	0.205	0.793	0.349	0.815	0.808	0.232	0.957	0.823	**0.057**	0.773
	WaNet	0.092	0.798	0.128	0.812	0.999	0.102	0.507	0.802	**0.033**	0.780
	TITIM-SIG	0.087	0.799	0.110	0.807	0.466	0.273	0.192	0.814	**0.018**	0.779
	Mean	0.155	0.791	0.221	0.809	0.646	0.198	0.607	0.810	**0.046**	0.780
GTSRB	Blended	0.363	0.875	0.481	0.890	**0.000**	0.127	0.939	0.907	0.324	0.839
	Adaptive Blend	0.382	0.853	0.505	0.878	**0.000**	0.157	0.927	0.900	0.370	0.822
	SSBA	0.370	0.882	0.536	0.900	**0.013**	0.189	0.930	0.900	0.298	0.860
	BadNet	0.318	0.880	0.524	0.900	**0.091**	0.114	0.936	0.905	0.299	0.868
	WaNet	0.107	0.875	0.181	0.897	0.998	0.006	0.625	0.899	**0.058**	0.844
	Mean	0.308	0.873	0.445	0.893	**0.220**	0.119	0.871	0.902	0.270	0.847
Tiny-ImageNet	Blended	1.000	0.320	1.000	0.722	1.000	0.014	1.000	0.744	**0.000**	0.677
	Adaptive Blend	1.000	0.320	1.000	0.722	1.000	0.014	1.000	0.744	**0.000**	0.665
	SSBA	0.994	0.307	0.998	0.718	0.996	0.428	0.997	0.749	**0.000**	0.654
	BadNet	0.993	0.326	0.995	0.724	0.990	0.013	0.995	0.751	**0.000**	0.651
	TITIM-SIG	0.997	0.315	0.998	0.719	0.993	0.365	0.998	0.746	**0.000**	0.645
	WaNet	0.991	0.333	0.992	0.702	**0.000**	0.005	0.001	0.726	**0.000**	0.625
	Mean	0.996	0.320	0.997	0.718	0.830	0.140	0.832	0.743	**0.000**	0.653
CelebA	Blended	0.550	0.765	0.610	0.772	0.762	0.496	0.922	0.792	**0.190**	0.656
	Adaptive Blend	0.520	0.759	0.588	0.770	0.711	0.512	0.908	0.790	**0.175**	0.638
	SSBA	0.380	0.785	0.445	0.788	0.593	0.588	0.800	0.791	**0.155**	0.702
	BadNet	0.495	0.775	0.568	0.780	0.700	0.520	0.890	0.789	**0.168**	0.672
	TITIM-SIG	0.315	0.788	0.389	0.789	0.520	0.602	0.658	0.793	**0.120**	0.718
	WaNet	0.470	0.782	0.540	0.785	0.880	0.380	0.850	0.790	**0.210**	0.680
	Mean	0.455	0.776	0.523	0.781	0.694	0.516	0.838	0.791	**0.170**	0.678
ImageNet-12	Blended	0.985	0.892	0.992	0.895	0.980	0.888	0.997	0.900	**0.000**	0.815
	Adaptive Blend	0.970	0.885	0.985	0.888	0.972	0.882	0.990	0.895	**0.000**	0.822
	SSBA	0.965	0.890	0.978	0.892	0.955	0.885	0.985	0.898	**0.000**	0.840
	BadNet	0.975	0.888	0.988	0.890	0.968	0.880	0.995	0.897	**0.000**	0.830
	TITIM-SIG	0.950	0.895	0.970	0.897	0.940	0.890	0.980	0.902	**0.000**	0.845
	WaNet	0.950	0.893	0.965	0.895	**0.000**	0.005	0.005	0.900	**0.000**	0.810
	Mean	0.966	0.891	0.980	0.893	0.803	0.738	0.825	0.899	**0.000**	0.827
Grand Mean (a)		0.577	0.730	0.635	0.819	0.631	0.343	0.796	0.829	**0.098**	0.756
(b) Suspicious pool contamination $\epsilon_b = 0,\ \epsilon_s = 0.1$											
CIFAR-10	Blended	0.033	0.805	0.218	0.818	0.660	0.200	0.629	0.818	**0.000**	0.757
	Adaptive Blend	0.088	0.801	0.252	0.814	0.349	0.250	0.514	0.818	**0.001**	0.758
	SSBA	0.035	0.784	0.054	0.802	0.373	0.306	0.147	0.797	**0.006**	0.745
	BadNet	0.073	0.808	0.270	0.819	0.643	0.325	0.637	0.822	**0.007**	0.740
	WaNet	0.025	0.799	0.051	0.812	1.000	0.102	0.196	0.807	**0.004**	0.759
	TITIM-SIG	0.018	0.802	0.035	0.810	0.298	0.315	0.110	0.815	**0.003**	0.750
	Mean	0.045	0.800	0.147	0.813	0.554	0.250	0.372	0.813	**0.004**	0.752
GTSRB	Blended	**0.000**	0.881	**0.000**	0.897	**0.000**	0.159	**0.000**	0.905	0.001	0.748
	Adaptive Blend	**0.000**	0.859	**0.000**	0.892	**0.000**	0.142	0.001	0.897	**0.000**	0.697
	SSBA	**0.000**	0.866	**0.000**	0.893	0.001	0.167	**0.000**	0.892	**0.000**	0.631
	BadNet	**0.000**	0.889	0.013	0.907	**0.000**	0.286	0.510	0.904	0.004	0.701
	WaNet	0.001	0.882	0.035	0.896	**0.000**	0.034	0.170	0.895	**0.000**	0.731
	Mean	**0.000**	0.875	0.010	0.897	**0.000**	0.158	0.136	0.899	0.001	0.702
Tiny-ImageNet	Blended	0.000	0.282	0.999	0.714	0.000	0.006	0.999	0.743	**0.000**	0.476
	Adaptive Blend	0.000	0.282	0.999	0.714	0.000	0.006	0.999	0.743	**0.000**	0.496
	SSBA	0.000	0.289	0.976	0.717	0.000	0.005	0.966	0.744	**0.000**	0.547
	BadNet	0.000	0.287	0.985	0.714	0.000	0.005	0.992	0.751	**0.000**	0.530
	TITIM-SIG	0.000	0.285	0.990	0.716	0.000	0.005	0.995	0.747	**0.000**	0.520
	WaNet	0.000	0.287	0.981	0.727	**0.000**	0.005	0.918	0.743	**0.000**	0.527
	Mean	0.000	0.285	0.988	0.717	0.000	0.005	0.978	0.745	**0.000**	0.516
CelebA	Blended	0.180	0.790	0.320	0.795	0.560	0.580	0.750	0.793	**0.080**	0.680
	Adaptive Blend	0.160	0.788	0.290	0.792	0.500	0.590	0.720	0.790	**0.070**	0.670
	SSBA	0.120	0.792	0.250	0.794	0.450	0.605	0.680	0.792	**0.055**	0.720
	BadNet	0.150	0.786	0.280	0.790	0.520	0.595	0.700	0.791	**0.065**	0.690
	TITIM-SIG	0.100	0.795	0.210	0.796	0.380	0.615	0.620	0.794	**0.045**	0.730
	WaNet	0.200	0.790	0.350	0.792	0.790	0.420	0.780	0.793	**0.095**	0.690
	Mean	0.152	0.790	0.283	0.793	0.533	0.568	0.708	0.792	**0.068**	0.697
ImageNet-12	Blended	0.010	0.895	0.995	0.898	0.970	0.890	0.998	0.902	**0.000**	0.845
	Adaptive Blend	0.005	0.890	0.985	0.892	0.960	0.885	0.990	0.897	**0.000**	0.850
	SSBA	0.008	0.892	0.980	0.895	0.950	0.888	0.985	0.900	**0.000**	0.855
	BadNet	0.006	0.888	0.990	0.890	0.965	0.882	0.995	0.895	**0.000**	0.840
	TITIM-SIG	0.004	0.897	0.975	0.899	0.945	0.892	0.980	0.903	**0.000**	0.860
	WaNet	0.012	0.893	0.970	0.894	**0.000**	0.005	0.010	0.898	**0.000**	0.830
	Mean	0.008	0.893	0.983	0.895	0.798	0.740	0.826	0.899	**0.000**	0.847
Grand Mean (b)		0.041	0.728	0.482	0.823	0.377	0.344	0.600	0.830	**0.015**	0.703

Table 8: Performance under both-pool contamination ($\epsilon_b = 0.1,\ \epsilon_s = 0.1$) on the PL-2 regime. Bold indicates lowest ASR per row. GTSRB results are mean±std over 5 seeds. '–' indicates unavailable FST+PL results.

Data	Attack	FST ASR	FST C-ACC	FT ASR	FT C-ACC	I-BAU ASR	I-BAU C-ACC	FT+SAM ASR	FT+SAM C-ACC	FST+PL ASR	FST+PL C-ACC	PL (Ours) ASR	PL (Ours) C-ACC
(c) Both pools contamination $\epsilon_b = 0.1,\ \epsilon_s = 0.1$													
CIFAR-10	Blended	0.333	0.801	0.475	0.819	0.751	0.150	0.888	0.821	–	–	**0.049**	0.716
	Adaptive Blend	0.282	0.796	0.425	0.812	0.217	0.218	0.873	0.814	–	–	**0.033**	0.740
	SSBA	0.088	0.783	0.105	0.797	0.418	0.258	0.273	0.796	–	–	**0.012**	0.739
	BadNet	0.323	0.798	0.539	0.816	0.772	0.181	0.955	0.822	–	–	**0.004**	0.722
	WaNet	0.126	0.794	0.179	0.812	1.000	0.100	0.493	0.800	–	–	**0.026**	0.746
	TITIM-SIG	0.265	0.804	0.315	0.816	0.715	0.139	0.750	0.815	**0.012**	0.732	0.024	0.732
	Mean	0.236	0.796	0.340	0.813	0.645	0.174	0.722	0.811	**0.012**	0.732	0.025	0.733
GTSRB (5 seeds)	Blended	0.468±0.075	0.878±0.005	0.604±0.041	0.898±0.005	0.198±0.290	0.149±0.069	0.939±0.018	0.903±0.003	**0.003±0.004**	0.698±0.008	0.016±0.027	0.742±0.017
	Adaptive Blend	0.312±0.069	0.868±0.010	0.416±0.077	0.896±0.006	**0.006±0.013**	0.144±0.047	0.898±0.033	0.905±0.004	**0.006±0.007**	0.665±0.049	0.012±0.019	0.684±0.050
	SSBA	0.376±0.074	0.876±0.008	0.602±0.050	0.896±0.005	**0.003±0.005**	0.244±0.024	0.954±0.010	0.896±0.007	0.040±0.034	0.695±0.057	0.053±0.038	0.726±0.062
	BadNet	0.344±0.062	0.881±0.005	0.607±0.041	0.897±0.003	0.609±0.088	0.205±0.025	0.939±0.013	0.902±0.004	–	–	**0.006±0.005**	0.695±0.016
	TITIM-SIG	0.355±0.056	0.882±0.007	0.486±0.038	0.900±0.003	0.081±0.101	0.092±0.026	0.846±0.021	0.909±0.003	**0.023±0.022**	0.715±0.035	0.047±0.027	0.735±0.034
	WaNet	0.040±0.083	0.891±0.012	0.067±0.095	0.902±0.008	0.129±0.209	0.037±0.017	0.148±0.271	0.900±0.008	**0.001±0.002**	0.730±0.013	0.003±0.003	0.772±0.012
	Mean	0.316	0.879	0.464	0.898	0.171	0.145	0.787	0.903	**0.015**	0.701	0.023	0.726
Tiny-ImageNet	Blended	1.000	0.307	1.000	0.704	0.984	0.360	1.000	0.738	–	–	**0.000**	0.512
	Adaptive Blend	1.000	0.307	1.000	0.704	0.984	0.360	1.000	0.738	–	–	**0.000**	0.493
	SSBA	0.998	0.284	0.999	0.722	0.999	0.393	0.999	0.746	–	–	**0.000**	0.557
	BadNet	0.995	0.297	0.995	0.716	0.976	0.011	0.994	0.739	–	–	**0.000**	0.537
	TITIM-SIG	0.995	0.249	0.982	0.730	–	–	0.994	0.741	**0.000**	0.080	**0.000**	0.556
	WaNet	0.991	0.286	0.997	0.723	1.000	0.005	0.995	0.744	–	–	**0.002**	0.462
	Mean	0.997	0.288	0.996	0.717	0.989	0.226	0.997	0.741	**0.000**	0.080	**0.000**	0.520
CelebA	Blended	0.350	0.768	0.508	0.771	0.278	0.553	0.998	0.791	**0.001**	0.428	0.032	0.414
	Adaptive Blend	0.270	0.758	0.459	0.772	0.084	0.499	1.000	0.792	**0.072**	0.333	0.118	0.476
	TITIM-SIG	0.222	0.787	0.352	0.785	0.909	0.469	0.965	0.790	0.027	0.514	**0.017**	0.536
	SSBA	0.406	0.791	0.643	0.790	0.360	0.558	0.998	0.793	0.057	0.607	**0.010**	0.717
	Visible	0.461	0.787	0.744	0.792	0.947	0.603	0.996	0.792	**0.027**	0.661	0.137	0.672
	WaNet	0.510	0.786	0.656	0.790	1.000	0.276	0.973	0.791	0.248	0.643	**0.147**	0.633
	Mean	0.370	0.780	0.560	0.783	0.596	0.493	0.988	0.792	0.081	0.505	**0.067**	0.571
ImageNet-12	Blended	0.976	0.897	0.971	0.898	–	–	0.993	0.893	**0.000**	0.823	**0.000**	0.827
	Adaptive Blend	0.833	0.883	0.802	0.877	–	–	0.791	0.898	**0.000**	0.835	**0.000**	0.830
	SSBA	0.978	0.897	0.989	0.897	–	–	0.991	0.898	**0.000**	0.845	**0.000**	0.855
	BadNet	0.985	0.890	0.992	0.892	–	–	0.995	0.895	–	–	**0.000**	0.840
	TITIM-SIG	0.967	0.890	0.971	0.898	–	–	0.978	0.912	**0.000**	0.783	**0.000**	0.802
	WaNet	0.907	0.900	0.916	0.903	–	–	0.927	0.897	**0.002**	0.772	**0.002**	0.792
	Mean	0.941	0.893	0.940	0.894	–	–	0.946	0.899	**0.000**	0.812	**0.000**	0.824
Grand Mean (c)		0.567	0.727	0.652	0.821	0.569	0.256	0.888	0.829	**0.022**	0.566	0.023	0.675

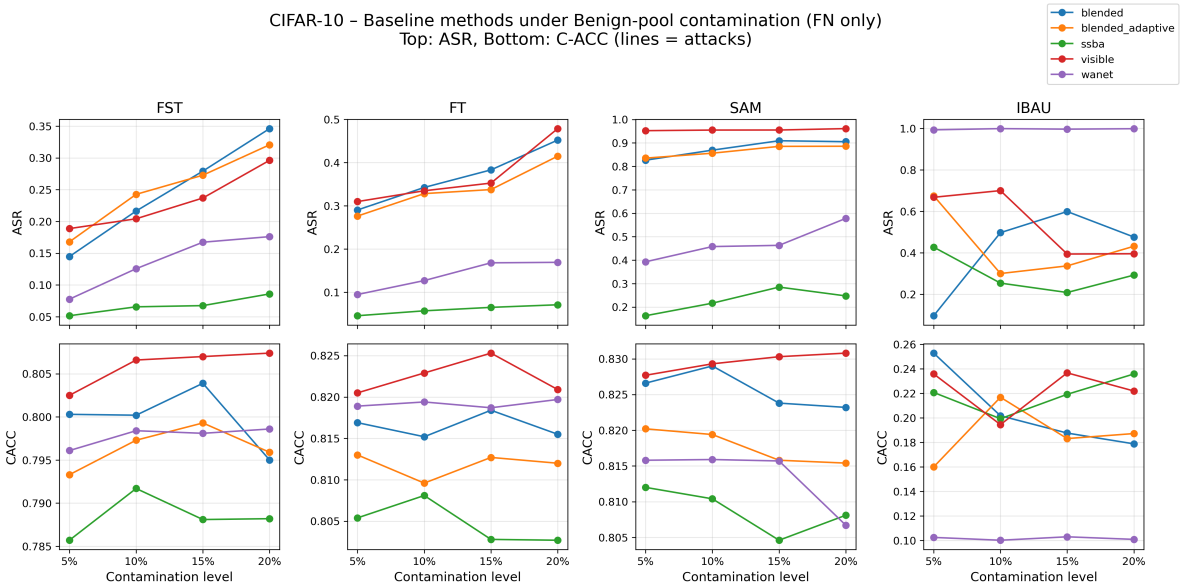

Figure 7: **Baseline under benign contamination (FN only).**

Table 9: Performance of PL compared with four baselines under contaminated fine-tuning data under the PL-5 regime, evaluated under three contamination settings. Bold values indicate the best performance (lowest ASR or highest C-ACC). PL consistently outperforms baselines on the safety-utility trade-off across all three settings.

		FST		FT		I-BAU		FT+SAM		PL (Ours)	
Data	Attack	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC
(a) Benign pool contamination $\epsilon_b = 0.1,\ \epsilon_s = 0$											
CIFAR-10	Blended	0.216	0.8	0.343	0.815	0.416	0.212	0.869	0.829	**0.063**	0.783
	Adaptive Blend	0.243	0.797	0.328	0.81	0.475	0.19	0.856	0.819	**0.09**	0.787
	SSBA	0.066	0.792	0.057	0.808	0.113	0.191	0.217	0.81	**0.019**	0.776
	BadNet	0.204	0.807	0.335	0.823	0.597	0.193	0.955	0.829	**0.05**	0.778
	WaNet	0.126	0.798	0.127	0.819	1	0.1	0.458	0.816	**0.037**	0.78
	Mean	0.171	0.799	0.238	0.815	0.52	0.177	0.671	0.821	**0.052**	0.781
GTSRB	Blended	0.3	0.881	0.437	0.903	**0.013**	0.17	0.927	0.917	0.173	0.875
	Adaptive Blend	0.393	0.872	0.502	0.893	**0.001**	0.193	0.95	0.914	0.212	0.858
	SSBA	0.296	0.879	0.567	0.897	**0.005**	0.162	0.975	0.9	0.117	0.893
	BadNet	0.322	0.911	0.495	0.914	**0.014**	0.109	0.941	0.917	0.12	0.89
	WaNet	0.107	0.894	0.161	0.909	0.864	0.018	0.651	0.905	**0.061**	0.877
	Mean	0.284	0.887	0.432	0.903	0.179	0.13	0.889	0.911	**0.137**	0.879
Tiny-ImageNet	Blended	1	0.413	1	0.688	1	0.005	1	0.744	**0**	0.696
	Adaptive Blend	1	0.413	1	0.688	**0**	0.003	1	0.744	**0**	0.696
	SSBA	0.999	0.41	0.999	0.719	1	0.005	1	0.757	**0**	0.715
	BadNet	0.994	0.406	0.994	0.716	0.985	0.004	0.994	0.756	**0**	0.734
	WaNet	0.986	0.413	0.986	0.726	**0**	0.005	0.068	0.746	**0**	0.628
	Mean	0.996	0.411	0.996	0.707	0.597	0.004	0.812	0.749	**0**	0.694
Grand Mean		0.483	0.699	0.555	0.809	0.432	0.104	0.791	0.827	**0.063**	0.784
(b) Suspicious pool contamination $\epsilon_b = 0,\ \epsilon_s = 0.1$											
CIFAR-10	Blended	0.03	0.802	0.193	0.821	0.45	0.227	0.306	0.827	**0.001**	0.616
	Adaptive Blend	0.049	0.793	0.173	0.817	0.173	0.137	0.231	0.82	**0.002**	0.568
	SSBA	0.028	0.785	0.035	0.809	0.584	0.196	0.065	0.805	**0.014**	0.747
	BadNet	0.081	0.803	0.376	0.822	0.176	0.251	0.805	0.829	**0.004**	0.63
	WaNet	0.035	0.8	0.074	0.819	0.986	0.118	0.209	0.818	**0.007**	0.704
	Mean	0.045	0.797	0.17	0.818	0.474	0.186	0.323	0.82	**0.006**	0.653
GTSRB	Blended	0.001	0.903	0.017	0.917	**0**	0.071	0.297	0.919	**0**	0.671
	Adaptive Blend	**0**	0.891	0.276	0.904	**0**	0.143	0.803	0.911	**0**	0.575
	SSBA	**0**	0.896	**0**	0.905	**0**	0.049	0.001	0.903	**0**	0.514
	BadNet	**0**	0.894	0.016	0.917	0.005	0.113	0.203	0.916	0.009	0.551
	WaNet	0.002	0.888	0.017	0.902	**0**	0.047	0.077	0.902	**0**	0.765
	Mean	**0.001**	0.894	0.065	0.909	0.001	0.085	0.276	0.91	0.002	0.615
Tiny-ImageNet	Blended	**0**	0.383	0.951	0.712	**0**	0.005	0.981	0.756	**0**	0.56
	Adaptive Blend	**0**	0.383	0.951	0.712	**0**	0.005	0.981	0.756	**0**	0.56
	SSBA	**0**	0.384	0.979	0.714	**0**	0.005	0.989	0.753	**0**	0.616
	BadNet	**0**	0.384	0.988	0.721	**0**	0.005	0.992	0.758	**0**	0.618
	WaNet	**0**	0.413	0.399	0.702	**0**	0.006	0.762	0.753	**0**	0.282
	Mean	**0**	0.389	0.854	0.712	**0**	0.005	0.941	0.755	**0**	0.527
Grand Mean		0.015	0.693	0.363	0.813	0.158	0.092	0.513	0.828	**0.002**	0.598
(c) Both pools contamination $\epsilon_b = 0.1,\ \epsilon_s = 0.1$											
CIFAR-10	Blended	0.312	0.794	0.442	0.819	0.695	0.163	0.83	0.828	**0.103**	0.644
	Adaptive Blend	0.329	0.792	0.44	0.812	0.496	0.153	0.851	0.816	**0.118**	0.652
	SSBA	0.078	0.79	0.097	0.808	0.17	0.163	0.258	0.805	**0.031**	0.717
	BadNet	0.289	0.803	0.515	0.824	0.89	0.147	0.955	0.827	**0.121**	0.656
	WaNet	0.179	0.796	0.197	0.82	0.999	0.102	0.515	0.811	**0.06**	0.734
	Mean	0.237	0.795	0.338	0.817	0.65	0.146	0.682	0.817	**0.087**	0.681
GTSRB	Blended	0.442	0.908	0.607	0.918	**0.006**	0.155	0.937	0.923	0.023	0.663
	Adaptive Blend	0.474	0.897	0.617	0.905	0.011	0.138	0.935	0.914	**0.001**	0.488
	SSBA	0.283	0.882	0.649	0.9	0.017	0.227	0.974	0.905	**0**	0.517
	BadNet	0.684	0.918	0.793	0.923	0.152	0.164	0.948	0.924	**0.002**	0.614
	WaNet	0.168	0.891	0.246	0.906	0.736	0.012	0.659	0.903	**0.045**	0.741
	Mean	0.41	0.899	0.582	0.91	0.184	0.139	0.891	0.914	**0.014**	0.605
Tiny-ImageNet	Blended	1	0.403	1	0.719	0.991	0.026	1	0.756	**0**	0.546
	Adaptive Blend	1	0.403	1	0.719	0.991	0.026	1	0.756	**0**	0.546
	SSBA	0.999	0.415	1	0.715	1	0.005	1	0.757	**0**	0.579
	BadNet	0.994	0.397	0.995	0.725	0.976	0.005	0.994	0.761	**0.001**	0.559
	WaNet	0.993	0.406	0.992	0.713	**0**	0.005	0.1	0.748	**0**	0.563
	Mean	0.997	0.405	0.997	0.718	0.792	0.013	0.819	0.756	**0**	0.559
Grand Mean		0.548	0.7	0.639	0.815	0.542	0.099	0.797	0.829	**0.034**	0.615

Table 10: Cross-tuning evaluation of PL under Clean pool contamination: $\epsilon_b = 0.1,\ \epsilon_s = 0$ (False Negatives only) with Train_prop = 5%. Columns denote evaluation (base) attack A; rows denote fine-tuning attack B. All results are reported under the known-target setting, where the suspicious pool target class matches the true backdoor target class. Each cell reports C-ACC and ASR.

		Evaluation Attack (A)									
		Blended		Adaptive Blend		SSBA		Visible		WaNet	
Data	Fine-tune (B)	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR
CIFAR-10	Blended	0.787	0.014	0.787	0.014	0.763	0.022	0.790	0.630	0.780	0.021
	Adaptive Blend	0.787	0.014	0.787	0.014	0.763	0.022	0.790	0.630	0.780	0.021
	SSBA	0.801	0.015	0.801	0.015	0.776	0.019	0.796	0.585	0.785	0.028
	Visible	0.797	0.008	0.797	0.008	0.768	0.026	0.778	0.050	0.783	0.014
	WaNet	0.798	0.012	0.798	0.012	0.770	0.036	0.786	0.222	0.780	0.008
GTSRB	Blended	0.880	0.000	0.880	0.000	0.868	0.877	0.879	0.046	0.872	0.000
	Adaptive Blend	0.880	0.000	0.880	0.000	0.868	0.877	0.879	0.046	0.872	0.000
	SSBA	0.894	0.060	0.894	0.060	0.893	0.117	0.896	0.000	0.886	0.000
	Visible	0.883	0.000	0.883	0.000	0.878	0.000	0.890	0.120	0.886	0.001
	WaNet	0.896	0.000	0.896	0.000	0.892	0.000	0.900	0.000	0.877	0.000
Tiny-ImageNet	Blended	0.696	0.000	0.696	0.000	0.686	0.000	0.697	0.937	0.698	0.002
	Adaptive Blend	0.696	0.000	0.696	0.000	0.686	0.000	0.697	0.937	0.698	0.002
	SSBA	0.673	0.002	0.673	0.002	0.698	0.000	0.686	0.109	0.680	0.002
	Visible	0.643	0.006	0.643	0.006	0.613	0.062	0.703	0.000	0.512	0.002
	WaNet	0.529	0.000	0.529	0.000	0.574	0.665	0.547	0.986	0.688	0.000

Table 11: Cross-tuning evaluation of PL under Suspicious pool contamination: $\epsilon_b = 0,\ \epsilon_s = 0.1$ (False Positives only) with Train_prop = 5%. Columns denote evaluation (base) attack A; rows denote fine-tuning attack B. All results are reported under the known-target setting, where the suspicious pool target class matches the true backdoor target class. Each cell reports C-ACC and ASR.

		Evaluation Attack (A)									
		Blended		Adaptive Blend		SSBA		Visible		WaNet	
Data	Fine-tune (B)	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR
CIFAR-10	Blended	0.743	0.013	0.743	0.013	0.717	0.041	0.738	0.903	0.698	0.250
	Adaptive Blend	0.743	0.013	0.743	0.013	0.717	0.041	0.738	0.903	0.698	0.250
	SSBA	0.742	0.100	0.742	0.100	0.747	0.014	0.760	0.508	0.742	0.143
	Visible	0.724	0.137	0.724	0.137	0.726	0.069	0.629	0.004	0.709	0.130
	WaNet	0.724	0.163	0.724	0.163	0.729	0.073	0.763	0.330	0.704	0.014
GTSRB	Blended	0.816	0.000	0.816	0.000	0.812	0.000	0.836	0.114	0.677	0.419
	Adaptive Blend	0.816	0.000	0.816	0.000	0.812	0.000	0.836	0.114	0.677	0.419
	SSBA	0.846	0.003	0.846	0.003	0.514	0.000	0.847	0.007	0.799	0.113
	Visible	0.819	0.049	0.819	0.049	0.818	0.000	0.551	0.009	0.821	0.016
	WaNet	0.861	0.002	0.861	0.002	0.853	0.000	0.863	0.000	0.765	0.010
Tiny-ImageNet	Blended	0.560	0.000	0.560	0.000	0.559	0.000	0.531	0.001	0.559	0.051
	Adaptive Blend	0.560	0.000	0.560	0.000	0.559	0.000	0.531	0.001	0.559	0.051
	SSBA	0.564	0.020	0.564	0.020	0.544	0.000	0.575	0.002	0.560	0.010
	Visible	0.546	0.001	0.546	0.001	0.574	0.125	0.573	0.001	0.519	0.003
	WaNet	0.580	0.003	0.580	0.003	0.573	0.002	0.571	0.001	0.631	0.000

Table 12: Cross-tuning evaluation of PL under both-pool contamination: $\epsilon_b = 0.1$, $\epsilon_s = 0.1$ (FN+FP) with Train_prop = 5%. Columns denote evaluation (base) attack A; rows denote fine-tuning attack B. All results are reported under the known-target setting, where the suspicious pool target class matches the true backdoor target class. Each cell reports C-ACC and ASR.

		Evaluation Attack (A)									
		Blended		Adaptive Blend		SSBA		Visible		WaNet	
Data	Fine-tune (B)	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR
CIFAR-10	Blended	0.679	0.122	0.679	0.122	0.699	0.150	0.703	0.964	0.635	0.554
	Adaptive Blend	0.679	0.122	0.679	0.122	0.699	0.150	0.703	0.964	0.635	0.554
	SSBA	0.706	0.253	0.706	0.253	0.717	0.031	0.741	0.749	0.719	0.255
	Visible	0.672	0.296	0.672	0.296	0.696	0.233	0.656	0.121	0.686	0.262
	WaNet	0.666	0.314	0.666	0.314	0.700	0.157	0.731	0.904	0.734	0.027
GTSRB	Blended	0.821	0.000	0.821	0.000	0.826	0.901	0.802	0.199	0.682	0.397
	Adaptive Blend	0.821	0.000	0.821	0.000	0.826	0.901	0.802	0.199	0.682	0.397
	SSBA	0.820	0.116	0.820	0.116	0.517	0.000	0.830	0.728	0.778	0.331
	Visible	0.822	0.001	0.822	0.001	0.808	0.079	0.614	0.002	0.810	0.006
	WaNet	0.833	0.130	0.833	0.130	0.827	0.000	0.840	0.000	0.741	0.004
Tiny-ImageNet	Blended	0.546	0.000	0.546	0.000	0.551	0.000	0.575	0.006	0.511	0.008
	Adaptive Blend	0.546	0.000	0.546	0.000	0.551	0.000	0.575	0.006	0.511	0.008
	SSBA	0.566	0.000	0.566	0.000	0.579	0.000	0.576	0.000	0.576	0.005
	Visible	0.532	0.000	0.532	0.000	0.565	0.707	0.559	0.001	0.518	0.017
	WaNet	0.445	0.000	0.445	0.000	0.485	0.579	0.541	0.990	0.615	0.001

Table 13: GTSRB cross-tuning results with per-seed variability (mean±std over 5 seeds) under perfect partition ($\epsilon_b = \epsilon_s = 0$) with Train_prop=5%. Columns denote evaluation attack A; rows denote fine-tuning attack B. Corresponds to the GTSRB block of Table 4 in the main text, which reports means only.

		Evaluation Attack (A)											
		Blended		Adaptive Blend		SSBA		Visible		WaNet		TITIM-SIG	
Data	Fine-tune (B)	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR
GTSRB	Blended	$0.880_{\pm 0.009}$	$0.000_{\pm 0.000}$	$0.881_{\pm 0.005}$	$0.000_{\pm 0.000}$	$0.880_{\pm 0.005}$	$0.000_{\pm 0.000}$	$0.888_{\pm 0.007}$	$0.000_{\pm 0.000}$	$0.880_{\pm 0.005}$	$0.000_{\pm 0.000}$	$0.898_{\pm 0.002}$	$0.000_{\pm 0.000}$
	Adaptive Blend	$0.880_{\pm 0.009}$	$0.000_{\pm 0.000}$	$0.881_{\pm 0.005}$	$0.000_{\pm 0.000}$	$0.880_{\pm 0.005}$	$0.000_{\pm 0.000}$	$0.888_{\pm 0.007}$	$0.000_{\pm 0.000}$	$0.880_{\pm 0.005}$	$0.000_{\pm 0.000}$	$0.898_{\pm 0.002}$	$0.000_{\pm 0.000}$
	SSBA	$0.897_{\pm 0.009}$	$0.060_{\pm 0.101}$	$0.894_{\pm 0.005}$	$0.002_{\pm 0.003}$	$0.872_{\pm 0.005}$	$0.001_{\pm 0.001}$	$0.894_{\pm 0.009}$	$0.000_{\pm 0.001}$	$0.899_{\pm 0.005}$	$0.000_{\pm 0.000}$	$0.901_{\pm 0.003}$	$0.000_{\pm 0.000}$
	Visible	$0.899_{\pm 0.006}$	$0.001_{\pm 0.001}$	$0.900_{\pm 0.005}$	$0.000_{\pm 0.000}$	$0.885_{\pm 0.008}$	$0.000_{\pm 0.000}$	$0.897_{\pm 0.004}$	$0.000_{\pm 0.000}$	$0.898_{\pm 0.003}$	$0.000_{\pm 0.000}$	$0.904_{\pm 0.004}$	$0.000_{\pm 0.000}$
	WaNet	$0.897_{\pm 0.006}$	$0.000_{\pm 0.001}$	$0.902_{\pm 0.005}$	$0.001_{\pm 0.002}$	$0.890_{\pm 0.005}$	$0.001_{\pm 0.001}$	$0.897_{\pm 0.005}$	$0.022_{\pm 0.049}$	$0.905_{\pm 0.003}$	$0.000_{\pm 0.000}$	$0.910_{\pm 0.005}$	$0.000_{\pm 0.000}$
	TITIM-SIG	$0.889_{\pm 0.007}$	$0.001_{\pm 0.001}$	$0.884_{\pm 0.004}$	$0.000_{\pm 0.000}$	$0.876_{\pm 0.006}$	$0.000_{\pm 0.000}$	$0.874_{\pm 0.003}$	$0.000_{\pm 0.000}$	$0.886_{\pm 0.006}$	$0.000_{\pm 0.000}$	$0.868_{\pm 0.011}$	$0.000_{\pm 0.000}$

Table 14: Extended cross-tuning evaluation of PL (all-candidate) on CelebA and ImageNet-12 under perfect partition ($\epsilon_b = \epsilon_s = 0$) with Train_prop=5%. Columns denote evaluation attack A; rows denote fine-tuning attack B. Complements Table 4 in the main text.

		Evaluation Attack (A)											
		Blended		Adaptive Blend		SSBA		Visible		WaNet		TITIM-SIG	
Data	Fine-tune (B)	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR
CelebA	Blended	0.731	0.170	0.744	0.103	0.707	0.030	0.725	0.046	0.730	0.064	0.754	0.052
	Adaptive Blend	0.731	0.170	0.744	0.103	0.707	0.030	0.725	0.046	0.730	0.064	0.754	0.052
	SSBA	0.760	0.247	0.766	0.174	0.750	0.075	0.757	0.114	0.771	0.066	0.780	0.048
	Visible	0.784	0.300	0.776	0.113	0.776	0.077	0.747	0.104	0.763	0.104	0.779	0.109
	WaNet	0.777	0.248	0.768	0.117	0.781	0.057	0.770	0.075	0.764	0.061	0.781	0.089
	TITIM-SIG	0.781	0.316	0.780	0.126	0.780	0.122	0.775	0.114	0.781	0.081	0.776	0.075
ImageNet-12	Blended	0.887	0.000	0.867	0.002	0.877	0.722	0.882	0.995	0.893	0.005	0.885	0.544
	Adaptive Blend	0.887	0.000	0.867	0.002	0.877	0.722	0.882	0.995	0.893	0.005	0.885	0.544
	SSBA	0.863	0.978	0.877	0.005	0.905	0.000	0.868	0.982	0.870	0.005	0.865	0.322
	Visible	0.860	0.964	0.897	0.007	0.860	0.098	0.892	0.993	0.882	0.005	0.865	0.447
	WaNet	0.842	0.884	0.870	0.000	0.853	0.115	0.857	0.902	0.863	0.000	0.840	0.209
	TITIM-SIG	0.872	0.965	0.882	0.007	0.853	0.109	0.892	0.995	0.885	0.007	0.897	1.000

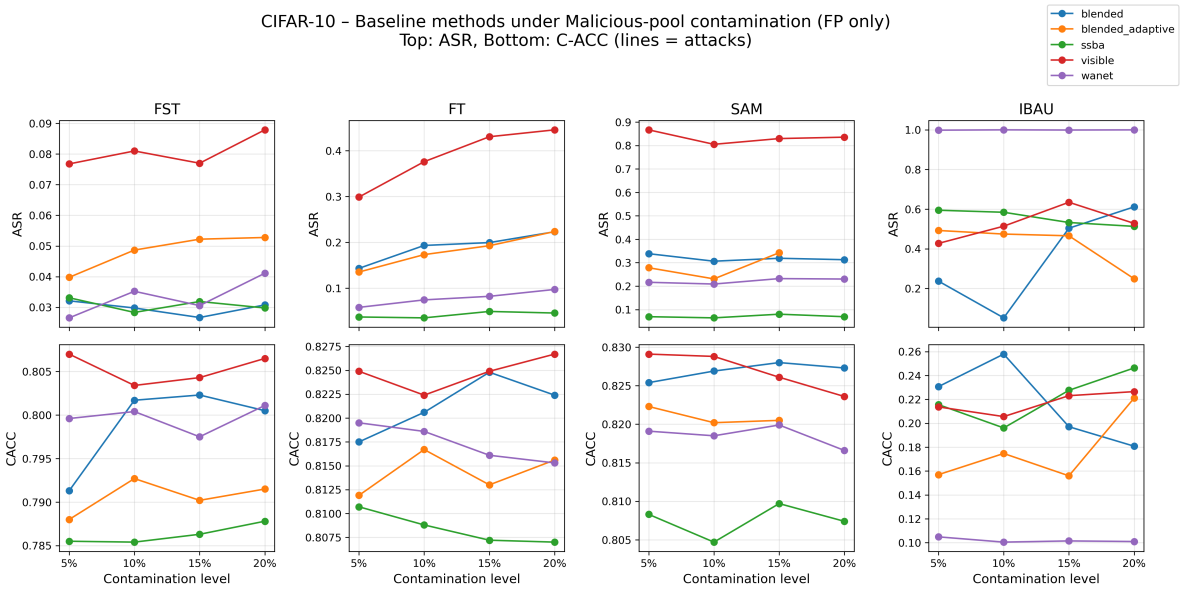

Figure 8: **Baseline under suspicious contamination (FP only).**

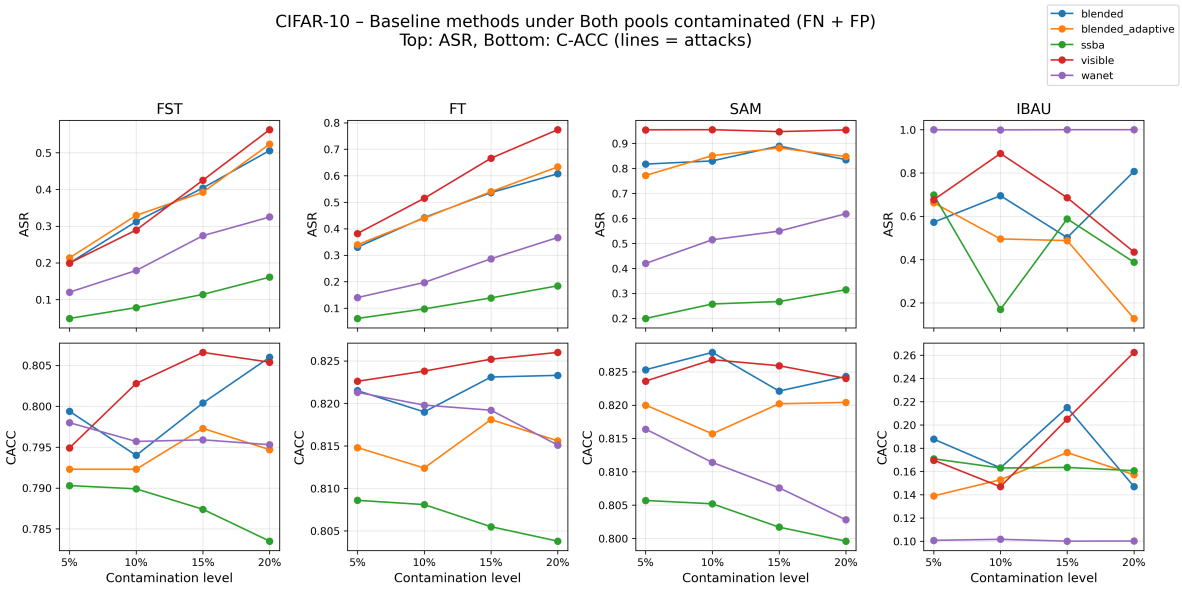

Figure 9: **Baseline under combined contamination (FN+FP).**