GARF: GRADIENT-ALIGNED REACTIVATION FORGETTING FOR BACKDOOR DEFENSE

Anonymous authors
Paper under double-blind review

ABSTRACT

Backdoor defenses often suppress visible trigger behavior, yet leave latent backdoor features that can be reactivated by small, adversarially optimized perturbations. We study this problem in a realistic post-training setting where the defender only has access to a possibly backdoored model, without any poisoned data. We propose Gradient-Aligned Reactivation Forgetting (GARF), a bilevel post-training defense that treats the parameter difference between the original backdoored model and its defended counterpart as a proxy backdoor task direction in parameter space. In the lower level, GARF learns a small input-agnostic perturbation on clean data whose induced parameter gradients are encouraged to align with this direction, thereby simulating an adaptive reactivation attack using only accessible information. In the upper level, the model is updated to jointly reduce the clean loss and a reactivation loss evaluated under the learned perturbation, forming an iterative attack–defense process that repeatedly exposes and suppresses reactivation-sensitive directions. Empirically, GARF reduces the white-box reactivation attack success rate by over 50% on average and keeps the black-box reactivation ASR below 10%, while preserving competitive clean accuracy across multiple benchmark datasets and diverse attack types.

1 INTRODUCTION

Deep neural networks have achieved impressive performance across a wide range of tasks (Yatbaz et al., 2023; Panagoulias et al., 2024; Shu et al., 2024), yet remain vulnerable to backdoor attacks. In these attacks, adversaries poison a small fraction of training samples to implant hidden trigger–label associations (Gu et al., 2019; Wu et al., 2025a; 2026), causing the model to misbehave whenever a specific trigger appears. This poses serious safety risks in security-critical domains such as autonomous driving, where even a single malicious prediction can be catastrophic. A fundamental defense objective is therefore to eliminate the influence of backdoors on the learned model.

A variety of defenses have been proposed at different stages of the machine-learning pipeline (Wu et al., 2025b). We focus on *post-training* defenses, which operate on a trained model without modifying the training procedure, including neuron pruning (Wu and Wang, 2021; Liu et al., 2018a), fine-tuning (Zhu et al., 2023; Li et al., 2021b), and adversarial unlearning (Wang et al., 2019; Zeng et al., 2022; Wei et al., 2023). These methods share an implicit assumption: once trigger activation is weakened, the backdoor is essentially removed. Recent work on backdoor reactivation (Zhu et al., 2024a) shows that this assumption is false. Even when a model appears sanitized, an adaptive adversary can craft small perturbations that revive the hidden trigger–label association. Because defenders lack poisoned samples, they cannot adequately explore or suppress reactivation-relevant regions, leaving deeper vulnerabilities untouched. Consequently, many defended models still exhibit high reactivation success rates under both white-box and black-box settings. As shown in Fig. 1, trigger amplification raises the white-box ASR of SAU- and FST-defended BadNets models from around 1% under the original trigger to above 90%. This underscores the need for defenses that explicitly address the *adaptive* nature of reactivation rather than assuming a static backdoor.

To address adaptive reactivation, we propose Gradient-Aligned Reactivation Forgetting (GARF), a post-training defense that formulates forgetting as a bilevel optimization problem. In the lower level, GARF simulates an adaptive attacker by learning a small input-agnostic perturbation on clean data. In the upper level, the model is updated to reduce both the standard clean loss and the loss

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

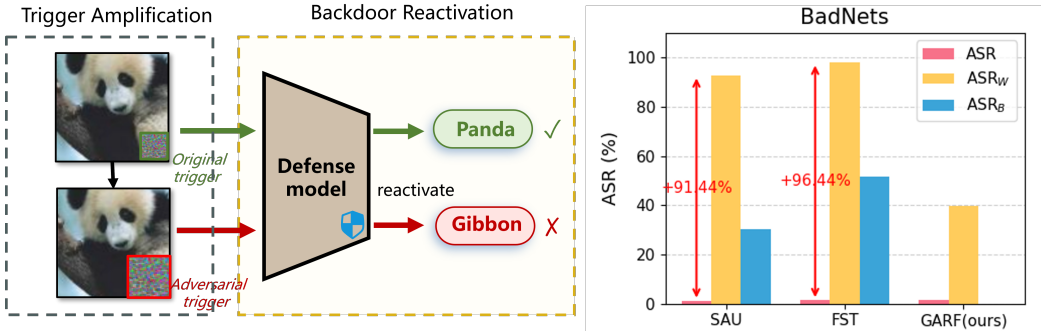

Figure 1: Trigger amplification–based backdoor reactivation on BadNets. An attacker optimizes an adversarial trigger from the original trigger to restore target-label predictions after defense. For SAU and FST, white-box reactivation ASR reaches 92.74% and 97.90%, respectively, while black-box reactivation ASR reaches 30.34% and 51.79%. GARF reduces reactivation ASR to 39.67% and 0.01% in the white-box and black-box settings, respectively.

under this perturbation, thereby suppressing the vulnerabilities exposed by the lower level. A key challenge is that the defender has no access to poisoned samples and therefore cannot directly probe the model's behaviour around poisoned samples. Our intuition is that the relationship between perturbation-induced gradients and the direction toward the original backdoored model can help identify reactivation-relevant perturbations. We therefore use the parameter difference as a proxy for the backdoor task direction in parameter space; this proxy depends only on model parameters and requires no poisoned samples. GARF augments the lower-level objective with a gradient-alignment term that uses the relationship between the induced loss gradients and this proxy direction to guide the perturbation search. This steers the perturbation search toward reactivation-relevant directions, yielding a stronger surrogate attacker than a naive clean-data surrogate. Our theoretical analysis characterizes the local first-order correction induced by gradient alignment under the first-order outer-update approximation. Under the stated regularity and backdoor sensitivity assumptions, and for sufficiently small positive alignment weights β, GARF's outer gradient has a larger unnormalized inner product with the true reactivation gradient than the base surrogate gradient (Appendix B.2). This establishes a conditional local geometric improvement in the update direction; reductions in reactivation risk and attack success are assessed empirically. On CIFAR-10, GARF lowers the mean white-box reactivation ASR from 85.78–98.11% for the compared defenses to 29.33%, a reduction of 56.45–68.78 percentage points, and achieves a mean black-box reactivation ASR of 3.52% while preserving competitive clean accuracy.

Contributions. Our work makes the following contributions: (1) We propose GARF, a gradient-aligned surrogate attacker that uses the parameter gap between backdoored and defended models as a proxy backdoor direction, thereby steering input-agnostic perturbations toward reactivation-relevant directions without requiring poisoned samples. (2) We conduct extensive experiments showing that GARF effectively suppresses backdoor reactivation, reducing mean white-box reactivation ASR by over 50 percentage points relative to the compared defenses on CIFAR-10 while maintaining competitive clean accuracy.

2 Related Work

Backdoor Attack. Backdoor attacks embed malicious trigger–label associations into models during training, allowing models to behave normally on clean inputs yet misclassify triggered ones. BadNets (Gu et al., 2019) first introduced this concept by selecting a subset of samples, adding a visible patch, and relabeling them to a target class. Subsequent work proposed techniques to enhance stealthiness and success. Blended (Chen et al., 2017) fuses triggers through image blending, SSBA (Li et al., 2021c) injects imperceptible watermarks via neural optimization, and LF (Zeng et al., 2021) modifies samples subtly in the frequency domain. Label-consistent (LC) attacks (Turner et al., 2019) preserve original labels, thereby evading label-based anomaly detection. TaCT (Tang et al., 2021) and Adap-Blend (Qi et al., 2023a) further increase stealthiness by minimizing the distributional gap between poisoned and clean samples. Recent work on backdoor reactivation (Zhu et al., 2024a)

shows that defended models can recover backdoor behavior under a small universal perturbation learned on poisoned training samples and evaluated on triggered test samples. This line of work highlights that backdoor threats extend beyond the initial poisoning stage, posing continuous risks to defended models.

Backdoor Defense. Extensive research has been devoted to mitigating backdoor threats (Wu et al., 2025b; Niu et al., 2024; Amula et al., 2025; Wei et al., 2025). Depending on when they intervene, defenses can be categorized into pre-processing, in-training, post-training, and inference-stage methods. Pre-processing defenses (Chen et al., 2022; Zhu et al., 2024b) and in-training defenses (Li et al., 2021a) operate before and during training, respectively, while inference-stage defenses (Guo et al., 2023) detect triggers at test time. Post-training defenses (Wang et al., 2019; Li et al., 2021b; Lin et al., 2025) assume access only to a pre-trained backdoored model and a small trusted set, and typically prune suspicious neurons (Liu et al., 2018a; Zheng et al., 2022), fine-tune to unlearn malicious behavior (Zeng et al., 2022; Zhu et al., 2023), or use adversarial optimization as in I-BAU, PBE, and SAU (Zeng et al., 2022; Mu et al., 2023; Wei et al., 2023). Backdoor trigger inversion (BTI) offers a complementary post-training strategy by reconstructing attack patterns directly from a backdoored model. BTI-DBF (Xu et al., 2024) estimates the target label and synthesizes pseudo-poisoned samples, but the recovered poisons are still imperfect, so directly fine-tuning on them cannot fully erase the backdoor (Min et al., 2024b). Closer to our focus on reactivation, PAM (Min et al., 2024b) show that many purification methods achieve only *superficial safety*: their "purified" models can quickly regain high ASR under retuning and query-based reactivation attacks, and they propose Path-Aware Minimization (PAM) to push models farther along backdoor-connected paths. In contrast, we model adaptive reactivation attacks that optimize perturbations against the defended model and develop a bilevel post-training defense that explicitly suppresses these reactivation directions.

3 METHOD

3.1 PROBLEM SETTING

Notation. We consider a K-class classification task. A neural network $f_{\boldsymbol{\theta}}$ maps an input $\mathbf{x} \in X$ to a probability vector $f_{\boldsymbol{\theta}}(\mathbf{x})$, and the predicted label is $\hat{y} = \arg\max_k f_{\boldsymbol{\theta}}(\mathbf{x})_k$. We denote by $L(\boldsymbol{\theta}; \mathbf{x}, y)$ the standard supervised loss on a labeled example $(\mathbf{x}, y)$, instantiated as the cross-entropy loss in all experiments. We use $D_p = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_p}$ to denote the poisoned training set, which is obtained by mixing clean images with poisoned samples. We write τ for a trigger pattern and $g(\mathbf{x}, \tau)$ for the poisoning operator that puts τ on a clean image $\mathbf{x}$. The resulting backdoor image is denoted by $\tilde{\mathbf{x}} = g(\mathbf{x}, \tau)$. The subset of poisoned samples that actually contain the trigger is $D_{\mathrm{bd}} = \{(\tilde{\mathbf{x}}_i, y_t)\} \subset D_p$, where $y_t \in \mathcal{Y}$ is the attacker-chosen target label. Target-label detection and related ablation studies are reported in Appendix E.2. We use $D_c = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_c}$ to denote a clean dataset whose samples are drawn from the benign data distribution and contain no triggers. For an input $\mathbf{x}$, we denote by $S(\mathbf{x})$ the admissible perturbation set around $\mathbf{x}$, for example an ℓ_p-ball of radius ε in the input space: $S = \{\boldsymbol{\delta} \in \mathbb{R}^d : \|\boldsymbol{\delta}\|_p \le \varepsilon\}$.

Threat Model. We consider a backdoor setting with two stages: a poisoning phase and a reactivation phase. In the poisoning phase, the attacker injects a trigger τ into a small fraction of training samples via $g(\mathbf{x}, \tau)$, relabels them to a target y_t, and trains a backdoored model $f_{\boldsymbol{\theta}_b}$. In the reactivation phase, the attacker optimizes an input perturbation against the fixed final defended model $f_{\boldsymbol{\theta}}$. The proxy changes during defense optimization, whereas the model parameters remain fixed during post-defense reactivation; the attacker does not modify the defense updates in this setting. The attacker may have white-box access, where it can inspect $f_{\boldsymbol{\theta}}$ and compute gradients, or black-box access, where it can only query $f_{\boldsymbol{\theta}}$. The defender does not observe poisoned samples, or trigger patterns. Tab. 1 summarizes the knowledge of each party.

Table 1: Knowledge of each party.

	Adv.	Def.
Model $f_{\boldsymbol{\theta}}$	✓	$\boldsymbol{\theta}_b$ only
Poisoned data D_p	✓	×
Trigger τ	✓	×

Defender's Goal and Risks. The defender starts from a backdoored model $f_{\boldsymbol{\theta}_b}$ and a clean dataset D_c as reference, without knowing the poisoning process, the trigger pattern τ, or the backdoor subset

D_{bd}. The goal is to obtain new parameters $\boldsymbol{\theta}$ that simultaneously: (i) maintain high accuracy on benign data and (ii) are robust to reactivation attacks, meaning that hidden backdoor behaviors cannot be easily re-triggered by small perturbations. We formulate reactivation risk using a target-cross-entropy attacker:

$$R_{\mathrm{react}}(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim D_c}\big[L(\boldsymbol{\theta}; \tilde{\mathbf{x}} + \boldsymbol{\delta}^\star(\boldsymbol{\theta}), y)\big], \tag{1}$$

$$\boldsymbol{\delta}^\star(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\delta}\in S} \mathbb{E}_{(\mathbf{x},y)\sim D_c}\big[L(\boldsymbol{\theta}; \tilde{\mathbf{x}} + \boldsymbol{\delta}, y_t)\big], \tag{2}$$

where for each $(\mathbf{x}, y) \sim D_c$ we write $\tilde{\mathbf{x}} = g(\mathbf{x}, \tau)$ for its conceptual poisoned counterpart. The standard performance on benign data is measured by the clean risk

$$R_{\mathrm{clean}}(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim D_c}\big[L(\boldsymbol{\theta}; \mathbf{x}, y)\big]. \tag{3}$$

The defender therefore aims to learn parameters $\boldsymbol{\theta}$ that achieve both low $R_{\mathrm{clean}}(\boldsymbol{\theta})$ and low $R_{\mathrm{react}}(\boldsymbol{\theta})$ in the presence of an adaptive reactivation attacker.

3.2 Bilevel Defense against Backdoor Reactivation

Limitation and Naive Surrogate on Clean Data. Directly minimizing $R_{\mathrm{react}}(\boldsymbol{\theta})$ is not feasible in the post-training setting. It depends on conceptual poisoned samples $\tilde{\mathbf{x}} = g(\mathbf{x}, \tau)$ and the unknown poisoned subset D_{bd}, which are not available to the defender. Instead, we consider a surrogate objective on clean data that searches for a universal perturbation.

A straightforward surrogate is to explicitly search for a universal perturbation on clean data. We introduce

$$\boldsymbol{\delta}_B(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\delta}\in S} \mathbb{E}_{(\mathbf{x},y)\sim D_c}\big[L(\boldsymbol{\theta}; \mathbf{x} + \boldsymbol{\delta}, y_t)\big], \tag{4}$$

and define the surrogate reactivation loss

$$\hat{R}_{\mathrm{react}}(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim D_c}\big[L(\boldsymbol{\theta}; \mathbf{x} + \boldsymbol{\delta}_B(\boldsymbol{\theta}), y)\big]. \tag{5}$$

The resulting bilevel objective is

$$\min_{\boldsymbol{\theta}} \; R_{\mathrm{clean}}(\boldsymbol{\theta}) + \lambda \hat{R}_{\mathrm{react}}(\boldsymbol{\theta}), \quad \text{s.t. } \boldsymbol{\delta}_B(\boldsymbol{\theta}) \text{ is provided by Eq. (4).} \tag{6}$$

The empirical defense procedure obtains the target label required by the inner objective from the detector ensemble described in Appendix E.2. The main experimental results use these detected labels. Given the detected label, the surrogate is optimized on the clean dataset D_c, with its inner problem constructing a universal adversarial perturbation from clean inputs (Wei et al., 2023; Zeng et al., 2022).

However, such a universal perturbation is overly generic: during defense, as backdoor sensitivity is progressively suppressed, optimizing on clean samples increasingly exploits decision-boundary weaknesses unrelated to the hidden trigger, causing a mismatch between $R_{\mathrm{react}}(\boldsymbol{\theta})$ and its surrogate $\hat{R}_{\mathrm{react}}(\boldsymbol{\theta})$ which is the main defect of this naive bilevel formulation.

Intuition from Gradient Directions. Prior defenses exploit backdoor-related sensitivity already present in a backdoored model. ANP (Wu and Wang, 2021) shows that adversarial neuron perturbations can expose target-label behavior on clean inputs, while PAM (Min et al., 2024b) relates post-purification vulnerability to insufficient displacement along a backdoor-connected path. These observations motivate using gradients to guide the search for perturbations that expose suppressed backdoor behavior. Drawing also on the task-vector view of model differences (Ilharco et al., 2023), we use the displacement $\boldsymbol{\theta}_b - \boldsymbol{\theta}$ from the current defended model toward the original backdoored model as a proxy backdoor task direction, without requiring poisoned samples.

We examine this direction empirically in Appendix E.4. Along the interpolation $\boldsymbol{\theta}(t) = \boldsymbol{\theta} + t(\boldsymbol{\theta}_b - \boldsymbol{\theta})$, ASR increases from 1.48% to 93.79%, while clean accuracy stays between 89.41% and 91.82%. Corrupting the proxy with increasing noise also substantially weakens reactivation suppression. Together, these results support the use of the displacement as an informative direction for backdoor reactivation in the evaluated setting. GARF uses the relationship between the induced gradients and this proxy to guide the inner perturbation search, as illustrated in Fig. 2.

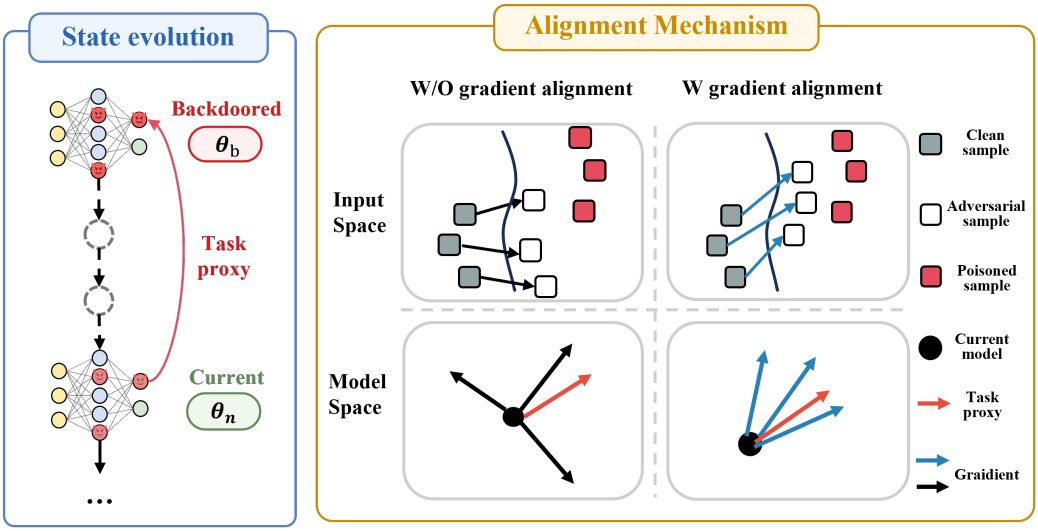

Figure 2: **Intuition of gradient alignment in GARF. Left:** State evolution from $\boldsymbol{\theta}_b$ to $\boldsymbol{\theta}_n$, with the displacement $\boldsymbol{\theta}_b - \boldsymbol{\theta}$ serving as a proxy backdoor direction. **Right:** Schematic illustration of the inner perturbation search; the model-space arrows illustrate target-loss parameter descent directions. Gradient alignment uses the relationship between these gradients and the proxy to guide the search toward backdoor-sensitive behavior. Empirical analysis of the proxy is provided in Appendix E.4.

Gradient-Consistent Bilevel Defense. Motivated by this observation, we guide the lower perturbation search using the relationship between the induced gradients and the proxy backdoor task direction. Let $A(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top \mathbf{v}$ denote the inner-product similarity used in the main experiments. We define the gradient-consistent perturbation as

$$\boldsymbol{\delta}^{\mathrm{GA}}(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\delta} \in S} \Big\{ \mathbb{E}_{(\mathbf{x},y)\sim D_c}\big[L(\boldsymbol{\theta}; \mathbf{x}+\boldsymbol{\delta}, y_t)\big] + \beta\, A\Big(\mathbb{E}_{(\mathbf{x},y)\sim D_c}\big[\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}; \mathbf{x}+\boldsymbol{\delta}, y_t)\big], \boldsymbol{\theta}_b - \boldsymbol{\theta}\Big) \Big\}, \quad (7)$$

Writing $g_\theta = \mathbb{E}_{D_c}[\nabla_\theta L(\theta; x+\delta, y_t)]$, minimizing the $+\beta g_\theta^\top v_{\mathrm{proxy}}$ term favors a larger projection of the parameter descent direction $-g_\theta$ onto the proxy, without guaranteeing positive cosine alignment. The parameter $\beta > 0$ controls the strength of gradient alignment, and the alignment term uses only the model difference $\boldsymbol{\theta}_b - \boldsymbol{\theta}$, which is fully determined by the original backdoored model and the current model being optimized during defense, requiring no information from the attacker. Following our earlier discussion, we interpret $\boldsymbol{\theta}_b - \boldsymbol{\theta}$ as an estimate of the backdoor task direction in parameter space. The proxy is recomputed as $\boldsymbol{\theta}$ evolves during defense, so each outer update provides a new reference direction for the inner search. Appendix E.4 examines its relation to backdoor behavior, and Appendix E.5 reports differences between reactivation-induced gradients across the evaluated attacks.

Replacing $\boldsymbol{\delta}_B(\boldsymbol{\theta})$ in Eq. (5) by $\boldsymbol{\delta}^{\mathrm{GA}}(\boldsymbol{\theta})$, we obtain the gradient-aligned surrogate reactivation loss

$$\hat{R}^{\mathrm{GA}}_{\mathrm{react}}(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim D_c}\big[L(\boldsymbol{\theta}; \mathbf{x}+\boldsymbol{\delta}^{\mathrm{GA}}(\boldsymbol{\theta}), y)\big], \quad (8)$$

and the corresponding bilevel objective

$$\min_{\boldsymbol{\theta}} \quad R_{\mathrm{clean}}(\boldsymbol{\theta}) + \lambda \hat{R}^{\mathrm{GA}}_{\mathrm{react}}(\boldsymbol{\theta}) \quad \text{s.t.} \quad \boldsymbol{\delta}^{\mathrm{GA}}(\boldsymbol{\theta}) \text{ is provided by Eq. (7).} \quad (9)$$

Compared to the naive surrogate in Eq. (6), this formulation directly ties the lower perturbation search to the estimated backdoor task direction in parameter space. Empirically, a defense-aware black-box attacker that penalizes alignment with a surrogate-estimated proxy does not substantially increase reactivation ASR on the tested BadNets and LF models. Appendix E.3 reports this evaluation and a complementary white-box perturbation-budget study on the fixed final model.

3.3 Effects on Reactivation Risk and Attack Success

Theoretical Intuition. The proxy acts as a compass: the inner search favors inputs whose target-loss parameter descent directions point toward the backdoored model, and the outer update trains

on these inputs with correct labels. The compass guides where to probe without reconstructing the true trigger. At fixed model parameters, increasing β changes the selected perturbation, which in turn changes the outer gradient. Denote the baseline and aligned outer gradients by G_B and G_{GA}, computed with their selected perturbations held fixed. Under local regularity and interior-solution conditions, the sensitivity-gap assumption makes the leading correction's projection onto the true risk gradient positive. Appendix B.2 then establishes $\langle G_{\mathrm{true}}, G_{\mathrm{GA}} \rangle > \langle G_{\mathrm{true}}, G_B \rangle$ for sufficiently small $\beta > 0$ and fixed $\lambda > 0$, where $G_{\mathrm{true}} = \nabla_\theta R_{\mathrm{react}}$. Geometrically, relative to the baseline, the negative outer gradient has a larger projection onto the true risk's descent direction, providing better alignment in this projection sense.

Empirical Validation. To complement the above theoretical result, we directly compare the true reactivation risk and the surrogate risk along the optimization trajectory. Fig. 3 plots the true risk $R_{\mathrm{react}}(\boldsymbol{\theta}_t)$ evaluated on D_{bd} and the surrogate risk on D_c for BadNets, both for the base surrogate without gradient alignment and for our method with gradient alignment. The surrogate risk on D_c continues to decrease, whereas the true risk on D_{bd} quickly plateaus at a relatively high level, revealing a substantial gap between the surrogate objective and the true backdoor behavior. In contrast, with gradient alignment the true risk decreases more rapidly and continues to decline throughout training, and the gap between the true and surrogate risks is notably smaller. These trajectories show a lower reactivation risk and a smaller true–surrogate risk gap in this setting; they do not directly measure the gradient inner product studied in the theory.

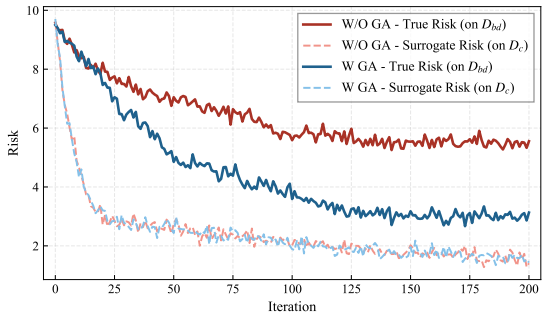

Figure 3: True and surrogate reactivation risks on BadNets.

4 Experiments

4.1 Experimental Setup

Attack Settings. In this study, we evaluate the efficacy of post-training defenses against a range of backdoor attacks within a unified experimental framework. Specifically, we include ten backdoor attack methods: BadNets (Gu et al., 2019), in both its all-to-one (BadNets-A2O) and all-to-all (BadNets-A2A) forms, Blended attack (Chen et al., 2017), Label-consistent attack (LC) (Turner et al., 2019), Low-frequency attack (LF) (Zeng et al., 2021), Sample-specific backdoor attack (SSBA) (Li et al., 2021c), BppAttack (Wang et al., 2022), Trojan attack (TrojanNN) (Liu et al., 2018b), Input-aware attack (Input-aware) (Nguyen and Tran, 2020), and Warping-based attack (WaNet) (Nguyen and Tran, 2021). Each attack is configured according to the default settings provided by BackdoorBench (Wu et al., 2025a). The experimental evaluation is conducted on three benchmark datasets: CIFAR-10 (Krizhevsky et al., 2009), Tiny ImageNet (Le and Yang, 2015), and GTSRB (Stallkamp et al., 2011), and implemented on three neural network architectures, namely PreAct-ResNet18 (He et al., 2016), VGG19-BN (Simonyan and Zisserman, 2014) and ViT-B-16 (Dosovitskiy et al., 2021) . Due to space constraints, the results for GTSRB, Tiny, VGG19-BN and ViT-B-16 are presented in the Sec. D.1 of the **Appendix**. For our experiments, we set the poisoning ratio uniformly to 10% across all attack types, and the target label of BadNets-A2A is reassigned to $y_t = (y+1) \mod K$ for each class y, where K represents the total number of classes. However, the LC attack is only evaluated on CIFAR-10, as the other datasets have too many categories to support a 10% poisoning rate on the target class.

Defense Settings. We systematically evaluate our method under a unified post-training defense protocol. We consider a set of representative defense approaches that span diverse mitigation paradigms, including Neural Cleanse (NC) (Wang et al., 2019), NAD (Li et al., 2021b), i-BAU (Zeng et al., 2022), FT-SAM (Zhu et al., 2023), SAU (Wei et al., 2023), and FST (Min et al., 2024a). Each defense is reimplemented under consistent optimization and fine-tuning settings to ensure fair comparison. Each method can access 5% benign training data. For GARF, the target label in the main

Table 2: Performance results in percentage of post-training defense methods on CIFAR-10 using PreAct-ResNet18 with a poisoning ratio of 10.0%. The metrics include Clean Accuracy, Attack Success Rate, white-box Reactivation ASR, and black-box Reactivation ASR. **The best results are highlighted in bold.**

Defense →	No Defense				SAU				FST				NAD			
Attack ↓	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B
Blended	93.69	99.76	–	–	91.75	5.20	98.39	66.42	92.61	**0.20**	95.63	85.74	92.40	79.92	99.82	94.16
BadNets	91.82	93.79	–	–	89.81	**1.30**	92.74	30.34	91.06	1.46	97.90	51.79	86.92	3.77	94.81	54.49
SSBA	92.88	97.07	–	–	88.96	2.16	89.46	43.83	92.06	0.54	94.27	53.94	91.66	75.48	99.72	89.04
LF	93.01	99.06	–	–	72.68	2.49	90.76	**0.08**	91.89	5.42	98.22	1.20	91.08	96.58	99.41	97.00
LC	84.43	99.88	–	–	87.86	**2.96**	68.04	34.88	86.28	3.23	72.99	14.44	89.78	22.61	83.04	12.08
TrojanNN	93.47	99.99	–	–	91.64	1.39	88.01	50.26	92.02	8.93	97.30	74.64	92.16	6.94	96.32	71.64
BadNets-A2A	91.89	74.42	–	–	90.95	1.30	96.25	43.65	90.94	**0.65**	97.15	44.04	88.77	2.40	89.44	36.58
WaNet	92.80	98.90	–	–	91.94	0.82	95.38	59.70	92.65	**0.26**	97.57	77.94	92.81	1.27	96.28	67.34
Input-aware	94.03	98.35	–	–	91.93	10.67	85.79	20.70	93.79	**0.00**	90.46	50.09	**94.00**	0.92	93.76	72.13
BppAttack	90.81	99.81	–	–	91.39	11.47	61.42	29.58	91.53	**2.49**	68.68	37.93	92.99	12.20	79.39	31.49
Average	91.88	96.10	–	–	88.89	3.98	86.62	37.94	91.48	**2.32**	91.02	49.18	91.26	30.21	93.20	62.59

Defense →	I-BAU				NC				FT-SAM				GARF			
Attack ↓	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B
Blended	89.43	26.82	99.61	88.57	**93.69**	99.76	99.93	99.13	92.87	12.16	99.56	85.74	91.92	4.00	**38.11**	**15.89**
BadNets	87.43	4.48	97.36	51.62	90.33	2.01	96.94	26.22	**92.21**	1.63	94.86	57.53	89.41	1.48	**39.67**	**0.01**
SSBA	86.53	2.89	91.46	64.09	**92.88**	97.07	99.90	94.51	92.18	4.06	92.90	75.27	90.71	**0.31**	**3.23**	**0.00**
LF	88.92	11.99	99.04	76.82	**93.01**	99.06	99.41	79.56	92.29	6.43	97.37	88.37	90.40	**0.43**	**32.42**	1.29
LC	85.66	4.61	80.01	9.59	90.65	10.62	99.96	99.99	**91.00**	3.28	91.24	0.83	88.92	4.09	**36.99**	**0.20**
TrojanNN	89.29	0.54	89.89	37.03	92.03	2.77	95.23	41.02	**92.75**	4.12	96.04	71.48	91.14	**0.32**	**28.48**	**4.87**
BadNets-A2A	89.39	1.29	94.18	50.22	89.79	1.11	97.60	46.05	**91.87**	1.03	95.71	39.23	90.23	1.46	**47.64**	**1.19**
WaNet	90.70	0.88	94.96	82.44	92.80	98.90	100.00	99.83	**92.87**	0.96	95.25	79.76	91.13	1.86	**44.11**	**0.00**
Input-aware	89.91	0.02	21.39	1.57	94.00	0.70	92.11	55.00	93.76	1.07	96.16	80.11	91.33	9.35	**1.45**	**1.32**
BppAttack	90.72	11.01	89.88	12.28	93.04	12.39	99.99	99.25	**93.21**	13.53	82.21	45.75	90.59	10.32	**21.25**	**10.46**
Average	88.80	6.45	85.78	47.42	92.22	42.44	98.11	74.06	**92.50**	4.83	94.13	62.41	90.58	3.36	**29.33**	**3.52**

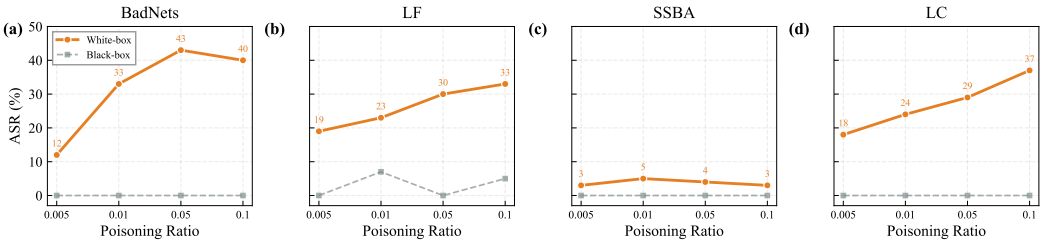

Figure 4: Defense performance for reactivation attack of our method GARF under different poisoning ratios on CIFAR-10 and PreAct-ResNet18.

experiments is supplied by the detector ensemble in Appendix E.2. We follow the recommended configurations for SOTA defenses as in BackdoorBench (Wu et al., 2025a). Further implementation details, hyperparameter configurations, and additional comparisons with PAM (Min et al., 2024b) and BTI (Xu et al., 2024) are provided in the Sec. D.2 and Sec. D.3 of the **Appendix**.

Evaluation Metrics. We employ four metrics. **Clean Accuracy (ACC)** measures utility on benign samples. **Attack Success Rate (ASR)** measures the fraction of triggered inputs classified to the target label. To assess residual vulnerability after defense, we further adopt **white-box reactivation ASR** (ASR_W) and **black-box reactivation ASR** (ASR_B) following (Zhu et al., 2024a): the former assumes full model access, where the attacker optimizes a perturbation via gradient-based search to maximize target-class prediction on defended inputs; the latter assumes query-only access and optimizes the same objective without knowledge of model internals. Lower ASR, ASR_W, and ASR_B indicate stronger robustness; higher ACC indicates better task retention.

4.2 Main Results

Effectiveness of GARF. To validate GARF, we compare it with I-BAU, NC, FT-SAM, SAU, FST, and NAD on CIFAR-10 using PreAct-ResNet18, as shown in Tab. 2. ❶ GARF achieves mean clean accuracy of 90.58% and standard ASR of 3.36%. ❷ Its mean white-box and black-box reactivation ASRs are 29.33% and 3.52%, respectively, the lowest averages among the compared defenses. ❸ The baseline mean white-box reactivation ASRs range from 85.78% to 98.11%, so GARF reduces this

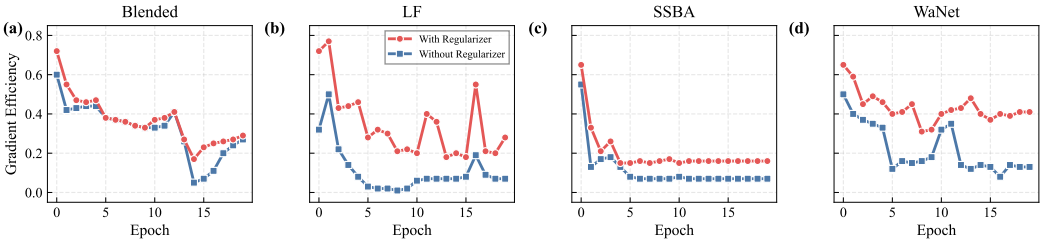

Figure 5: Evolution of gradient efficiency during GARF training under four backdoor attacks. Gradient efficiency is a linearly rescaled cosine similarity between the lower-level gradient and the proxy direction, using the empirical means for random clean and pure backdoor gradients as reference endpoints.

Table 3: Hyperparameter analysis of GARF on CIFAR-10 using PreAct-ResNet18 with poisoning ratio 10.0%. The first subtable reports the effect of clean data size, the second reports the effect of the outer regularization weight λ, and the third reports the effect of the inner penalty strength β.

Defense →	GARF (OOD)				GARF (1%)				GARF (5%)				GARF (10%)			
Attack ↓	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B
BadNets	86.93	2.11	42.31	0.83	87.76	1.94	38.52	0.27	89.41	1.48	39.67	0.01	89.95	1.57	41.72	0.10
SSBA	88.54	0.63	6.41	0.58	89.37	0.24	2.17	0.11	90.71	0.31	3.23	0.00	90.96	0.42	4.87	0.19
LF	87.82	0.71	34.15	7.06	88.91	0.56	30.81	4.43	90.40	0.43	32.42	1.29	90.11	0.68	33.57	6.02
WaNet	88.97	2.43	46.82	1.92	90.18	1.74	43.07	0.73	91.13	1.86	44.11	0.00	90.79	2.05	45.53	0.41
Defense →	GARF ($\lambda = 0.0$)				GARF ($\lambda = 0.5$)				GARF ($\lambda = 1.0$)				GARF ($\lambda = 1.5$)			
Attack ↓	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B
BadNets	91.12	5.23	98.91	76.54	89.85	1.73	40.52	0.19	89.41	1.48	39.67	0.01	89.02	1.34	38.95	0.04
SSBA	92.08	95.41	99.53	94.32	90.42	0.46	3.87	0.06	90.71	0.31	3.23	0.00	90.53	0.37	3.65	0.02
LF	91.47	97.82	99.31	93.17	90.69	0.58	31.13	4.87	90.40	0.43	32.42	1.29	90.21	0.51	33.04	5.61
WaNet	92.03	2.61	98.07	90.13	90.81	1.64	43.82	0.12	91.13	1.86	44.11	0.00	90.94	1.92	45.73	0.05
Defense →	GARF ($\beta = 0.0$)				GARF ($\beta = 5.0$)				GARF ($\beta = 10.0$)				GARF ($\beta = 50.0$)			
Attack ↓	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B
BadNets	90.37	5.32	83.64	15.54	89.76	2.11	43.07	2.81	89.41	1.48	39.67	0.01	88.89	1.31	40.12	0.15
SSBA	89.93	4.89	78.06	26.23	91.05	0.57	5.38	2.92	90.71	0.31	3.23	0.00	90.02	0.44	3.51	0.12
LF	89.71	5.73	80.56	0.04	90.02	0.81	35.60	8.96	90.40	0.43	32.42	1.29	89.88	0.57	31.98	5.74
WaNet	92.04	3.96	87.48	38.90	90.87	2.21	46.74	3.94	91.13	1.86	44.11	0.00	90.41	1.92	43.58	0.08

metric by 56.45–68.78 percentage points. These results demonstrate stronger average reactivation suppression under the evaluated attack settings while preserving competitive clean accuracy.

Performance under Different Poisoning Ratios. We evaluate GARF across poisoning ratios from 0.5% to 10% on CIFAR-10, as shown in Fig. 4. The reported black-box reactivation ASRs remain low across the evaluated settings. White-box reactivation varies with both the attack and the poisoning ratio. In particular, BadNets reaches approximately 43% at a 5% poisoning ratio, compared with approximately 40% at 10%, while SSBA also exhibits non-monotonic variation. Thus, the sweep shows the behavior of GARF over the tested poisoning ratios without establishing a monotonic relationship between poisoning ratio and reactivation ASR. The 10% setting is used for the main comparison.

Hyperparameter Sensitivity of GARF. We study GARF on CIFAR-10 with PreAct-ResNet18 at a 10% poisoning ratio, varying the clean data, the outer weight λ, and the inner alignment weight β. Table 3 shows broadly similar white-box reactivation ASRs across the tested clean-data settings and nonzero weights, with attack-dependent variation in clean accuracy and black-box reactivation. Setting $\lambda = 0$ removes the outer reactivation loss and yields high white-box reactivation ASR for all four attacks. Setting $\beta = 0$ removes gradient alignment and also increases white-box reactivation ASR relative to the tested positive-β settings. For example, the BadNets white-box ASR decreases from 83.64% at $\beta = 0$ to 39.67% at $\beta = 10$. The improvement is not uniform across every metric: for LF, black-box ASR is 0.04% at $\beta = 0$ and 1.29% at $\beta = 10$. Overall, the ablation supports the contribution of gradient alignment to white-box reactivation suppression in the evaluated settings. Additional alignment and random-perturbation ablations are reported in Appendix E.4, Tab. 18.

Table 4: Plug-in cold start for GARF. X → GARF (5) denotes initializing GARF from defense X and running 5 rounds; GARF (20) is our default setting. Entries report white-box / black-box reactivation ASR (%).

Attack → Defense ↓	BadNets ASR_W / ASR_B	SSBA ASR_W / ASR_B	LF ASR_W / ASR_B	TrojanNN ASR_W / ASR_B
SAU → GARF (5)	45.83 / 4.02	10.21 / 4.00	56.98 / 8.91	34.57 / 5.01
I-BAU → GARF (5)	46.52 / 2.02	15.28 / 5.05	37.56 / 6.00	36.41 / 6.00
FT-SAM → GARF (5)	41.43 / 5.07	8.37 / 3.32	55.93 / 7.07	32.62 / 5.84
GARF (5)	62.12 / 10.00	36.31 / 7.12	63.33 / 10.01	18.09 / 5.14
GARF (20)	39.67 / 0.01	3.23 / 0.00	32.42 / 1.29	28.48 / 4.87

4.3 Analysis of GARF

Plug-in Cold Start for GARF. We evaluate GARF as a plug-in refinement stage where the original backdoored model used in the gradient-alignment term is kept fixed and only the defended initialization θ_{init} is varied. Specifically, we take models produced by SAU, I-BAU, and FT-SAM as θ_{init} and then run GARF for five rounds, as summarized in Tab. 4. Compared with running GARF for five rounds from the original backdoored checkpoint θ_b, all plug-in variants achieve lower average white-box and black-box reactivation ASR. These results show that the evaluated defended checkpoints can provide useful initializations for a five-round GARF refinement stage. The five rounds refer to the additional GARF stage, rather than the total cost of producing the initial defended checkpoint.

Effect of Gradient Alignment Regularizer. Fig. 5 reports the reference-rescaled gradient-efficiency statistic during GARF training under four attacks. It uses cosine similarity between the lower-level gradient and the proxy, rescaled with empirical means for random clean and pure backdoor gradients as reference endpoints. The gradient-aligned variant has higher reported scores than the base surrogate after the first few epochs, especially for LF and WaNet. These trajectories characterize the reported statistic; they do not measure alignment with the full reactivation-risk gradient or establish a larger raw gradient inner product. Reactivation-risk curves in Fig. 3 and the ablations separately evaluate the defense's empirical effect.

Additional Analyses. Beyond the main comparisons, we conduct several further analyses in the **Appendix**. ❶ We evaluate a defense-aware black-box reactivation attack that penalizes alignment with the estimated proxy, observing low reactivation ASR on BadNets and LF; ❷ We compare the computational cost of GARF against existing defenses; ❸ We describe the target label detection strategy used in GARF and validate its reliability.

5 Conclusion

In this paper, we revisit backdoor defense through the lens of post-defense reactivation, where a seemingly sanitized model can still be pushed back to its backdoor behavior by adaptive perturbations. We formalize reactivation risk and show that it is hard to control when the defender has a backdoored model and a small clean dataset, without poisoned samples or trigger information. We propose GARF, a bilevel post-training defense that learns gradient-aligned perturbations on clean data to repeatedly expose and suppress reactivation-sensitive directions without trigger information. Our analysis establishes a conditional local improvement in the surrogate gradient's unnormalized inner product with the true reactivation gradient. Experiments across multiple datasets, architectures, and attacks show reduced reactivation ASR while preserving competitive clean accuracy under the evaluated settings. We view this work as a first step toward explicitly modeling adaptive reactivation in post-training defenses and highlight post-defense reactivation as a concrete risk that merits further study.

Future Extension. This work focuses on backdoor reactivation in supervised image classification under a post-training threat model. Extending the reactivation phenomenon and the proposed defense to other learning paradigms, such as self-supervised learning, federated learning, or generative models, remains an open empirical question beyond the scope of the present study.

REFERENCES

Venkat Adithya Amula, Sunayana Samavedam, Saurabh Saini, Avani Gupta, and PJ Narayanan. Prototype guided backdoor defense via activation space manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Weixin Chen, Baoyuan Wu, and Haoqian Wang. Effective backdoor defense by exploiting sensitivity of poisoned samples. In *Advances in Neural Information Processing Systems*, 2022.

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

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

Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. In *The Eleventh International Conference on Learning Representations*, 2023.

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

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009.

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

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

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *International Conference on Learning Representations*, 2021b.

Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *International Conference on Computer Vision*, 2021c.

Weilin Lin, Li Liu, Jianze Li, and Hui Xiong. Fusing pruned and backdoored models: Optimal transport-based data-free backdoor mitigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 2025.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Research in Attacks, Intrusions, and Defenses*, 2018a.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *Network and Distributed System Security Symposium*, 2018b.

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

Rui Min, Zeyu Qin, Nevin L Zhang, Li Shen, and Minhao Cheng. Uncovering, explaining, and mitigating the superficial safety of backdoor defense. *Advances in Neural Information Processing Systems*, 37, 2024b.

Hossein Mirzaei, Zeinab Taghavi, Sepehr Rezaee, Masoud Hadi, Moein Madadi, and Mackenzie W Mathis. Distil: Data-free inversion of suspicious trojan inputs via latent diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

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

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

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

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

Zhenxing Niu, Yuyao Sun, Qiguang Miao, Rong Jin, and Gang Hua. Towards unified robustness against both backdoor and adversarial attacks. *IEEE transactions on pattern analysis and machine intelligence*, 2024.

Dimitrios P Panagoulias, Maria Virvou, and George A Tsihrintzis. Evaluating llm–generated multimodal diagnosis from medical images and symptom analysis. *arXiv preprint arXiv:2402.01730*, 2024.

Xiangyu Qi, Tinghao Xie, Yiming Li, Saeed Mahloujifar, and Prateek Mittal. Revisiting the assumption of latent separability for backdoor defenses. In *International Conference on Learning Representations*, 2023a.

Xiangyu Qi, Tinghao Xie, Jiachen T Wang, Tong Wu, Saeed Mahloujifar, and Prateek Mittal. Towards a proactive ml approach for detecting backdoor poison samples. In *USENIX Security Symposium*, 2023b.

Zhiyao Shu, Xiangguo Sun, and Hong Cheng. When llm meets hypergraph: A sociological analysis on personality via online social networks. In *ACM International Conference on Information and Knowledge Management*, 2024.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

Di Tang, XiaoFeng Wang, Haixu Tang, and Kehuan Zhang. Demon in the variant: Statistical analysis of dnns for robust backdoor contamination detection. In *USENIX Security Symposium*, 2021.

Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv e-prints*, 2019.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *Symposium on Security and Privacy*, 2019.

Zhenting Wang, Juan Zhai, and Shiqing Ma. Bppattack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. In *Conference on Computer Vision and Pattern Recognition*, 2022.

Zhenting Wang, Kai Mei, Juan Zhai, and Shiqing Ma. Unicorn: A unified backdoor trigger inversion framework. In *International Conference on Learning Representations*, 2023.

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

Shaokui Wei, Jiayin Liu, and Hongyuan Zha. Backdoor mitigation by distance-driven detoxification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, Mingli Zhu, Ruotong Wang, Li Liu, and Chao Shen. Backdoorbench: A comprehensive benchmark and analysis of backdoor learning. *International Journal of Computer Vision*, 133(8):5700–5787, 2025a.

Baoyuan Wu, Mingli Zhu, Meixi Zheng, Zihao Zhu, Shaokui Wei, Mingda Zhang, Hongrui Chen, Danni Yuan, Li Liu, and Qingshan Liu. Defenses in adversarial machine learning: A systematic survey from the lifecycle perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b.

Baoyuan Wu, Zihao Zhu, Li Liu, Qingshan Liu, Zhaofeng He, and Siwei Lyu. Attacks in adversarial machine learning: A systematic survey from the lifecycle perspective. *International Journal of Computer Vision*, 134 (5):197, 2026.

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

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

Hakan Yekta Yatbaz, Mehrdad Dianati, and Roger Woodman. Introspection of dnn-based perception functions in automated driving systems: State-of-the-art and open research challenges. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

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

Yi Zeng, Si Chen, Won Park, Zhuoqing Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *International Conference on Learning Representations*, 2022.

Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Pre-activation distributions expose backdoor neurons. In *Advances in Neural Information Processing Systems*, 2022.

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

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

Zihao Zhu, Mingda Zhang, Shaokui Wei, Bingzhe Wu, and Baoyuan Wu. Vdc: Versatile data cleanser based on visual-linguistic inconsistency by multimodal large language models. In *The Twelfth International Conference on Learning Representations*, 2024b.

GARF: Gradient-Aligned Reactivation Forgetting for Backdoor Defense

Appendix

A Broader Impact **14**

B Details of GARF **14**

B.1 GARF Algorithm 14

B.2 Theoretical Analysis of Gradient-Aligned Bilevel Defense 15

B.3 Relation to Adversarial-Training-based Defenses 18

C Details of Experiment Settings **19**

C.1 Datasets and Model Architectures 19

C.2 Detailed Training Settings 19

C.3 Detailed Attack Settings 19

C.4 Detailed Defense Settings 20

D Additional Experiments **21**

D.1 Main Experiments on Different Datasets and Models. 21

D.2 Analysis with PAM. 23

D.3 Analysis with BTI 23

E Additional Analysis **24**

E.1 Computational Cost 24

E.2 Ablation Studies on Target Labels 24

E.3 Robustness to Adaptive Attacks. 25

E.4 Empirical Analysis of the Proxy Backdoor Direction. 27

E.5 Heterogeneity of backdoor perturbations. 28

AI USE STATEMENT

We used generative AI tools to assist with manuscript editing and LaTeX preparation, the interpretation and presentation of existing experimental results, and clarification of the scope of theoretical claims. The authors take full responsibility for the accuracy and integrity of the final manuscript, including all AI-assisted text and interpretations.

REPRODUCIBILITY STATEMENT

The GARF optimization procedure is specified in Algorithm 1 and Appendix B. Appendix C describes the datasets, model architectures, training protocols, attack configurations, and defense hyperparameters. The assumptions and proofs underlying the theoretical analysis are provided in Appendix B.2. Additional settings and results for target-label handling, adaptive reactivation, and proxy analysis are reported in Appendices E.2, E.3, and E.4, respectively.

ETHICS STATEMENT

This work aims to improve the security of machine learning models by evaluating and mitigating backdoor reactivation after defense. The experiments use established image-classification benchmarks to study this threat. Reactivation analysis has dual-use implications: techniques used to evaluate defenses could also help an adversary recover malicious behavior. We present these analyses to support defensive evaluation and recommend that they be applied only to models and systems for which testing is authorized.

A BROADER IMPACT

Deep neural networks are increasingly deployed in safety-critical applications such as autonomous driving, medical diagnosis, and financial fraud detection. The vulnerability of these systems to backdoor attacks poses a serious societal risk: an adversary who successfully implants a hidden trigger–label association can cause models to behave maliciously on demand, potentially leading to catastrophic real-world consequences. While existing post-training defenses have made progress in suppressing visible backdoor behavior, our work reveals that seemingly sanitized models remain vulnerable to adaptive reactivation attacks that can restore malicious behavior with high success rates. GARF addresses this gap by explicitly modeling and suppressing reactivation-sensitive directions, improving empirical robustness under the evaluated reactivation attacks without requiring access to poisoned data or trigger information. We believe this work contributes positively to the trustworthiness of machine learning systems and encourages the community to evaluate defenses under adaptive, post-defense threat models rather than assuming a static backdoor.

B DETAILS OF GARF

B.1 GARF ALGORITHM

GARF is a post-training bilevel defense that starts from a pre-defense checkpoint f_{θ_b} and uses only a small clean subset of the training data. During training, we regard the current model f_θ as the defended model, and use the parameter difference $v_{\text{proxy}} = \boldsymbol{\theta}_b - \boldsymbol{\theta}$ as a proxy for the backdoor task direction. At each iteration, GARF constructs a surrogate reactivation attacker guided by the relationship between the induced gradients and the current proxy direction v_{proxy}. The outer level then updates θ using both the clean loss and the surrogate reactivation loss, driven by these gradient-aligned perturbations.

During each inner solve, the model parameters and the current proxy are held fixed. The dependence of $g_\theta(\delta)$ on δ is retained when differentiating the alignment term, so that it contributes to the perturbation update. For the outer model update, the optimized perturbation is treated as constant, without backpropagating through the inner optimization trajectory, consistent with the first-order outer approximation in Appendix B.2.

Algorithm 1 GARF: Gradient-aligned Reactivation Forgetting

Input: pre-defense parameters $\boldsymbol{\theta}_b$, clean subset $\mathcal{D}_c$, target label y_t, hyperparameters $\lambda, \beta, \varepsilon, \alpha, T, lr_{\text{GARF}}, E$.
Initialize: set model parameters $\theta \leftarrow \boldsymbol{\theta}_b$.
for $e = 1$ to E **do**
  for each mini-batch $\{(x_i, y_i)\}_{i=1}^{B} \subset \mathcal{D}_c$ **do**
    Compute proxy direction $v_{\text{proxy}} \leftarrow \boldsymbol{\theta}_b - \theta$.
    Initialize inner perturbation $\delta^{(0)} \leftarrow 0$.
    for $t = 0$ to $T - 1$ **do**
      Compute reactivation surrogate loss

$$\mathcal{L}_{\text{react}}(\theta, \delta^{(t)}) = \frac{1}{B} \sum_{i=1}^{B} L(\theta; x_i + \delta^{(t)}, y_t).$$

      Compute average parameter gradient

$$g_\theta(\delta^{(t)}) = \frac{1}{B} \sum_{i=1}^{B} \nabla_\theta L(\theta; x_i + \delta^{(t)}, y_t).$$

      Define inner objective

$$\mathcal{L}_{\text{inner}}(\theta, \delta^{(t)}) = \mathcal{L}_{\text{react}}(\theta, \delta^{(t)}) + \beta\, g_\theta(\delta^{(t)})^\top v_{\text{proxy}}.$$

      Take one projected SGD step on δ:

$$\delta^{(t+1)} \leftarrow \Pi_{\|\delta\|_\infty \le \varepsilon}\big(\delta^{(t)} - \alpha \nabla_\delta \mathcal{L}_{\text{inner}}(\theta, \delta^{(t)})\big).$$

    end for
    Set $\delta_{\text{GA}} \leftarrow \text{stopgrad}(\delta^{(T)})$.
    Compute clean and reactivation losses

$$\mathcal{L}_{\text{clean}}(\theta) = \frac{1}{B} \sum_{i=1}^{B} L(\theta; x_i, y_i),$$

$$\mathcal{L}_{\text{react-outer}}(\theta) = \frac{1}{B} \sum_{i=1}^{B} L(\theta; x_i + \delta_{\text{GA}}, y_i).$$

    Outer loss:

$$\mathcal{L}_{\text{GARF}}(\theta) = \mathcal{L}_{\text{clean}}(\theta) + \lambda\, \mathcal{L}_{\text{react-outer}}(\theta).$$

    Update model parameters:

$$\theta \leftarrow \theta - lr_{\text{GARF}} \nabla_\theta \mathcal{L}_{\text{GARF}}(\theta).$$

  end for
end for
Output: defended model f_θ with reduced reactivation.

Algorithm 1 summarizes the procedure. In practice, we approximate the expectations over the clean distribution $\mathcal{D}_c$ by mini-batches from a small clean subset, and use a few steps of projected SGD to solve the inner problem in Eq. (7) of the main paper.

B.2 Theoretical Analysis of Gradient-Aligned Bilevel Defense

We analyze the local effect of gradient alignment on the first-order outer gradient. Under the regularity, interior-solution, and sensitivity-gap assumptions stated below, the correction increases the unnormalized inner product with the true reactivation gradient for sufficiently small positive alignment weights.

B.2.1 Notation and Bilevel Formulation

Recall that the defender observes a suspicious pre-defense model $f_{\boldsymbol{\theta}_b}$ and a current defended model f_θ. The reactivation risk is

$$R_{\text{react}}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_c}\big[L(\theta; \tilde{x} + \delta^\star(\theta), y)\big], \tag{10}$$

where $\tilde{x} = g(x, \tau)$ is the conceptual poisoned counterpart of x and $\delta^\star(\theta)$ is the perturbation selected by the true reactivation attacker. The clean risk is

$$R_{\text{clean}}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_c}[L(\theta; x, y)]. \tag{11}$$

The defender cannot directly optimize the poisoned-input risk and instead uses clean-data surrogate perturbations in the feasible set S. The baseline inner problem is

$$\delta_B(\theta) = \arg\min_{\delta\in S} \mathbb{E}_{(x,y)\sim\mathcal{D}_c}[L(\theta; x + \delta, y_t)], \tag{12}$$

where y_t is the attack target. The analysis considers the target-matched case, in which the surrogate uses this label. With outer weight $\lambda = 1$, the baseline outer objective is

$$\mathcal{L}_B(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_c}[L(\theta; x, y) + L(\theta; x + \delta_B(\theta), y)]. \tag{13}$$

Let G_B denote its first-order outer gradient, defined below by holding the selected inner solution fixed when differentiating with respect to θ.

GARF uses $v_{\text{proxy}} = \boldsymbol{\theta}_b - \theta$ to guide the inner search. For the dot-product alignment used in the main experiments, the surrogate perturbation is

$$\delta_{\text{GA}}(\theta, \beta) = \arg\min_{\delta\in S}\Big\{\mathbb{E}_{(x,y)\sim\mathcal{D}_c}[L(\theta; x+\delta, y_t)] + \beta\, \mathbb{E}_{(x,y)\sim\mathcal{D}_c}[\nabla_\theta L(\theta; x+\delta, y_t)]^\top v_{\text{proxy}}\Big\}. \tag{14}$$

For $\beta > 0$, the regularizer penalizes larger inner products between the target-loss gradient and v_{proxy}. A negative inner product corresponds to a positive projection of the parameter descent direction $-\nabla_\theta L$ onto the proxy. The two terms are optimized jointly: the objective encourages this behavior without guaranteeing separate minimization of the two terms or exact recovery of the true attack. The corresponding outer objective is

$$\mathcal{L}_{\text{GA}}(\theta, \beta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_c}[L(\theta; x, y) + L(\theta; x + \delta_{\text{GA}}(\theta, \beta), y)]. \tag{15}$$

Let $G_{\text{GA}}(\beta)$ denote its first-order outer gradient. At the parameter value considered, assume that R_{react} is differentiable and retain its full derivative $G_{\text{true}} = \nabla_\theta R_{\text{react}}(\theta)$ as the theoretical reference. Define $C(\beta) = G_{\text{GA}}(\beta) - G_B$.

Remark (First-order approximation)**.** Fix θ and v_{proxy} throughout the local analysis in β. Write $\delta^*(\beta) = \delta_{\text{GA}}(\theta, \beta)$ for the selected surrogate solution, with $\delta^*(0) = \delta_B$. This is distinct from the true attacker's solution $\delta^\star(\theta)$. The first-order outer gradient is

$$G_{\text{GA}}(\beta) \triangleq \nabla_\theta \mathbb{E}[L(\theta; x, y) + L(\theta; x + \delta^*(\beta), y)]|_{\delta^* \text{ fixed}} \tag{16}$$

$$= \mathbb{E}[\nabla_\theta L(\theta; x, y)] + \mathbb{E}[\nabla_\theta L(\theta; x + \delta^*(\beta), y)]. \tag{17}$$

Here and below, expectations are over $(x, y) \sim \mathcal{D}_c$. Define G_B analogously at δ_B. These vectors omit the dependence of the surrogate inner solution on θ; they are not the full derivatives of the composite bilevel objectives. This approximation does not apply to G_{true}, which need not be available to the defender. When differentiating with respect to β, the selected surrogate solution is allowed to vary.

B.2.2 Assumptions

For fixed θ, define the scalar functions f, a and the parameter-gradient vector q by

$$f(\delta) = \mathbb{E}[L(\theta; x + \delta, y_t)], \qquad a(\delta) = \mathbb{E}[\nabla_\theta L(\theta; x + \delta, y_t)]^\top v_{\text{proxy}},$$

$$\Phi(\delta, \beta) = f(\delta) + \beta a(\delta), \qquad q(\delta) = \mathbb{E}[\nabla_\theta L(\theta; x + \delta, y)].$$

Assumption 1 (Backdoor sensitivity gap). Define the following matrices at the baseline solution δ_B:

- $M_y = \mathbb{E}[\nabla^2_{\theta\delta} L(\theta; x+\delta, y)]\big|_{\delta=\delta_B} \in \mathbb{R}^{|\theta|\times|\delta|}$, the mixed Hessian of the outer loss;
- $M_{y_t} = \mathbb{E}[\nabla^2_{\theta\delta} L(\theta; x+\delta, y_t)]\big|_{\delta=\delta_B} \in \mathbb{R}^{|\theta|\times|\delta|}$, the mixed Hessian of the inner loss;
- $H_{\delta\delta} = \nabla^2_\delta f(\delta_B) \in \mathbb{R}^{|\delta|\times|\delta|}$, assumed positive definite in Assumption 2;
- $w = -H^{-1}_{\delta\delta} M^\top_{y_t} v_{\text{proxy}} \in \mathbb{R}^{|\delta|}$;
- $P = M_y H^{-1}_{\delta\delta} M^\top_{y_t} \in \mathbb{R}^{|\theta|\times|\theta|}$.

We assume

$$\langle G_{\text{true}}, P v_{\text{proxy}} \rangle = \langle M_y^\top G_{\text{true}}, -w \rangle < 0. \tag{18}$$

Physical Interpretation. The vector $M^\top_{y_t} v_{\text{proxy}}$ is the directional derivative of the inner input gradient along the proxy in parameter space. As shown below, w is the first-order response of the selected inner solution to β, accounting for the local curvature through $H^{-1}_{\delta\delta}$. Its induced outer-gradient correction is $M_y w = -P v_{\text{proxy}}$. Condition equation 18 explicitly assumes $\langle G_{\text{true}}, M_y w \rangle > 0$. It is not implied solely by the presence of a residual backdoor or a change in predicted label, and is not asserted to hold at every defense iteration.

Assumption 2 (Local regularity and nondegeneracy). The stationarity map $(\delta, \beta) \mapsto \nabla_\delta \Phi(\delta, \beta)$ is twice continuously differentiable near $(\delta_B, 0)$, and $q(\delta)$ is twice continuously differentiable near δ_B. The derivatives used below may be interchanged with expectation. Moreover, $H_{\delta\delta} \succ 0$. In particular, $\nabla_\delta a(\delta_B) = M^\top_{y_t} v_{\text{proxy}}$ and $D_\delta q(\delta_B) = M_y$.

Assumption 3 (Continuous interior solution branch). For β in a neighborhood of zero, the selected minimizers of the surrogate inner problem satisfy $\delta^*(0) = \delta_B$ and $\delta^*(\beta) \to \delta_B$ as $\beta \to 0$. They lie in the strict interior of S; for the ℓ_∞ ball, $\|\delta^*(\beta)\|_\infty < \varepsilon$. Their stationarity condition is

$$\nabla_\delta \Phi(\delta^*(\beta), \beta) = \nabla_\delta f(\delta^*(\beta)) + \beta \nabla_\delta a(\delta^*(\beta)) = 0. \tag{19}$$

This assumption excludes active constraints and switching away from the local solution branch through δ_B.

B.2.3 First-order gradient correction

Proposition 1 (First-order gradient correction)**.** *Under Assumptions 2 and 3, the correction $C(\beta) = G_{\text{GA}}(\beta) - G_B$ satisfies*

$$\left.\frac{\mathrm{d}C(\beta)}{\mathrm{d}\beta}\right|_{\beta=0} = -P v_{\text{proxy}}, \tag{20}$$

where $P = M_y H^{-1}_{\delta\delta} M^\top_{y_t}$. Consequently,

$$C(\beta) = -\beta P v_{\text{proxy}} + O(\beta^2). \tag{21}$$

Proof. **Step 1: First-order optimality condition (FOC).** The inner objective is

$$\Phi(\delta, \beta) = f(\delta) + \beta a(\delta). \tag{22}$$

By Assumption 3, the selected solution satisfies

$$\nabla_\delta f(\delta^*(\beta)) + \beta \nabla_\delta a(\delta^*(\beta)) = 0. \tag{FOC}$$

At $\beta = 0$, $\nabla_\delta a(\delta_B) = M^\top_{y_t} v_{\text{proxy}}$. At nonzero β, the derivatives in equation FOC are evaluated at the current solution.

Step 2: Implicit function theorem (IFT). At $(\delta_B, 0)$, the Jacobian of the stationarity map with respect to δ is $H_{\delta\delta}$, invertible by Assumption 2. The implicit function theorem gives a locally unique C^2 stationary branch. By continuity and stationarity in Assumption 3, the selected minimizers coincide with this branch for sufficiently small β. Differentiating equation FOC at $\beta = 0$ gives

$$H_{\delta\delta} \left.\frac{\mathrm{d}\delta^*(\beta)}{\mathrm{d}\beta}\right|_{\beta=0} + M^\top_{y_t} v_{\text{proxy}} = 0. \tag{23}$$

Thus

$$\left.\frac{\mathrm{d}\delta^*(\beta)}{\mathrm{d}\beta}\right|_{\beta=0} = -H_{\delta\delta}^{-1} M_{y_t}^\top v_{\text{proxy}} = w. \tag{IFT}$$

Step 3: Chain rule on the first-order outer gradient. The clean-gradient term is independent of β, so $C(\beta) = q(\delta^*(\beta)) - q(\delta_B)$. Hence

$$C'(0) = D_\delta q(\delta_B)\, \delta^{*\prime}(0) = M_y\, \delta^{*\prime}(0). \tag{24}$$

Step 4: First-order coefficient and remainder. Substituting equation IFT,

$$C'(0) = M_y w = -M_y H_{\delta\delta}^{-1} M_{y_t}^\top v_{\text{proxy}} = -P v_{\text{proxy}}. \tag{25}$$

The solution branch and q are C^2, so C is C^2 near zero. Since $C(0) = 0$, Taylor's theorem gives $C(\beta) = -\beta P v_{\text{proxy}} + O(\beta^2)$. No cancellation of the second-order term is required. □

B.2.4 Gradient Alignment Improvement

Theorem 1 (Gradient alignment improvement of GARF)**.** Under Assumptions 1–3, there exists $\beta_0 > 0$ such that, for every $0 < \beta < \beta_0$, the first-order GARF outer gradient has a strictly larger unnormalized inner product with G_{true} than the baseline first-order outer gradient:

$$\langle G_{\text{true}}, G_{\text{GA}}(\beta)\rangle > \langle G_{\text{true}}, G_B\rangle. \tag{26}$$

Proof. By Proposition 1,

$$\begin{aligned}\langle G_{\text{true}}, G_{\text{GA}}(\beta)\rangle - \langle G_{\text{true}}, G_B\rangle &= \langle G_{\text{true}}, C(\beta)\rangle \\ &= -\beta\langle G_{\text{true}}, P v_{\text{proxy}}\rangle + O(\beta^2).\end{aligned} \tag{27}$$

Let $\kappa = -\langle G_{\text{true}}, P v_{\text{proxy}}\rangle > 0$ by Assumption 1. For fixed θ, the scalar remainder is bounded in magnitude by $K\beta^2$ for some $K > 0$ in a neighborhood of zero. Choose β_0 within this neighborhood with $\beta_0 \le \kappa/(2K)$. For $0 < \beta < \beta_0$, the difference is at least $\kappa\beta/2 > 0$. □

Remark. The theorem is a conditional local inner-product comparison. It does not imply larger cosine similarity, smaller Euclidean distance to G_{true}, or a guaranteed decrease in risk or ASR after a finite model update. Neither the sensitivity gap nor the interior-solution conditions are asserted to hold throughout training. The analysis also does not bound the approximation error of the finite projected inner iterations or guarantee improvement for the experimental values of β. For any fixed outer weight $\lambda > 0$, restoring that weight multiplies $C(\beta)$ by λ, preserving the sign of the local comparison; for $\lambda = 0$, the correction is zero.

B.3 Relation to Adversarial-Training-based Defenses

GARF follows a bilevel template similar to adversarial training: an inner problem generates adversarial perturbations and an outer problem updates the model using a combined loss. The key difference is that GARF explicitly models reactivation and backdoor directions, while standard adversarial training treats all adversarial perturbations in the same way.

(1) Surrogate Attacker under Progressive Backdoor Suppression. As post-training defense progresses, the backdoor signal in the model is progressively suppressed, and clean-data surrogate perturbations increasingly exploit generic decision-boundary weaknesses rather than the residual backdoor. This creates a growing mismatch: the true reactivation attacker starts from poisoned samples and directly targets the latent backdoor, whereas the defender's surrogate attacker—optimized purely on clean data—drifts away from the backdoor-relevant region as suppression advances. Consequently, the outer model update receives a biased gradient signal that fails to expose and suppress the directions an adaptive reactivation attacker would actually exploit. GARF addresses this by continuously steering the inner surrogate toward the most current estimate of the backdoor direction, so that even as the backdoor is suppressed, the simulated attacker remains focused on the reactivation-relevant subspace rather than scattering toward arbitrary adversarial directions.

(2) Task-Vector-Guided Search for Reactivation-Relevant Perturbations. To steer the surrogate attacker without access to poisoned samples, GARF adopts a task-vector interpretation of the parameter difference $v_{\text{proxy}} = \boldsymbol{\theta}_b - \theta$ (Ilharco et al., 2023). The sensitivity exposed by ANP (Wu and Wang, 2021) and the backdoor-connected paths studied by PAM (Min et al., 2024b) motivate examining directions toward the original backdoored model. Appendix E.4 provides complementary evidence for our chosen displacement: moving along it restores backdoor behavior with comparatively small changes in clean accuracy, and perturbing it degrades defense performance. GARF therefore guides the inner search using the relationship between the induced gradients and this proxy direction. The proxy is recomputed at every outer step. Its role is to supply reactivation-relevant guidance; the method does not require it to isolate a pure backdoor component in parameter space.

C Details of Experiment Settings

In this section, we detail the datasets, training protocols, attack configurations, and defense configurations used in our experiments. Unless otherwise specified, all attacks and defenses are implemented based on the official BackdoorBench framework (Wu et al., 2025a), and their hyperparameters follow the default settings therein.

C.1 Datasets and Model Architectures

Our experiments are conducted on three benchmark datasets: CIFAR-10 (Krizhevsky et al., 2009), Tiny (Le and Yang, 2015), and GTSRB (Stallkamp et al., 2011). For CIFAR-10, GTSRB and Tiny, we follow the conventional train and test splits in BackdoorBench and apply standard data augmentations including random cropping and random horizontal flipping. We collect the OOD dataset of CIFAR-10 from the same 10 classes of CIFAR-5m (Nakkiran et al., 2020).

We consider three widely used architectures in backdoor literature: PreAct-ResNet18 (He et al., 2016), VGG19-BN (Simonyan and Zisserman, 2014) and VIT-B-16 (Dosovitskiy et al., 2021). Unless otherwise noted, PreAct-ResNet18 is used as the default backbone in the main tables, while the results on VGG19-BN and on additional datasets such as Tiny and GTSRB are reported in the supplementary material.

C.2 Detailed Training Settings

For training potentially backdoored models on poisoned datasets, we largely follow the standard training protocol in BackdoorBench. All models are trained with Stochastic Gradient Descent (SGD) using an initial learning rate of 0.1, momentum of 0.9, and weight decay of 1×10^{-4}. For VGG19-BN, we reduce the initial learning rate to 0.01 for improved stability. We adopt a multi-step learning rate schedule with decays at fixed epochs and train for 100 epochs on all datasets unless otherwise specified.

For the post-training defenses and our proposed GARF, we fine-tune starting from a possibly backdoored checkpoint. All methods are allowed to access only a small fraction of benign training data. Specifically, following BackdoorBench, each defense accesses 5% of the original training set as a clean subset, while the rest of the training data is assumed unavailable in the post-training setting.

C.3 Detailed Attack Settings

Backdoor attacks. We evaluate our method under a diverse set of training-time backdoor attacks as implemented in BackdoorBench (Wu et al., 2025a). Unless otherwise stated, all hyperparameters (e.g., trigger size, location, blending factor) follow the default settings in BackdoorBench, and the poisoning ratio is fixed to 10% across all attack types. We include both all-to-one (A2O) and all-to-all (A2A) variants of BadNets, as well as several more advanced attacks:

- **BadNets-A2O/A2A** (Gu et al., 2019): The classic patch-based backdoor that stamps a fixed trigger patch on a local region of the image. We adopt both all-to-one and all-to-all variants; in the A2A case, the target label is reassigned as $y_t = (y+1) \bmod K$ for each class y, where K is the number of classes.

- **Blended** (Chen et al., 2017): A global blended backdoor where a fixed pattern is overlaid on the whole image using an α-blending scheme. We follow the default pattern and blending factor.
- **Label-consistent (LC)** (Turner et al., 2019): A clean-label attack that only poisons images originally belonging to the target class by adding a subtle trigger while keeping their labels unchanged. Due to the requirement of sufficient target-class samples, LC is only applied on CIFAR-10; for Tiny and GTSRB, the target-class sample size is insufficient to support a 10% poisoning ratio.
- **Low-frequency (LF)** (Zeng et al., 2021): An attack that constructs a smooth low-frequency trigger by filtering a universal adversarial perturbation. The trigger is visually smooth and less noticeable than high-frequency patterns.
- **Sample-specific backdoor attack (SSBA)** (Li et al., 2021c): A sample-specific attack where a generator learns to produce an adaptive trigger for each input, leading to stronger stealthiness.
- **BppAttack** (Wang et al., 2022): An attack that constructs triggers using image quantization and dithering, and injects them through contrastive adversarial learning.
- **Trojan attack (TrojanNN)** (Liu et al., 2018b): A Trojan attack that jointly optimizes the trigger and the last-layer parameters to maximize targeted misclassification under trigger presence.
- **Input-aware** (Nguyen and Tran, 2020): An input-aware dynamic backdoor that uses a trigger generator to produce sample-specific triggers conditioned on the input image.
- **Warping-based attack (WaNet)** (Nguyen and Tran, 2021): A warping-based attack that applies a smooth spatial transformation as the backdoor pattern rather than directly modifying pixel values.

Reactivation Attacks. To measure the vulnerability of defended models to backdoor reactivation, we adopt both white-box and black-box reactivation attacks following (Zhu et al., 2024a). In both settings, the attacker learns a universal reactivation perturbation on poisoned samples from the training split and evaluates it unchanged on triggered held-out test samples, with the defended model kept fixed throughout perturbation optimization and evaluation. The white-box attack performs a projected gradient-based search for a small perturbation that minimizes the target-class loss under an ℓ_∞ budget ε. The black-box attack uses a query-based optimizer to search for a similar perturbation without access to model gradients. In all cases, the perturbation budget and attack iterations follow the default configuration in (Zhu et al., 2024a); details of ε and the number of iterations are consistent across all methods and datasets.

C.4 Detailed Defense Settings

We evaluate our proposed GARF under a unified post-training defense protocol and compare it with several representative state-of-the-art (SOTA) post-training defenses, covering diverse mitigation paradigms. All defenses are implemented following BackdoorBench (Wu et al., 2025a), and their recommended hyperparameters are used unless otherwise specified. Each defense has access to 5% benign training data and operates solely in a post-training manner without using poisoned samples.

The compared defenses include:

- **Neural Cleanse (NC)** (Wang et al., 2019): NC searches for minimal triggers for each class by solving an inverse problem, and treats classes with unusually small triggers as suspicious. In our post-training setup, once a trigger is identified, the model is fine-tuned on a filtered set to mitigate the backdoor effect.
- **NAD** (Li et al., 2021b): NAD distills knowledge from a teacher network trained on the clean subset into the suspicious student network, thereby suppressing backdoor behaviors while preserving clean performance.
- **I-BAU** (Zeng et al., 2022): I-BAU formulates backdoor removal as a minimax problem on a small clean dataset and uses implicit hypergradients to account for the dependence between the inner adversarial optimization and the outer model update.
- **FT-SAM** (Zhu et al., 2023): FT-SAM fine-tunes the model with Sharpness-Aware Minimization, which improves robustness to backdoor triggers by seeking flatter minima using the clean subset.
- **SAU** (Wei et al., 2023): Shared Adversarial Unlearning constructs adversarial examples shared by the original backdoored model and the model being purified, and unlearns these examples using a small clean dataset to mitigate backdoor behavior.

Table 5: Performance results in percentage of post-training defense methods on GTSRB using PreAct-ResNet18 with a poisoning ratio of 10.0%. The metrics include Clean Accuracy, Attack Success Rate, White-box Reactivation ASR, and Black-box Reactivation ASR. **The best results are highlighted in bold.**

Defense →	No Defense				SAU				FST				NAD			
Attack ↓	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B
Blended	98.34	99.95	–	–	95.60	4.63	77.21	71.79	94.45	4.56	80.96	78.85	**98.38**	99.85	95.82	94.65
BadNets	97.74	94.75	–	–	94.95	**0.00**	31.52	**0.02**	95.13	2.29	56.66	48.72	97.47	0.01	64.24	58.55
SSBA	98.15	99.55	–	–	96.20	2.44	47.20	43.21	97.67	1.75	65.96	58.24	**98.25**	99.47	99.25	98.29
LF	97.97	99.40	–	–	19.56	**0.00**	**0.00**	**0.00**	98.28	30.46	52.47	46.29	98.03	34.25	98.97	98.19
TrojanNN	98.67	100.00	–	–	95.22	0.05	35.66	26.88	96.00	1.13	61.67	56.09	**98.58**	97.43	99.50	98.08
BadNets-A2A	97.14	87.71	–	–	93.94	0.59	60.32	52.19	93.13	4.01	57.99	52.59	97.29	0.55	58.68	50.87
WaNet	96.79	95.61	–	–	98.18	0.05	63.24	52.01	98.49	3.35	66.89	55.16	98.77	0.41	95.49	92.06
Input-aware	98.16	96.54	–	–	98.15	**0.00**	34.83	15.09	97.59	3.23	62.35	14.88	98.49	99.64	99.26	99.92
BppAttack	98.66	97.96	–	–	98.57	0.48	24.02	10.16	97.40	2.30	48.41	35.34	99.14	0.48	50.26	34.71
Average	97.96	96.83	–	–	87.82	0.92	41.56	30.15	96.46	5.90	61.48	49.57	98.27	48.01	84.61	80.59

Defense →	I-BAU				NC				FT-SAM				GARF			
Attack ↓	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B
Blended	83.69	57.96	99.72	98.47	59.25	5.70	99.98	99.78	97.75	1.77	79.22	60.55	97.49	**0.76**	**38.21**	**12.06**
BadNets	95.15	0.05	55.56	52.12	89.16	0.29	98.31	94.03	**98.45**	0.00	57.43	47.61	96.63	0.00	**27.44**	3.84
SSBA	47.56	**0.00**	**0.00**	**0.00**	66.23	0.23	99.67	99.34	97.75	0.01	57.97	27.61	98.13	0.05	34.43	12.82
LF	77.20	0.00	3.32	1.51	88.21	1.69	99.53	99.57	**98.32**	0.24	60.96	27.61	97.92	0.00	0.00	0.00
TrojanNN	21.19	**0.00**	**0.00**	**0.00**	80.86	1.42	100.00	100.00	96.43	0.09	55.63	44.85	97.89	0.00	30.39	19.29
BadNets-A2A	70.84	4.69	31.06	24.28	87.32	1.70	68.16	58.95	**98.44**	**0.30**	49.83	38.47	97.01	0.39	**30.41**	**16.83**
WaNet	98.22	0.01	47.18	35.68	96.90	0.11	100.00	99.96	**99.08**	0.03	70.01	57.61	98.41	**0.00**	**42.68**	**18.47**
Input-aware	96.74	0.00	27.61	14.93	96.57	0.01	98.67	98.10	**98.81**	72.45	30.90	12.61	98.12	0.00	**24.57**	**0.00**
BppAttack	97.69	0.54	48.84	35.18	97.68	**0.42**	99.75	98.80	**99.47**	0.48	56.17	48.47	99.13	0.48	**21.74**	**7.25**
Average	76.47	7.03	34.81	29.13	84.69	1.29	96.01	94.28	**98.28**	8.37	57.57	40.60	97.86	**0.19**	**27.76**	**10.06**

- **FST** (Min et al., 2024a): Feature Shift Tuning encourages feature shifts by moving the classifier weights away from the original compromised weights during clean-data fine-tuning.
- **GARF**(Ours): GARF is applied in a post-training manner on a defended or suspicious model f_θ using only a small clean subset of the training data; on CIFAR-10, Tiny and GTSRB we use 5% of the original training images. The outer reactivation term has weight $\lambda = 1$ on all three datasets. The lower-level problem introduces a gradient-alignment penalty with weight β, set to $\beta = 10$ on CIFAR-10 and $\beta = 20$ on both Tiny and GTSRB. The perturbation δ is constrained to an ℓ_∞ ball of radius $\varepsilon = 0.05$ and is optimized by projected SGD with step size $\alpha = 0.05$ for $T = 5$ inner iterations per outer update and 20 outer epochs by default; the plug-in study separately evaluates shorter GARF refinements.

D Additional Experiments

D.1 Main Experiments on Different Datasets and Models

We evaluate GARF on additional datasets and models to check whether the observations on CIFAR-10 transfer to different label spaces, image resolutions, visual domains, and backbone designs.

On GTSRB with the PreAct ResNet18 backbone, reported in Tab. 5, the backdoored models again suffer from very high attack success across diverse triggers including patch based, sample specific, low frequency, Trojan style, warping and input aware attacks. Classical post training defenses such as SAU, FST and NAD notably reduce the standard attack success but still allow strong reactivation, especially for SSBA, LF and Input aware. I-BAU and NC can almost remove certain attacks, but typically incur a large drop in clean accuracy or very strong reactivation elsewhere. GARF instead keeps clean accuracy close to the strongest fine tuning based baseline while consistently reducing both white box and black box reactivation, with clear improvements on LF, SSBA and BppAttack. Reactivation is evaluated using a universal perturbation optimized on training samples and then applied unchanged to held-out test samples, without per-test-input optimization. Consequently, the learned perturbation need not increase test-set ASR over the original trigger, which can explain the occasional decrease after reactivation. More generally, GARF's suppression of residual backdoor features may further hinder the ability of such training-derived perturbations to generalize to unseen test inputs.

To further study the model dependence of GARF, we replace PreAct-ResNet18 with VGG19-BN on CIFAR-10, as reported in Tab. 6. Without defense, all attacks still achieve very high success while

Table 6: Performance in percent of post-training defenses on CIFAR-10 using VGG19-BN with a poisoning ratio of 10.0%. Metrics include Clean Accuracy ACC, Attack Success Rate ASR, White-box Reactivation ASR_W, and Black-box Reactivation ASR_B. **The best reactivation results are highlighted in bold.**

Defense →	No Defense				SAU				FST				NAD			
Attack ↓	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B
Blended	91.24	99.43	–	–	88.72	6.93	87.09	41.18	89.61	9.27	90.53	56.07	88.36	78.47	98.39	91.13
BadNets	90.58	95.21	–	–	88.91	0.39	83.18	31.69	89.77	1.71	86.87	50.82	87.62	6.14	93.13	73.17
SSBA	90.93	98.96	–	–	89.08	1.18	79.46	36.29	90.09	2.37	83.69	49.18	88.12	82.63	98.87	96.19
LF	89.84	97.31	–	–	87.84	0.81	71.27	17.42	88.49	1.43	78.23	21.28	85.94	49.27	95.06	77.93
TrojanNN	91.47	99.98	–	–	88.39	0.73	82.06	29.87	89.18	1.86	88.43	47.62	88.32	69.37	97.76	88.57
BadNets-A2A	89.65	88.07	–	–	87.97	1.92	76.42	34.09	88.72	3.09	81.56	43.87	86.69	11.73	90.27	69.41
WaNet	90.43	96.02	–	–	89.27	0.62	84.75	38.71	90.38	2.21	89.72	55.37	88.93	5.27	96.66	89.72
Input-aware	90.84	96.68	–	–	88.96	0.48	80.12	26.91	89.97	1.98	84.47	29.69	88.55	7.82	97.25	93.38
BppAttack	91.03	98.06	–	–	89.54	1.07	78.27	33.24	90.59	2.47	82.11	35.82	89.16	4.72	92.37	72.82
Average	90.67	96.64	–	–	88.74	1.57	80.29	32.16	89.64	2.93	85.07	43.30	87.97	35.05	95.53	83.59
Defense →	I-BAU				NC				FT-SAM				GARF			
Attack ↓	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B
Blended	86.06	47.23	95.28	82.94	86.86	13.39	99.07	95.69	90.61	8.35	70.39	43.68	89.43	2.57	**36.08**	**13.71**
BadNets	85.84	3.41	82.71	58.62	86.34	3.82	95.92	91.78	90.23	1.12	64.13	36.29	88.93	0.42	**28.36**	**7.87**
SSBA	83.65	1.79	76.43	47.47	85.69	7.18	98.46	97.83	90.92	1.72	59.63	32.54	89.29	0.94	**31.18**	**9.43**
LF	82.88	2.14	71.53	25.81	84.59	11.21	94.83	92.71	88.72	1.88	54.66	19.24	87.64	0.83	**22.73**	**5.59**
TrojanNN	84.72	5.93	88.61	63.09	85.93	9.69	97.38	94.27	89.82	2.39	67.32	41.76	88.72	1.13	**34.83**	**15.07**
BadNets-A2A	84.12	7.58	79.83	42.32	84.77	14.47	92.13	89.24	89.31	3.62	61.83	35.19	88.07	1.72	**29.57**	**11.28**
WaNet	86.57	2.73	87.14	54.18	86.52	5.92	96.81	93.58	90.65	2.06	69.92	53.38	89.73	0.71	**33.43**	**17.82**
Input-aware	85.92	3.28	80.28	40.73	86.08	6.84	97.87	96.13	90.27	2.74	63.47	28.62	89.16	0.92	**27.87**	**8.68**
BppAttack	86.83	4.06	81.94	52.57	86.72	5.13	93.68	90.46	90.86	3.23	60.73	34.91	89.79	1.27	**26.84**	**6.89**
Average	85.18	8.68	82.64	51.97	85.94	8.63	96.24	93.52	90.15	3.01	63.56	36.18	88.97	**1.17**	**30.10**	**10.70**

the clean accuracy remains high, consistent with the ResNet results. Standard post-training defenses such as SAU, FST and FT-SAM substantially reduce the standard attack success and preserve most of the clean accuracy, but the defended models continue to exhibit strong white-box and black-box reactivation. Other methods including NAD, I-BAU and NC can further suppress some attacks, yet typically suffer from high reactivation or a noticeable drop in clean accuracy. In contrast, GARF maintains clean accuracy comparable to FT-SAM and consistently yields lower reactivation on almost all attacks. Taken together, these results indicate that modeling and minimizing reactivation risk is beneficial across different datasets and architectures, and that the robustness gain of GARF is stable rather than tied to a particular backbone.

Table 7: Performance in percent of post-training defenses on Tiny-ImageNet using ViT-B-16 with a poisoning ratio of 10.0%. Metrics include Clean Accuracy ACC, Attack Success Rate ASR, White-box Reactivation ASR_W, and Black-box Reactivation ASR_B. **The best reactivation results are highlighted in bold.**

Defense →	No Defense				SAU				FST				NAD			
Attack ↓	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B
Blended	75.32	99.81	–	–	69.38	6.71	92.06	39.58	71.79	3.47	89.28	54.19	73.88	73.42	97.83	89.47
BadNets	74.19	99.12	–	–	70.72	0.39	87.64	27.87	72.51	1.09	81.72	44.28	73.02	3.61	85.94	59.08
SSBA	76.13	98.82	–	–	69.63	1.68	84.19	34.72	73.21	0.91	79.37	41.83	74.68	81.19	98.73	94.12
LF	73.37	98.29	–	–	68.21	0.58	79.33	5.41	69.03	1.43	75.06	10.87	69.44	41.47	91.55	71.18
Average	74.75	99.01	–	–	69.48	2.34	85.80	26.89	71.63	1.73	81.36	37.79	72.75	49.92	93.51	78.46
Defense →	I-BAU				NC				FT-SAM				GARF			
Attack ↓	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B
Blended	68.71	41.27	95.79	84.13	62.93	10.83	99.37	96.84	71.20	8.88	69.13	46.31	70.50	2.42	**33.71**	**11.93**
BadNets	70.82	1.27	77.18	53.69	65.38	2.81	94.53	91.68	71.60	0.62	61.81	36.09	70.80	0.32	**24.91**	**6.72**
SSBA	60.48	0.37	69.32	41.18	66.12	5.71	98.62	97.86	71.80	1.26	57.39	31.63	72.00	0.83	**28.73**	**8.81**
LF	68.17	0.71	64.83	21.27	68.56	7.57	95.94	93.53	70.24	1.88	52.57	16.69	69.02	0.47	**19.34**	**4.52**
Average	67.04	10.91	76.78	50.07	65.75	6.73	97.11	94.98	71.21	3.16	60.22	32.68	70.58	**1.01**	**26.67**	**8.00**

On Tiny with a ViT based backbone, shown in Tab. 7, all attacks yield very high attack success on the backdoored model while keeping reasonable clean accuracy. Standard post training defenses such as SAU, FST and FT-SAM clearly reduce the standard attack success, but the defended models remain easy to reactivate, especially under strong and stealthy attacks such as SSBA and Blended. I-BAU and NC sometimes further lower the attack success, yet often at the cost of degraded clean accuracy or large reactivation. In contrast, GARF maintains clean performance similar to FT-SAM

Table 8: White-box and black-box reactivation attack success rates on CIFAR-10 with PreAct-ResNet18 under four attacks. All defenses except GARF are augmented by PAM as described in the text. Lower values indicate better robustness. The best results in each row are highlighted in bold.

Defense →	SAU+PAM		FT-SAM+PAM		NAD+PAM		I-BAU+PAM		PAM		GARF	
Attack ↓	ASR_W	ASR_B	ASR_W	ASR_B	ASR_W	ASR_B	ASR_W	ASR_B	ASR_W	ASR_B	ASR_W	ASR_B
Blended	76.88	56.86	83.31	65.40	86.20	85.23	83.03	71.91	71.34	49.27	**38.11**	**15.89**
BadNets	79.69	21.57	85.70	48.06	89.62	40.43	75.41	40.10	75.42	17.83	**39.67**	**0.01**
SSBA	70.94	34.29	80.67	53.03	82.36	73.58	75.13	52.94	66.18	29.74	**3.23**	**0.00**
LF	80.12	6.04	85.24	77.10	97.15	80.34	82.26	58.23	73.07	4.82	**32.42**	**1.29**

and consistently lowers both white box and black box reactivation, with particularly visible gains on LF and Blended, showing that gradient aligned bilevel optimization is effective on a transformer based architecture.

D.2 Analysis with PAM

PAM promotes deviation from the backdoored model along a backdoor-connected path. At each iteration, it recomputes the parameter difference between the original and current models and evaluates gradients at an interpolated model using clean data and a fixed set of BTI-synthesized backdoor samples. GARF instead re-optimizes surrogate input perturbations on clean data at each outer iteration, guided by the current model and parameter-difference proxy.

We perform experiments on CIFAR-10 using PreAct-ResNet18 under Blended, BadNets, SSBA, and LF attacks, comparing GARF against PAM used as a standalone defense. Since PAM can also be integrated into existing post-training defenses such as SAU, FT-SAM, NAD, and I-BAU to further enhance their robustness, we additionally evaluate these combined variants. As seen in Tab. 8, adding PAM to existing defenses provides only moderate improvements in reactivation suppression, and both white-box and black-box reactivation ASR remain high across all four attacks. When used standalone, PAM similarly fails to reduce reactivation to a competitive level. In contrast, GARF achieves consistently lower reactivation across all four attacks.

The distinction concerns how surrogate attack inputs are constructed during defense. PAM reuses a precomputed synthetic backdoor set, whereas GARF continually refreshes its surrogate perturbations as the model evolves. This allows GARF to search for reactivation-relevant inputs under the current model without relying on a fixed reconstructed trigger set. The parameter-difference proxy guides this search without requiring exact recovery of the true backdoor direction. The results in Table 8 show lower reactivation ASR for GARF under the evaluated settings.

D.3 Analysis with BTI

BTI (Xu et al., 2024) synthesizes a dangerous subset of data by inverting the backdoor, and then either performs forgetting (BTI-Forget) or uses the subset for adversarial training (BTI-AdvTrain). We apply both variants to CIFAR-10 using PreAct-ResNet18 under Blended, BadNets, SSBA, and LF attacks under the same post-training data budget as other defenses. As shown in Table 9, BTI-based methods effectively reduce the standard attack success rate, yet still exhibit high reactivation ASR in both white-box and black-box settings, with particularly pronounced failures on Blended and SSBA. GARF achieves lower black-box reactivation ASR across all four attacks and lower white-box reactivation ASR on Blended, SSBA, and LF while maintaining competitive clean accuracy. On BadNets, BTI-AdvTrain has a slightly lower white-box reactivation ASR than GARF (38.92% versus 39.67%).

The BTI-based variants use a precomputed synthetic backdoor subset as the basis for forgetting or adversarial training. GARF instead re-optimizes surrogate perturbations on clean samples at each outer iteration using the current model and proxy $v_{\text{proxy}} = \boldsymbol{\theta}_b - \theta$. This distinction concerns the construction of surrogate attack inputs during defense. The comparison evaluates the complete methods and does not isolate this design choice as the cause of the observed performance differences.

Table 9: Comparison with BTI-DBF based defenses on CIFAR-10 with PreAct-ResNet18. BTI-Forget denotes forgetting on BTI-generated dangerous subsets. BTI-AdvTrain denotes adversarial training using the same synthetic subset. Lower ASR, ASR_W, and ASR_B are better, while higher ACC is better. **The best results per row are highlighted in bold.**

Defense →	BTI-Forget				BTI-AdvTrain				GARF			
Attack ↓	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B	ACC	ASR	ASR_W	ASR_B
Blended	**93.84**	40.13	96.26	91.51	92.06	10.12	62.47	37.38	91.92	**4.00**	**38.11**	**15.89**
BadNets	**93.12**	2.01	41.05	3.12	91.57	1.68	**38.92**	1.46	89.41	**1.48**	39.67	**0.01**
SSBA	**93.45**	2.28	45.73	14.68	91.84	1.95	28.36	9.57	90.71	**0.31**	**3.23**	**0.00**
LF	**93.23**	3.76	69.82	2.37	91.65	2.83	52.14	5.03	90.40	**0.43**	**32.42**	**1.29**

Table 10: Normalized training time of post-training defenses on CIFAR-10 with PreAct-ResNet18. Values are wall-clock time normalized by vanilla fine-tuning (FT), averaged over multiple runs. Lower is better.

Defense	CIFAR-10
FT	1.00
NAD	2.35
I-BAU	3.72
FT-SAM	2.41
SAU	2.73
GARF	2.70

E Additional Analysis

E.1 Computational Cost

We further compare the computational cost of different post-training defenses on CIFAR-10 with PreAct-ResNet18. For each method, we measure the wall-clock training time under the same number of fine-tuning epochs and normalize it by the cost of vanilla fine-tuning on CIFAR-10. The normalized results are summarized in Tab. 10.

The results show that advanced post-training defenses such as NAD and I-BAU incur a clear computational overhead compared with vanilla fine-tuning, with I-BAU being the most expensive among the compared methods. FT-SAM and SAU lie in the middle range, providing improved robustness at a moderate increase in time. GARF introduces a time cost that is very similar to SAU and remains substantially cheaper than I-BAU while delivering significantly stronger robustness against reactivation as demonstrated in previous sections. Overall, GARF achieves its robustness gains with a computational overhead that is comparable to existing strong fine-tuning based defenses on CIFAR-10.

E.2 Ablation Studies on Target Labels

Knowing the target label y_t is a common assumption adopted by many backdoor defenses (Xu et al., 2024; Qi et al., 2023b) to improve defense effectiveness, as it allows the defender to focus the unlearning objective on the attacker-chosen class rather than treating all classes uniformly. Several methods have been proposed to infer y_t directly from a backdoored model without access to poisoned data, including Neural Cleanse (NC) (Wang et al., 2019), BTI-DBF (Xu et al., 2024), Unicorn (Wang et al., 2023), and DISTIL (Mirzaei et al., 2025), each of which exploits different model behaviors to identify the most likely target class.

To obtain a reliable estimate of y_t for GARF's inner-level optimization, we evaluate each of the above detectors individually and further combine them via an ensemble strategy. Specifically, each detector independently produces a candidate target label from the backdoored model, and the final label $\hat{y}_t$ is determined by majority vote across all detectors. In the case of a tie, we select the candidate returned by the detector with the highest historical accuracy on the evaluated dataset. We denote the detected

target by $\hat{y}_t$ and substitute it for the target label in the empirical defense objective. The theoretical analysis considers the target-matched case $\hat{y}_t = y_t$.

Tab. 11 reports detection success rates on CIFAR-10 for BadNets, SSBA, and LF. The ensemble achieves 100% on these evaluated configurations, including cases where individual detectors misidentify the target. These results support using the ensemble to obtain the target label in our experiments. When detection is ambiguous, we can instead defend sequentially against a set of candidate labels, as evaluated below.

Table 11: Target label detection success rate (%) of different detection methods on CIFAR-10.

Attack	NC	BTI-DBF	Unicorn	DISTIL	Ensemble
BadNets	100	100	100	100	100
SSBA	55	100	95	100	100
LF	60	95	85	95	100

Sequential defense with candidate target labels. Given multiple candidate targets, we apply the label-conditioned GARF defense to them in sequence, continuing to update the same model rather than creating an independent model for each target. At each stage, the inner objective uses the current target label, while the outer objective retains the true clean labels. The original checkpoint $\boldsymbol{\theta}_b$ remains fixed, and the global proxy $\boldsymbol{\theta}_b - \boldsymbol{\theta}$ is recomputed as the model is updated. Thus, target conditioning changes the inner objective; it does not introduce a separate parameter-difference proxy for each target.

Tab. 12 evaluates candidate sets containing the correct target on CIFAR-10 with BadNets. Covering two or three candidates yields white-box reactivation ASR of 35.35% and 37.89%, respectively, compared with 39.67% for the single correct label. Clean accuracy decreases from 89.41% to 86.37% and 84.52%, showing the utility cost of broader label coverage. These results establish robustness to additional incorrect candidates when the correct label is covered; they do not evaluate a candidate set that omits the true target.

Multiple triggers and shared-trigger all-to-all attacks. For multiple triggers associated with distinct targets, we apply the same defense sequentially to the target labels, using one evolving model and its shared proxy. In the two-trigger/two-target experiment, GARF achieves 87.16% ACC, 1.47% ASR, and 34.69% white-box reactivation ASR (Tab. 13).

For BadNets-A2A, every class-dependent target mapping $y_t = (y + 1) \bmod K$ uses the same trigger. We use a single target label for defense in this setting: the label-conditioned objective suppresses shared trigger-related behavior that supports multiple target mappings. The BadNets-A2A result in Tab. 2, reproduced in Tab. 13, shows that this procedure achieves 1.46% ASR, 47.64% white-box reactivation ASR, and 1.19% black-box reactivation ASR at 90.23% clean accuracy. This demonstrates the effectiveness of single-label defense in the evaluated shared-trigger setting.

E.3 Robustness to Adaptive Attacks

Our evaluations concern post-defense reactivation: the final defended model remains fixed while the attacker optimizes input perturbations. The proxy is updated during defense training, not during this attack phase. We examine a defense-aware black-box objective and increased white-box perturbation budgets as complementary tests of this setting.

GARF uses gradient alignment to construct stronger surrogate reactivation perturbations during defense. A natural concern is whether an adaptive attacker who knows this design can explicitly avoid the proxy direction $\boldsymbol{\theta}_b - \theta$ and thereby bypass the defense. We consider a challenging adaptive threat model: the attacker is aware of the GARF defense strategy and attempts to circumvent the gradient-alignment mechanism, but has only black-box query access to the defended model. To approximate the proxy direction, the attacker maintains a surrogate model θ trained to mimic the defended model's query outputs. The adaptive attacker then solves

$$\min_{\delta} \mathcal{L}_{\mathrm{CE}}(f_\theta(\tilde{x} + \delta), y_t) + \alpha \cdot |A\left(\nabla_\theta \mathcal{L}_{\mathrm{CE}}(f_\theta(\tilde{x} + \delta), y_t), \boldsymbol{\theta}_b - \theta\right)|, \tag{28}$$

Table 12: Sequential defense over candidate target labels on CIFAR-10 with BadNets. Each candidate set contains the correct target. All metrics are percentages.

Target-label handling	ACC	ASR	ASR_W	ASR_B
Single correct label	89.41	1.48	39.67	0.01
Two candidates	86.37	1.26	35.35	1.42
Three candidates	84.52	0.98	37.89	0.59

Table 13: Defense under different trigger–target configurations on CIFAR-10. All metrics are percentages. The all-to-one baseline and BadNets-A2A results are repeated from Tab. 2 for comparison.

Attack configuration	ACC	ASR	ASR_W	ASR_B
One trigger, one target	89.41	1.48	39.67	0.01
Two triggers, two targets	87.16	1.47	34.69	0.56
One shared trigger, K targets (A2A)	90.23	1.46	47.64	1.19

where θ denotes the surrogate model, and $\alpha \geq 0$ controls the strength of the orthogonality penalty. When $\alpha = 0$, this reduces to the standard black-box reactivation attack. When $\alpha = 1$, the attacker additionally steers perturbations so that their surrogate-estimated gradients are orthogonal to the estimated proxy direction, directly targeting the key mechanism used by GARF.

We evaluate this adaptive orthogonal attack on CIFAR-10 with PreAct-ResNet18. Tab. 14 reports the black-box reactivation ASR for BadNets and LF. Under the standard attack ($\alpha = 0$), GARF achieves very low reactivation ASR: 0.01% on BadNets and 1.29% on LF. Under the adaptive orthogonal attack ($\alpha = 1$), the ASR remains almost unchanged, with 0.03% on BadNets and 1.26% on LF. These results show that explicitly forcing the surrogate-estimated attack gradient away from the proxy direction does not meaningfully increase the reactivation success rate.

The orthogonality penalty changes the attacker's search objective on a surrogate model; it is not a constraint enforced by GARF at inference time. The low ASR in these two tests shows that this particular mechanism-aware black-box objective does not expose a stronger reactivation attack under the evaluated settings. It does not rely on the attacker being unable to track a changing proxy after defense.

Table 14: Black-box reactivation ASR (%) under orthogonal adaptive attacks on CIFAR-10 (PreAct-ResNet18).

Attack	$\alpha = 0$ (standard)	$\alpha = 1$ (orthogonal)
BadNets	0.01	0.03
LF	1.29	1.26

White-box attacks with larger perturbation budgets. We additionally evaluate white-box reactivation on CIFAR-10 with BadNets at ℓ_∞ budgets $\varepsilon \in \{0.05, 0.10, 0.20\}$, retaining the attack's 50 outer and 5 inner iterations across this sweep. All three defenses are evaluated at the same budgets. As shown in Tab. 15, GARF has lower reactivation ASR than SAU and FT-SAM at $\varepsilon = 0.05$ and 0.10. At $\varepsilon = 0.20$, all three methods approach 100% ASR, indicating that the advantage does not persist under arbitrarily enlarged perturbations.

Together with the standard white-box and black-box evaluations, these experiments assess robustness under the specified access conditions, objectives, and perturbation budgets. The budget sweep does not evaluate additional restarts or alternative white-box objectives; those search strategies remain separate dimensions of adaptive evaluation. Attacks that manipulate defense updates instead change the post-defense setting considered here.

Table 15: White-box reactivation ASR (%) under increasing ℓ_∞ budgets on CIFAR-10 with BadNets. The standard-budget row ($\varepsilon = 0.05$) repeats the corresponding results in Tab. 2.

Budget ε	GARF	SAU	FT-SAM
0.05	39.67	92.74	94.86
0.10	68.54	98.91	99.99
0.20	99.92	99.96	99.98

Table 16: Clean accuracy and standard ASR (%) along the proxy direction on CIFAR-10. The endpoints are the defended model ($t = 0$) and the original backdoored model ($t = 1$).

Metric	$t = 0$	$t = 0.25$	$t = 0.5$	$t = 0.75$	$t = 1$
ACC	89.41	90.26	91.42	90.93	91.82
ASR	1.48	12.39	50.21	80.29	93.79

E.4 Empirical Analysis of the Proxy Backdoor Direction

We examine whether the displacement $v_{\text{proxy}} = \boldsymbol{\theta}_b - \boldsymbol{\theta}$ identifies a direction along which backdoor behavior is recovered, and whether its quality affects the defense. These analyses complement the backdoor-sensitivity and path-based observations motivating the method (Wu and Wang, 2021; Min et al., 2024b).

Backdoor recovery along the proxy direction. On CIFAR-10, we interpolate between the defended model and the original backdoored model:

$$\boldsymbol{\theta}(t) = \boldsymbol{\theta} + t(\boldsymbol{\theta}_b - \boldsymbol{\theta}), \qquad t \in \{0, 0.25, 0.5, 0.75, 1\}. \tag{29}$$

Tab. 16 reports clean accuracy and the standard ASR under the original trigger at each checkpoint. ASR increases monotonically from 1.48% to 93.79%, including substantial recovery at intermediate points, while ACC remains within a 2.41 percentage-point range. Thus, in this experiment, moving along the displacement restores backdoor behavior while largely preserving clean classification performance. This supports treating the displacement as a backdoor-relevant direction, without requiring a decomposition into exclusively benign and malicious parameters.

Sensitivity to proxy corruption. For BadNets on CIFAR-10, we replace the proxy by $v'_{\text{proxy}} = v_{\text{proxy}} + \eta$, where a random vector η is rescaled to satisfy $\|\eta\| = \rho\|v_{\text{proxy}}\|$. The relative noise level ρ is distinct from the outer loss weight λ. As shown in Tab. 17, noise at $\rho = 0.001$ leaves white-box reactivation ASR close to the original value, whereas larger noise levels progressively degrade robustness: ASR_W reaches 56.32% at $\rho = 0.01$ and 97.36% at $\rho = 1$. These results show that the quality of the proxy matters for the observed defense gains and identify a practical sensitivity to inaccurate directions.

Taken together, the interpolation and corruption experiments provide empirical support for the proxy's reactivation-relevant role. They establish its utility in the evaluated setting, rather than a necessary direction for every possible reactivation attack.

Similarity choice and inner optimization. Tab. 18 compares alignment choices and a random-perturbation variant on CIFAR-10 with BadNets. The dot-product-alignment configuration serves as the reference in this comparison. Cosine and L2-based variants yield white-box reactivation ASRs of 39.47% and 38.68%, respectively, close to the dot-product result of 39.67%, while maintaining similar clean accuracy. Thus, the observed reactivation suppression is not specific to the dot-product choice in this setting. Replacing the inner perturbation optimization with a random perturbation instead raises standard ASR to 90.36% and white-box and black-box reactivation ASRs to 99.76% and 95.86%. This comparison supports the role of the optimized inner perturbation in the defense. Together with the existing $\beta = 0$ ablation, which removes gradient alignment, these results provide complementary evidence for the design.

Table 17: Proxy corruption on CIFAR-10 with BadNets. All metrics are percentages; $\rho = 0$ is the unmodified proxy.

Relative noise ρ	ACC	ASR	ASR_W	ASR_B
0	89.41	1.48	39.67	0.01
0.001	88.49	2.02	38.38	1.41
0.01	88.53	3.52	56.32	4.82
0.1	87.29	4.69	88.46	36.23
1	89.36	7.32	97.36	58.65

Table 18: Alignment choices and random-perturbation ablation on CIFAR-10 with BadNets. All metrics are percentages. The random variant replaces the inner perturbation optimization; the dot-product row is the reference GARF configuration for this comparison.

Variant	ACC	ASR	ASR_W	ASR_B
GARF (dot-product alignment)	89.41	1.48	39.67	0.01
Cosine alignment	89.79	1.58	39.47	1.49
L2-based alignment	88.68	2.16	38.68	1.07
Random perturbation	90.32	90.36	99.76	95.86

E.5 Heterogeneity of backdoor perturbations

GARF constructs its proxy from the displacement between the original checkpoint and the current model. To characterize variation across attacks, we report pairwise cosine similarities between reactivation-induced parameter gradients for BadNets, LF, and SSBA on CIFAR-10 with PreAct-ResNet18.

As shown in Tab. 19, the measured pairwise cosine similarities range from 0.04 to 0.06. These values describe low similarity between the gradients in this comparison; they do not establish the absence of shared backdoor-related components.

The proxy $\boldsymbol{\theta}_b - \boldsymbol{\theta}$ is recomputed at each outer step to track the current displacement from the original checkpoint. The cross-attack comparison does not test whether updating this proxy is necessary or superior to freezing it during a single defense run. Appendix E.4 separately evaluates its relation to backdoor recovery and the defense's sensitivity to proxy corruption.

Table 19: Cosine similarity between parameter-space gradients induced by reactivation perturbations of different attacks on CIFAR-10 (PreAct-ResNet18).

Attack Pair	Cosine Similarity
BadNets vs. LF	0.05
BadNets vs. SSBA	0.04
LF vs. SSBA	0.06