STATIC BACKDOORS, DYNAMIC TRIGGERING: RETHINKING THE STATIC–DYNAMIC DISTINCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Static backdoors repeatedly use fixed triggers, producing stable responses that can be exploited by runtime detection, whereas dynamic backdoors improve stealth through trigger diversity but typically require additional training. This raises two questions: can a backdoor implanted with a fixed trigger be activated by alternative trigger configurations without retraining, and, if so, will existing black-box runtime detectors remain reliable under such triggering behavior? We find that static implantation does not necessarily imply static triggering, as different trigger configurations can induce similar target-oriented responses, allowing a static backdoor implanted with a fixed trigger to remain activatable by multiple alternative triggers after training. Based on this observation, we propose SBDT (Static-Backdoor Dynamic Triggering), a post-training dynamic triggering framework that discovers effective alternative triggers and diversifies their use across samples without retraining, fine-tuning, or modifying model parameters. SBDT preserves effective attack performance while reducing the average true positive rate under BBCaL and SCALE-UP from 99.83% to 0.37%. These results show that static implantation does not necessarily constrain a backdoor to a single fixed triggering configuration after training, motivating a reconsideration of the conventional static–dynamic distinction. They also suggest that the effectiveness of black-box runtime detection can be sensitive to post-training trigger configurations, and that runtime detection is therefore better viewed as a complement to, rather than a replacement for, model-level backdoor auditing and mitigation.

1 INTRODUCTION

Backdoor attacks remain a fundamental security threat to deep neural networks and are commonly categorized into static and dynamic variants. Static backdoors use a fixed trigger to establish a stable association with the target class (Gu et al., 2017; Chen et al., 2017; Wang et al., 2022a). They are simple and effective, but repeatedly using the same trigger at inference time often induces stable input-output anomalies or feature responses, providing exploitable signals for clustering-, spectral-, statistic-, and test-time consistency-based detectors (Chen et al., 2019; Tran et al., 2018; Tang et al., 2021; Ma et al., 2023; Liu et al., 2023). In contrast, dynamic backdoors introduce multiple or sample-specific triggers to increase trigger diversity (Cheng et al., 2024; Li et al., 2021c), but such diversity is typically designed during backdoor implantation and requires additional poisoning-based training or model optimization.

Yet a fixed trigger used for implantation need not be the only configuration capable of activating the resulting backdoor. If a well-established static attack such as BadNets can be activated through alternative trigger configurations after training, it could be given dynamic triggering behavior without redesigning the attack or updating the model. This possibility raises a practical defense question: *can black-box runtime detection reliably identify such inputs, or does effective protection still require model-level auditing and mitigation?* It also challenges the conventional static–dynamic distinction, which ties a backdoor's training-time implantation to its inference-time triggering behavior.

We find that **static implantation does not necessarily imply static triggering**. As shown in Figure 1, by scanning the trigger configuration spaces of representative static backdoors, including BadNets (Gu et al., 2017), Blended (Chen et al., 2017), and FTrojan (Wang et al., 2022a), we observe multiple high-response regions beyond the original trigger configurations. These phenomena

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.

arise across spatial, blending-related, and frequency-domain variations, indicating that a static backdoor implanted with a fixed trigger can retain exploitable trigger-response flexibility after training and remain activatable by multiple alternative triggers.

Based on this observation, we propose **SBDT** (**S**tatic-**B**ackdoor **D**ynamic **T**riggering), a training-free framework for dynamically triggering already-implanted static backdoors after training. SBDT constructs and filters candidate triggers that can activate the existing backdoor, characterizes their response differences, and performs balanced sample-wise trigger assignment to achieve diversified triggering. The entire process requires no retraining, fine-tuning, or modification of model parameters, and changes only how the implanted backdoor is triggered after training, avoiding the additional training typically required by dynamic backdoors.

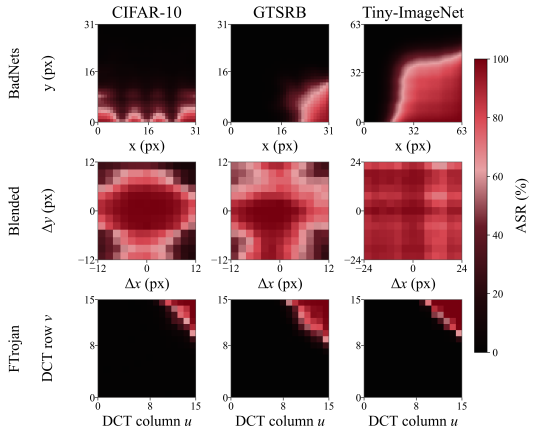


Figure 1: Trigger-response landscapes of representative static backdoors.

This phenomenon not only motivates a rethinking of the conventional static–dynamic distinction, but also exposes a potential limitation of black-box runtime detection. Our experiments show that even when the backdoored model itself remains unchanged, altering only its post-training triggering behavior can substantially reduce the effectiveness of multiple runtime detectors. This suggests that black-box runtime detection can be sensitive to post-training changes in trigger configuration, even when the backdoored model itself remains unchanged. Runtime detection should therefore be viewed as a complement to, rather than a replacement for, model-level backdoor auditing and mitigation.

Our main contributions are summarized as follows:

- We revisit the conventional static–dynamic distinction by separating backdoor implantation from inference-time activation behavior. Across representative static backdoors, we reveal exploitable multi-configuration response regions that support diversified inference-time activation while all model parameters remain frozen.
- We propose SBDT, a training-free post-training dynamic triggering framework that leverages latent trigger flexibility for diversified triggering through trigger discovery and characterization, response-aware modeling, and balanced sample-wise assignment, without retraining, fine-tuning, or parameter updates.
- We reveal that post-training triggering behavior can substantially affect backdoor detectability even when the backdoored model itself remains unchanged. On BadNets, SBDT reduces the average true positive rate under BBCaL and SCALE-UP from 99.83% to 0.37%, with reductions observed under additional runtime detectors and stronger-access diagnostics. These results highlight the need to complement runtime detection with model-level backdoor auditing and mitigation.

2 Related Work

2.1 Backdoor Attacks

Backdoor attacks embed a hidden association between a trigger pattern and an attacker-specified target class while preserving normal performance on clean inputs (Gu et al., 2017). Representative static backdoors include fixed pixel patches in BadNets (Gu et al., 2017), blended triggers (Chen et al., 2017), and frequency-domain triggers in FTrojan (Wang et al., 2022a). Subsequent studies improve stealthiness through sparse, imperceptible, quantization-based, or geometric triggers (Gao et al., 2024; Wang et al., 2022b; Nguyen & Tran, 2021).

Dynamic backdoors introduce multiple or sample-specific trigger patterns. ISSBA generates sample-specific invisible triggers (Li et al., 2021c), while LOTUS introduces trigger diversity through data sub-partitioning (Cheng et al., 2024). Recent studies further explore input-dependent, clean-label sample-specific, hidden-feature, frequency-domain, and semantic triggers (Li et al., 2025a; Zhu et al., 2025; Chen et al., 2025a;b; Wang et al., 2026). However, such diversity is typically designed during backdoor implantation. SBDT instead systematically exploits post-training trigger flexibility for sample-wise dynamic triggering while keeping the statically backdoored model fixed, separating implantation from inference-time triggering behavior.

2.2 Backdoor Detection

Backdoor detection methods can be broadly divided into model-level auditing and runtime-oriented detection. Neural Cleanse and SmoothInv recover potential triggers through backdoor inversion (Wang et al., 2019; Sun & Kolter, 2023), while Activation Clustering, Spectral Signatures, SCAn, SPECTRE, and BEATRIX detect anomalous feature or statistical structures (Chen et al., 2019; Tran et al., 2018; Tang et al., 2021; Hayase et al., 2021; Ma et al., 2023). Recent methods further exploit prediction uncertainty or test-time behaviors for backdoor detection (Guan et al., 2024; Li et al., 2025b).

Runtime-oriented methods detect suspicious inputs from prediction behaviors during inference. STRIP uses prediction entropy under input perturbations (Gao et al., 2019), SCALE-UP analyzes scaled prediction consistency (Guo et al., 2023), BBCaL studies black-box detection from a causality perspective (Hu et al., 2024), and TeCo exploits prediction consistency under image corruptions (Liu et al., 2023). Recent test-time detection has also been extended to object detection under black-box access (Zhang et al., 2025). These methods motivate reconsidering the reliability of inference-time behavioral signals when a frozen static backdoor is activated through diverse post-training trigger configurations.

3 Problem Setting

3.1 Threat Model

We consider a cost-constrained post-training exploitation setting in which an attacker has already implanted a static backdoor into a trained image classifier. The backdoor is established during training using an original trigger τ_0 associated with a target class y_t. After training, the model parameters remain frozen, and the attacker performs no additional poisoning-based training, retraining, fine-tuning, or parameter updates.

Before deployment, the attacker has white-box access to the backdoored model, knows the original trigger τ_0 and target class y_t, and may retain a limited labeled clean reference set from the task distribution together with an identical frozen local copy of the model. Access to the complete original training set is not required. This local copy is used only to analyze the responses of alternative triggers and determine sample-specific trigger configurations; the deployed model itself remains unchanged.

We assume that the defender performs online backdoor detection during inference. In the black-box runtime setting, the defender observes only the model's input–output behavior and does not access model parameters, intermediate representations, training data, or the original poisoning procedure. SBDT does not require access to the detector's scores, thresholds, predictions, or internal information. The attacker's objective is to preserve effective target activation while enabling the already-implanted static backdoor to be triggered in a more diversified manner after training, without incurring additional model-training cost.

3.2 Problem Definition

Given a statically backdoored model f_θ, where θ denotes the frozen model parameters, together with its original trigger τ_0 and target class y_t, we study the problem of *post-training dynamic triggering*: whether a backdoor implanted with a fixed trigger can be activated in a diversified manner after training, without modifying the model parameters or learning a new backdoor association.

Conventional static triggering repeatedly applies the same trigger τ_0 across attack samples. In contrast, we consider selecting different trigger configurations from an admissible configuration space Ω. For an attack sample x_i, the resulting triggered input is written as

$$x_i^{\text{trig}} = x_i \oplus \tau_i, \qquad \tau_i = \mathcal{R}(\tau_0; \lambda_i), \quad \lambda_i \in \Omega, \tag{1}$$

where $\oplus$ denotes trigger injection, $\mathcal{R}(\cdot)$ denotes a trigger configuration transformation, λ_i specifies the configuration applied to the original trigger τ_0, Ω is the admissible configuration space, and τ_i is the resulting post-training trigger.

Post-training dynamic triggering should preserve the target activation of the existing backdoor while keeping the model parameters unchanged. For an effective trigger configuration, this requires

$$f_\theta(x_i \oplus \tau_i) = y_t. \tag{2}$$

Therefore, our focus is not to implant or train a new dynamic backdoor, but to study whether an already-implanted static backdoor retains sufficient post-training trigger flexibility to be activated through different trigger configurations while preserving its original backdoor association.

4 METHOD

4.1 OVERVIEW

We propose **SBDT** (**S**tatic-**B**ackdoor **D**ynamic **T**riggering), a post-training dynamic triggering framework for already-implanted static backdoors. Given a frozen backdoored model and a small clean reference set, SBDT exploits the trigger-response flexibility retained after training to activate the same implanted backdoor through diverse trigger configurations, without retraining, fine-tuning, or modifying model parameters.

Figure 2 illustrates the overall workflow. SBDT first constructs alternative trigger candidates from a constrained configuration space and characterizes their target-activation ability and feature responses. It then filters and refines these candidates to obtain a valid trigger pool. Based on the resulting response characteristics, SBDT performs response-aware sample-wise trigger assignment, encouraging different attack samples to activate the same static backdoor through diverse trigger configurations and response routes.

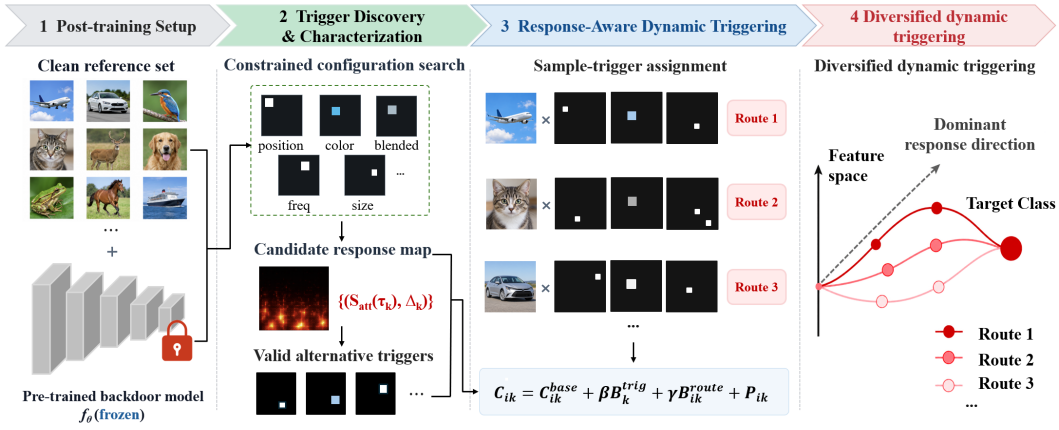


Figure 2: Overview of SBDT for post-training dynamic triggering of static backdoors.

Unlike dynamic backdoors constructed during training, SBDT neither learns a new trigger-generation mechanism nor establishes new trigger–target associations. Instead, it changes only how an already-implanted static backdoor is triggered after training, enabling diversified dynamic triggering while keeping the backdoored model fixed.

4.2 Trigger Discovery and Characterization

SBDT first constructs a candidate trigger pool by applying constrained configuration transformations to the original trigger τ_0, rather than learning new trigger patterns. Specifically, using the transformation $\mathcal{R}$ and configuration space Ω defined in Section 3.2, we define

$$\mathcal{T} = \{\tau_k\}_{k=1}^{K}, \qquad \tau_k = \mathcal{R}(\tau_0; \lambda_k), \quad \lambda_k \in \Omega, \tag{3}$$

where K denotes the number of candidate triggers. Depending on the original backdoor type, Ω may include spatial position, trigger size, color, transparency, blending strength, or frequency location. This process explores alternative trigger configurations beyond the original training configuration that can still activate the implanted static backdoor.

Given the candidate set $\mathcal{T}$, SBDT characterizes each candidate from two complementary aspects. $S_{\text{att}}(\tau_k)$ measures the ability of candidate trigger τ_k to activate the target backdoor, while Δ_k describes its average feature-response displacement over reference samples. Given a small clean reference set $\mathcal{D}_r$ consisting of non-target samples, they are defined as

$$\begin{aligned} S_{\text{att}}(\tau_k) &= \frac{1}{|\mathcal{D}_r|} \sum_{x_i \in \mathcal{D}_r} \mathbf{1}\left[f_\theta(x_i \oplus \tau_k) = y_t\right], \\ \Delta_k &= \frac{1}{|\mathcal{D}_r|} \sum_{x_i \in \mathcal{D}_r} \left[\phi_\theta^\ell(x_i \oplus \tau_k) - \phi_\theta^\ell(x_i)\right]. \end{aligned} \tag{4}$$

Here, $\mathbf{1}[\cdot]$ denotes the indicator function, f_θ denotes the frozen backdoored model, y_t the target class, $\oplus$ trigger injection, and $\phi_\theta^\ell(\cdot)$ the feature representation at layer ℓ.

Finally, SBDT filters the candidate set according to target activation and backdoor-specific diversity constraints, yielding a valid trigger pool $\mathcal{T}_v \subseteq \mathcal{T}$. Since different backdoor types expose different configurable trigger attributes and response structures, the corresponding candidate filtering and diversity-refinement strategies are instantiated accordingly, with detailed construction rules provided in Appendix A. The resulting valid candidates and their response descriptors $\{(S_{\text{att}}(\tau_k), \Delta_k)\}_{\tau_k \in \mathcal{T}_v}$ are used for candidate-level response characterization and, where applicable, diversity-aware trigger-pool refinement.

4.3 Response-Aware Dynamic Triggering

Given $\mathcal{T}_v$, SBDT models each sample–trigger response as $\delta_{ik} = \phi_\theta^\ell(x_i \oplus \tau_k) - \phi_\theta^\ell(x_i)$ and performs response-aware dynamic triggering accordingly. The key idea is to avoid repeatedly using trigger configurations that induce similar internal responses. To this end, SBDT first organizes different trigger-induced responses into diverse response routes and then performs balanced sample-wise trigger assignment, as illustrated in Figure 3.

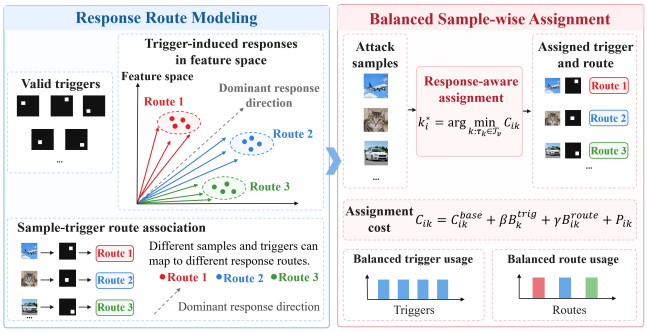


Figure 3: Response-aware dynamic triggering in SBDT.

4.3.1 Response Route Modeling

Although different valid triggers can activate the same implanted backdoor, their feature responses may still exhibit substantial concentration around shared response patterns. Such concentration limits the benefit of trigger diversification, because visually different trigger configurations can still induce similar internal behaviors.

SBDT therefore explicitly models response differences rather than relying only on output-level attack success. Based on the response characteristics of valid triggers, SBDT organizes trigger-induced responses into different response routes so that candidates with distinct internal behaviors can be distinguished during subsequent assignment. Since different backdoor types exhibit different

trigger forms and response structures, the concrete response representation and route construction are instantiated accordingly, while following the same principle of reducing concentration around shared response patterns.

The resulting route association is sample–trigger dependent: different candidate triggers may induce different response routes for the same sample, and the same trigger may also exhibit different response behaviors across samples. The detailed response representation and route construction for different backdoor types are provided in Appendix B.

4.3.2 Balanced Sample-wise Assignment

For each attack sample x_i, SBDT evaluates the valid triggers $\tau_k \in \mathcal{T}_v$ and computes a response-aware base cost C_{ik}^{base}. Building on the candidate-level response characterization in Section 4.2 and the sample–trigger response routes modeled above, this cost measures the suitability of candidate trigger τ_k for sample x_i.

Selecting only the trigger with the lowest base cost may repeatedly favor a small subset of triggers or response routes. SBDT therefore introduces trigger-level and route-level balancing penalties:

$$C_{ik} = C_{ik}^{\text{base}} + \beta B_k^{\text{trig}} + \gamma B_{ik}^{\text{route}} + P_{ik}, \tag{5}$$

where β and γ are the weights for trigger-level and route-level balancing, respectively, B_k^{trig} and B_{ik}^{route} penalize overused triggers and response routes, and P_{ik} discourages candidates that fail to activate the target class.

SBDT then greedily selects

$$k_i^\star = \arg\min_{k:\tau_k \in \mathcal{T}_v} C_{ik}, \qquad \tau_i = \tau_{k_i^\star}, \tag{6}$$

where $k_i^\star$ denotes the selected trigger index for sample x_i. The balancing statistics are updated online after each assignment, encouraging different samples to activate the same implanted static backdoor through diverse trigger configurations and response routes. Detailed cost definitions and the complete assignment procedure are provided in Appendix B.

5 Experiments

5.1 Experimental Setup

Datasets and models. We evaluate SBDT on CIFAR-10 (Krizhevsky, 2009), GTSRB (Stallkamp et al., 2011), and Tiny-ImageNet (Le & Yang, 2015). Unless otherwise specified, we use PreActResNet18 as the target classifier. Each SBDT experiment starts from an already-trained backdoored model and keeps all model parameters frozen throughout post-training triggering.

Backdoor settings and baselines. We apply SBDT to three representative static backdoors: BadNets (Gu et al., 2017), Blended (Chen et al., 2017), and FTrojan (Wang et al., 2022a), using a 10% poisoning rate with target class 0. SBDT performs post-training sample-wise diversified triggering on these frozen backdoors, using eight valid triggers for BadNets and Blended and 23 frequency candidates organized into eight response routes for FTrojan.

For training-stage baselines, LOTUS (Cheng et al., 2024) is a dynamic multi-trigger attack, while BPP (Wang et al., 2022b) and SIBA (Gao et al., 2024) are stealthy backdoor attacks. They follow their original poisoning settings of 10% global, 10%, and 1%, respectively. We also include BadNets-1×1 Retrain and BadNets-1×1 Finetune.

Detectors and metrics. We evaluate **black-box runtime detection** using BBCaL, SCALE-UP, STRIP, and TeCo (Hu et al., 2024; Guo et al., 2023; Gao et al., 2019; Liu et al., 2023), and stronger-access diagnostics including Activation Clustering (AC), Spectral Signatures (SS), SCAn, and BEATRIX (Chen et al., 2019; Tran et al., 2018; Tang et al., 2021; Ma et al., 2023). We further report **model-level auditing and mitigation** using Neural Cleanse (NC), ABL, NAD, and SAU (Wang et al., 2019; Li et al., 2021a;b; Wei et al., 2023). Metrics include Clean Accuracy (ACC), Attack Success Rate (ASR), and, unless otherwise noted, detector True Positive Rate (TPR) at False Positive Rate (FPR) $\leq 5\%$.

5.2 Experimental Results

Main results. Table 1 reports the overall performance of SBDT across different datasets and representative static backdoors. Since SBDT keeps all model parameters frozen, Clean ACC remains identical to that of the original backdoored model in every setting. This confirms that SBDT changes only how the implanted backdoor is triggered after training and does not affect normal classification behavior on clean inputs.

In terms of attack effectiveness, SBDT retains substantial target activation across the evaluated datasets and backdoor types. Although ASR decreases in several settings due to the use of alternative trigger configurations, with the larger reductions being approximately 10–17 percentage points, it remains above 82% in all main settings. These results provide empirical evidence that a statically implanted backdoor can retain exploitable trigger-response flexibility beyond the exact trigger used during training.

For detectability, SBDT substantially lowers TPR for completed detector runs. The reduction is consistently observed across local patch, global blended, and frequency-domain backdoors, indicating that post-training dynamic triggering is not limited to a specific trigger form.

Table 1: Main results of SBDT across representative static backdoors and datasets. Original and SBDT use the same frozen backdoored model and differ only in inference-time triggering. Detection results are reported as TPR at FPR $\leq 5\%$. Following previous work (Li et al., 2025b), OOT denotes that the corresponding detector did not finish within the allocated time limit.

Dataset	Attack	Method	Performance		Detection TPR ↓				
			Clean ACC ↑	ASR ↑	AC	SCAn	SS	BEATRIX	TeCo
CIFAR-10	BadNets	Original	91.84%	93.87%	0.960	0.955	0.950	0.956	0.715
		SBDT	91.84%	83.72%	**0.069**	**0.255**	**0.195**	**0.008**	**0.104**
	Blended	Original	92.85%	99.69%	0.978	0.989	0.817	0.809	0.785
		SBDT	92.85%	83.61%	**0.302**	**0.000**	**0.377**	**0.524**	**0.433**
	FTrojan	Original	93.34%	100.00%	0.999	1.000	0.880	0.205	0.953
		SBDT	93.34%	95.38%	**0.000**	**0.000**	**0.018**	**0.009**	**0.621**
GTSRB	BadNets	Original	98.21%	87.49%	0.424	OOT	0.691	0.964	0.880
		SBDT	98.21%	90.10%	**0.000**	OOT	**0.304**	**0.000**	**0.097**
	Blended	Original	98.57%	99.36%	0.939	0.964	0.768	OOT	0.994
		SBDT	98.57%	84.95%	**0.327**	**0.753**	**0.309**	OOT	**0.489**
	FTrojan	Original	98.42%	100.00%	1.000	1.000	0.768	OOT	0.976
		SBDT	98.42%	88.75%	**0.000**	**0.000**	**0.109**	OOT	**0.652**
Tiny-ImageNet	BadNets	Original	55.93%	98.81%	0.994	0.986	0.522	OOT	0.877
		SBDT	55.93%	82.18%	**0.001**	**0.249**	**0.495**	OOT	**0.413**
	Blended	Original	56.39%	99.57%	0.551	OOT	0.522	OOT	0.495
		SBDT	56.39%	82.11%	**0.011**	OOT	**0.000**	OOT	**0.379**
	FTrojan	Original	55.99%	100.00%	0.520	1.000	OOT	OOT	1.000
		SBDT	55.99%	83.55%	**0.000**	**0.319**	OOT	OOT	**0.538**

Security implication: black-box runtime detection can be insufficient. Table 1 first evaluates broad detectability, including stronger-access diagnostics, to examine whether SBDT reduces feature- and statistical-level regularity across datasets and backdoor types. We then use CIFAR-10 BadNets as a representative case study to evaluate black-box runtime detectors under input–output-only access.

Table 2 reveals two complementary findings. First, SBDT substantially weakens black-box runtime detection in this setting: BBCaL and SCALE-UP are nearly ineffective, while STRIP and TeCo also exhibit clear TPR reductions. Second, model-level auditing and mitigation methods remain effective under their respective access assumptions. These results indicate that black-box runtime detection provides useful deployment-time protection, but should complement, rather than replace, model-level backdoor auditing and mitigation.

Table 2: Black-box runtime detection and model-level defenses under CIFAR-10 BadNets. Left: black-box runtime detection results reported as TPR at FPR $\leq 5\%$. Right: model-level auditing and mitigation results. For ABL, TPR is evaluated under the corresponding triggering configuration.

Black-box runtime detection		
Detector	BadNets	SBDT
BBCaL	0.9968	**0.0000**
SCALE-UP	0.9998	**0.0073**
STRIP	0.9090	**0.6640**
TeCo	0.7150	**0.1040**

Model-level auditing / mitigation		
Method	BadNets	SBDT
NC Index	5.2656	5.2656
ABL TPR	72.02%	91.84%
NAD ASR	93.87%→3.40%	83.72%→5.33%
SAU ASR	93.87%→1.94%	83.72%→2.83%

Stealthiness and additional training cost. For reference, we further compare SBDT with retraining, fine-tuning, and representative training-stage backdoor attacks in terms of attack performance, detectability, and additional training requirements.

Table 3: Comparison of attack performance and backdoor detectability on CIFAR-10. Detection results are reported as TPR at FPR $\leq 5\%$; lower is better.

Method	Performance		Detection TPR ↓			
	Clean ACC ↑	ASR ↑	AC	SCAn	SS	BEATRIX
BadNets-4×4	91.84%	93.87%	0.960	0.955	0.950	0.956
BadNets-1×1 Retrain	89.01%	68.86%	0.157	0.853	0.518	0.545
BadNets-1×1 Finetune	87.63%	90.34%	0.553	0.896	0.730	<u>0.001</u>
LOTUS	94.46%	90.40%	**0.000**	1.000	0.267	0.201
BPP	91.28%	99.89%	**0.000**	<u>0.484</u>	**0.033**	0.513
SIBA	93.68%	97.80%	0.476	0.976	1.000	**0.000**
SBDT	91.84%	83.72%	<u>0.069</u>	**0.255**	<u>0.195</u>	0.008

Table 4: Additional training requirements and stage-specific runtime on CIFAR-10. SBDT runtime refers only to the post-training assignment stage and excludes offline candidate mining and detector evaluation.

Strategy	Method	Training	Weight Update	Schedule	Stage Runtime (min)
Retraining	BadNets-1×1 Retrain	Yes	Yes	100	18.68
Fine-tuning	BadNets-1×1 Finetune	Yes	Yes	20	5.47
	LOTUS	Yes	Yes	300	118.57
Training-stage	BPP	Yes	Yes	100	37.67
	SIBA	Yes	Yes	$(100 + 200 + 100)^*$	156.1
Post-training	**SBDT**	**No**	**No**	**0**	**5.43**

*For SIBA, $100 + 200 + 100$ denotes surrogate training, trigger optimization, and victim-model training, respectively.

As shown in Tables 3 and 4, SBDT achieves low detection TPRs without additional model training or parameter updates. In particular, it reduces the TPRs of AC, SCAn, SS, and BEATRIX to 0.069, 0.255, 0.195, and 0.008, respectively.

The post-training response-aware dynamic triggering stage of SBDT takes 5.43 minutes. These results demonstrate that dynamic triggering can be introduced into an already-implanted static backdoor without additional model training or modifying the backdoored model.

Feature-space analysis. To examine the internal distribution of triggered samples, we extract and standardize intermediate-layer features under the CIFAR-10 / BadNets setting. Figure 4 provides a qualitative t-SNE visualization. Compared with the original fixed-trigger BadNets, triggered samples under SBDT are distributed across multiple feature regions rather than forming the highly concentrated patterns induced by repeated use of a fixed trigger.

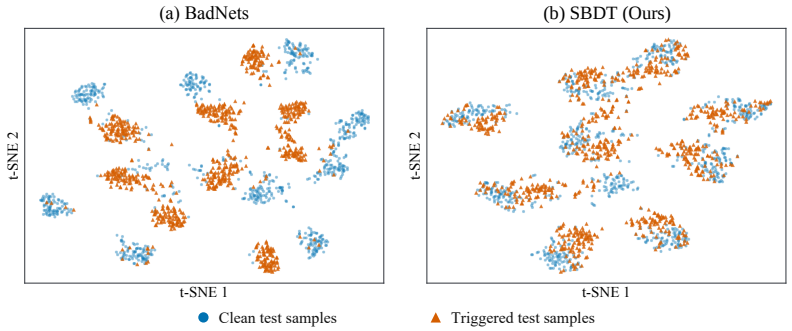


Figure 4: t-SNE visualization of intermediate-layer features under CIFAR-10 BadNets.

We further quantify this effect using feature-distribution diagnostics, including within-feature dispersion, subcluster concentration, and cluster-size entropy. The quantitative results are provided in Appendix F.2. Together, the qualitative and quantitative results indicate that post-training dynamic triggering reduces the feature-level concentration associated with repeatedly using a fixed trigger.

5.3 Ablation Studies

Component ablation. Table 5 evaluates the major components of SBDT. Random assignment preserves the trigger pool while removing response-aware matching; the other variants remove response modeling, route balancing, or trigger balancing. These ablations generally increase feature/statistical detection TPRs, showing that the components provide complementary benefits. Results on Blended and FTrojan are provided in Appendix D.2, with additional sensitivity analyses in Appendix E.

Table 5: Component ablation on CIFAR-10 BadNets. Detection TPR is reported at FPR $\leq 5\%$; lower is better.

Setting	Clean ACC ↑	ASR ↑	AC ↓	SCAn ↓	SS ↓	BEATRIX ↓	TeCo ↓
Full SBDT	91.84%	83.72%	**0.069**	**0.255**	**0.195**	**0.008**	0.104
Random assignment	91.84%	83.54%	0.881	0.329	0.211	0.106	**0.094**
w/o response modeling	91.84%	83.54%	0.130	0.374	0.212	0.117	0.098
w/o route balance	91.84%	83.54%	0.880	0.329	0.209	0.093	0.095
w/o trigger balance	91.84%	83.56%	0.146	0.328	0.211	0.096	0.092

6 Discussion

Our results show that a fixed training-time trigger does not imply a fixed inference-time trigger. By diversifying post-training trigger configurations without modifying model parameters, SBDT can weaken black-box runtime detectors, while model-level auditing and mitigation remain effective under stronger access assumptions. Runtime detection should therefore account for post-training trigger diversity and complement model-level backdoor auditing and mitigation. More broadly, these results suggest that backdoor implantation and inference-time triggering behavior should be treated as related but distinct properties of a compromised model. The effectiveness of post-training diversification nevertheless varies across backdoor mechanisms and detectors. In some settings, it is also accompanied by a moderate reduction in attack success.

7 Conclusion

We presented SBDT, a post-training dynamic triggering framework that discovers alternative triggers and performs response-aware sample-wise assignment without additional model training or parameter updates. Across the evaluated datasets, backdoor types, and model architectures, SBDT retains effective target activation while generally reducing backdoor detectability. These results identify post-training trigger-response flexibility as an important property of static backdoors.

AI USE STATEMENT

Generative AI tools were used to assist with English language editing, LaTeX formatting, and figure/table presentation. They were also used to improve the clarity and readability of manuscript descriptions. All AI-assisted content was carefully reviewed, revised, and verified by the authors. The experimental implementations, results, analyses, and scientific conclusions were independently conducted and validated by the authors. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work studies the post-training trigger-response flexibility of backdoored vision models in controlled research settings. SBDT is evaluated only on standard benchmarks and publicly available backdoor settings to better understand backdoor risks and model security. We do not release malicious models or deployment-oriented attack resources.

REPRODUCIBILITY STATEMENT

We provide experimental settings, datasets, attack configurations, evaluation metrics, and detection protocols in Section 5 and the supplementary materials. We will release the evaluation code, configurations, and scripts necessary to reproduce the reported results, while excluding ready-to-deploy malicious models or deployment-oriented attack resources.

REFERENCES

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

Wenmin Chen, Xiaowei Xu, Xiaodong Wang, Zhipeng Kang, Zewen Li, and Yangming Chen. Dynamic frequency domain trigger backdoor attack with steganography against deep neural networks. *Information Sciences*, 718:122368, 2025a.

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.

Xiong Chen, Mohan Li, Yanbin Sun, and Zhihong Tian. Invisible trigger image: A dynamic neural backdoor attack based on hidden feature. *Neurocomputing*, 639:130296, 2025b.

Siyuan Cheng, Guanhong Tao, Yingqi Liu, Guangyu Shen, Shengwei An, Shiwei Feng, Xiangzhe Xu, Kaiyuan Zhang, Shiqing Ma, and Xiangyu Zhang. LOTUS: Evasive and resilient backdoor attacks through sub-partitioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24798–24809, 2024.

Yansong Gao, Chang Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. STRIP: A defence against trojan attacks on deep neural networks. In *Annual Computer Security Applications Conference*, pp. 113–125, 2019.

Yinghua Gao, Yiming Li, Xueluan Gong, Zhifeng Li, Shu-Tao Xia, and Qian Wang. Backdoor attack with sparse and invisible trigger. *IEEE Transactions on Information Forensics and Security*, 19: 6364–6376, 2024.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Identifying vulnerabilities in the machine learning model supply chain. *CoRR*, abs/1708.06733, 2017.

Jiyang Guan, Jian Liang, and Ran He. Backdoor defense via test-time detecting and repairing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24564–24573, 2024.

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 *International Conference on Learning Representations*, 2023.

Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. SPECTRE: Defending against backdoor attacks using robust statistics. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4129–4139, 2021.

Mengxuan Hu, Zihan Guan, Junfeng Guo, Zhongliang Zhou, Jielu Zhang, and Sheng Li. BBCaL: Black-box backdoor detection under the causality lens. *Transactions on Machine Learning Research*, 2024.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. CS231N Course Project Report, Stanford University, 2015.

Qingyun Li, Wei Chen, Xiaotang Xu, Yiting Zhang, and Lifa Wu. Precision strike: Precise backdoor attack with dynamic trigger. *Computers & Security*, 148:104101, 2025a.

Wei Li, Pin-Yu Chen, Sijia Liu, and Ren Wang. PSBD: Prediction shift uncertainty unlocks backdoor detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10255–10264, 2025b.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In *Advances in Neural Information Processing Systems*, volume 34, pp. 14900–14912, 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16463–16472, 2021c.

Xiaogeng Liu, Minghui Li, Haoyu Wang, Shengshan Hu, Dengpan Ye, Hai Jin, Libing Wu, and Chaowei Xiao. Detecting backdoors during the inference stage based on corruption robustness consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16363–16372, 2023.

Wanlun Ma, Derui Wang, Ruoxi Sun, Minhui Xue, Sheng Wen, and Yang Xiang. The "Beatrix" resurrections: Robust backdoor detection via gram matrices. In *Network and Distributed System Security Symposium (NDSS)*, 2023.

Tuan Anh Nguyen and Anh Tuan Tran. WaNet – imperceptible warping-based backdoor attack. In *International Conference on Learning Representations (ICLR)*, 2021.

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 (IJCNN)*, pp. 1453–1460, 2011.

Mingjie Sun and Zico Kolter. Single image backdoor inversion via robust smoothed classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8113–8122, 2023.

Di Tang, Xiaofeng Wang, Haixu Tang, and Kehuan Zhang. Demon in the variant: Statistical analysis of DNNs for robust backdoor contamination detection. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 1541–1558, 2021.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8011–8021, 2018.

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

Ruotong Wang, Hongrui Chen, Zihao Zhu, Li Liu, and Baoyuan Wu. Versatile backdoor attack with visible, semantic, sample-specific and compatible triggers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3):3680–3697, 2026.

Tong Wang, Yuan Yao, Feng Xu, Shengwei An, Hanghang Tong, and Ting Wang. An invisible black-box backdoor attack through frequency domain. In *European Conference on Computer Vision (ECCV)*, pp. 396–413, 2022a.

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

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

Hangtao Zhang, Yichen Wang, Shihui Yan, Chenyu Zhu, Ziqi Zhou, Linshan Hou, Shengshan Hu, Minghui Li, Yanjun Zhang, and Leo Yu Zhang. Test-time backdoor detection for object detection models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24377–24386, 2025.

Mingyan Zhu, Yiming Li, Junfeng Guo, Tao Wei, Shu-Tao Xia, and Zhan Qin. Towards sample-specific backdoor attack with clean labels via attribute trigger. *IEEE Transactions on Dependable and Secure Computing*, 22(5):4685–4698, 2025.

A Candidate Trigger Discovery Details

This section describes how the candidate trigger pool is instantiated for different static backdoors. SBDT does not learn new trigger patterns or update model parameters. Instead, it explores constrained trigger configurations derived from the original trigger, filters candidates by target activation, and retains diverse candidates for subsequent sample-wise assignment.

BadNets. For CIFAR-10 BadNets, the original model is trained with a 4×4 trigger. During candidate mining, we scan 3×3 white patches over the 32×32 image with a stride of 3 pixels, resulting in 100 spatial anchors. These anchors are further combined with attack-specific trigger configurations before refinement. Candidates are ranked by ASR, and high-response regions are retained. To avoid spatial concentration, we apply non-maximum suppression (NMS) with a minimum Manhattan distance of 6 pixels. The final BadNets pool contains eight triggers covering five coarse spatial groups.

Blended. For Blended, candidates are constructed through local region rearrangement rather than whole-image translation. We use local crop sizes $12, 16, 20, 24, 28$, and 32 with crop origins sampled on a 4-pixel grid, corresponding to maximum single-axis offsets of $20, 16, 12, 8, 4$, and 0 pixels, respectively. We additionally include band crops with widths $6, 8, 10$, and 12 sampled on a 2-pixel grid. The final CIFAR-10 pool contains five local-crop triggers and three band-crop triggers.

FTrojan. For FTrojan, candidates are generated by perturbing frequency locations around the original trigger coordinates rather than sampling the entire frequency plane. In our CIFAR-10 implementation, we perturb base locations such as $(15, 15)$ and $(31, 31)$ in a 32×32 frequency window. Horizontal, vertical, diagonal, and single-frequency perturbations use offsets in $\{-4, -3, -2, -1, 1, 2, 3, 4\}$, while local single-frequency search uses $\Delta x, \Delta y \in \{-3, -2, -1, 0, 1, 2, 3\}$. Coordinates are clipped to $[0, 31]$, with channel combinations and amplitudes also varied. FTrojan retains 23 frequency candidates whose residual responses are organized into eight response routes.

Candidate filtering and refinement. Because different backdoors expose different configurable trigger attributes, SBDT uses attack-specific candidate-mining settings rather than a universal threshold. In all cases, $S_{\mathrm{att}}(\tau_k)$ is first used to remove candidates with insufficient target activation. The remaining candidates are then refined to preserve configuration or response diversity. The thresholds and search ranges in Table 6 are implementation settings rather than additional constraints of the general SBDT framework.

Table 6: Candidate-trigger discovery settings on CIFAR-10.

Attack	Search space	Filtering / refinement	Final pool
BadNets	100 anchors + variants	ASR $\geq 60\%$ + spatial NMS	8 triggers
Blended	1500 local/band crops	ASR $\geq 70\%$ + configuration diversity	8 triggers
FTrojan	23 frequency candidates	ASR $\geq 80\%$ + route-aware refinement	23 candidates / 8 routes

Reference sets. SBDT uses reference samples only for post-training response estimation and trigger assignment. The clean non-target reference set $\mathcal{D}_r$ introduced in Section 4.2 is used to evaluate candidate activation and compute $S_{\mathrm{att}}(\tau_k)$ and Δ_k.

For target-side prototype construction, SBDT uses the target-class subset of the labeled clean reference data. In our experiments, these samples are drawn from the clean training split. In the main CIFAR-10 BadNets setting, up to 1000 target-class samples are used to construct $R = 6$ target-side prototypes at `layer4`, as detailed in Appendix B. These samples are used only to estimate post-training response statistics; no retraining, fine-tuning, or parameter update is performed.

B Sample-wise Assignment Details

This section provides the implementation details of the response-aware sample-wise trigger assignment introduced in Section 4.3.2. Given the valid trigger pool $\mathcal{T}_v$, SBDT evaluates each sample–trigger pair and performs a sequential greedy assignment while keeping the backdoored model fixed. We first describe the feature-aware assignment used for BadNets and Blended, followed by the frequency-domain residual-response variant used for FTrojan.

Target-side prototype construction. In the primary feature-aware implementation, response routes are defined using clean target-class features. Let

$$\mathcal{D}_t = \{(x_j, y_j) : y_j = y_t\} \tag{7}$$

denote the clean target-class reference samples. Their intermediate representations are

$$z_j = \phi_\theta^\ell(x_j), \qquad y_j = y_t. \tag{8}$$

When layer ℓ outputs a spatial feature map, global average pooling is applied before clustering. We cluster these features into R target-side prototypes using K-means, where R denotes the number of prototypes:

$$\{\mu_r\}_{r=1}^R = \mathrm{KMeans}\left(\{\phi_\theta^\ell(x_j) : y_j = y_t\}, R\right), \tag{9}$$

where μ_r denotes the r-th target-side prototype. For CIFAR-10 BadNets, $\ell =$ `layer4` and $R = 6$.

Sample-trigger route association. Given the target-side prototypes $\{\mu_r\}_{r=1}^R$, we define the response-route index c_{ik} for each sample–trigger pair (x_i, τ_k) as

$$c_{ik} = \arg\min_{r \in \{1,\ldots,R\}} \left\| \phi_\theta^\ell(x_i \oplus \tau_k) - \mu_r \right\|_2. \tag{10}$$

Thus, c_{ik} depends on both x_i and τ_k, and the same trigger may be associated with different routes for different samples.

Response-aware base cost. For each pair (x_i, τ_k), SBDT computes a base cost using target-prototype proximity, dominant-response projection, and target-class activation margin. For any sample–trigger scalar a_{ik}, we define

$$Z(a_{ik}) = \frac{a_{ik} - \mathrm{mean}_{i,k}(a_{ik})}{\mathrm{std}_{i,k}(a_{ik}) + 10^{-6}}, \tag{11}$$

where the statistics are computed over all sample–trigger pairs.

The target-prototype proximity is

$$D_{ik} = Z\left(\left\|\phi_\theta^\ell(x_i \oplus \tau_k) - \mu_{c_{ik}}\right\|_2\right). \tag{12}$$

The trigger-induced feature displacement is

$$\delta_{ik} = \phi_\theta^\ell(x_i \oplus \tau_k) - \phi_\theta^\ell(x_i). \tag{13}$$

Target-class margin and activation penalty. Let $s_y(\cdot)$ denote the logit for class y. The target-class logit margin is

$$m_{ik} = s_{y_t}(x_i \oplus \tau_k) - \max_{y \neq y_t} s_y(x_i \oplus \tau_k), \tag{14}$$

with cost

$$M_{ik} = Z\left((m_{ik} - m_0)_+\right) - \alpha_m Z(m_{ik}), \tag{15}$$

where $(a)_+ = \max(a, 0)$, $m_0 = 8$ is the margin threshold, and $\alpha_m = 0.03$ controls the preference for larger target-class margins. The first term penalizes excessively large margins beyond m_0, while the second mildly favors larger target-class margins.

To discourage candidates that fail to activate the target class when successful alternatives exist, let $\mathbf{1}[\cdot]$ denote the indicator function and define

$$h_{ik} = \mathbf{1}\left[f_\theta(x_i \oplus \tau_k) = y_t\right]. \tag{16}$$

The corresponding target-activation penalty is

$$P_{ik} = \lambda_{\mathrm{pen}} \mathbf{1}\left[\sum_{j:\tau_j \in \mathcal{T}_v} h_{ij} > 0\right](1 - h_{ik}), \qquad \lambda_{\mathrm{pen}} = 1000, \tag{17}$$

where λ_{pen} is the target-activation penalty weight. Thus, unsuccessful candidates receive a large penalty whenever at least one successful alternative exists.

Assignment-dependent dominant response direction. The initial assignment uses the target-prototype proximity D_{ik}, target-class margin cost M_{ik}, and target-activation penalty P_{ik}. Here, $\lambda_{\mathrm{man}} = 1.0$ and $\lambda_{\mathrm{om}} = 0.05$ weight the prototype-proximity and margin terms, respectively. The initial assignment is

$$k_i^{(0)} = \arg\min_{k:\tau_k \in \mathcal{T}_v} \left(\lambda_{\mathrm{man}} D_{ik} + \lambda_{\mathrm{om}} M_{ik} + P_{ik}\right). \tag{18}$$

The dominant response direction is then estimated from this assignment and iteratively refined during response-aware reassignment. Given the current assignment, let τ_i denote the trigger assigned to x_i. Its feature displacement is

$$e_i = \phi_\theta^\ell(x_i \oplus \tau_i) - \phi_\theta^\ell(x_i). \tag{19}$$

For the n attack samples, we compute

$$\bar{e} = \frac{1}{n}\sum_{i=1}^{n} e_i \tag{20}$$

and construct

$$E_c = \begin{bmatrix} (e_1 - \bar{e})^\top \\ \vdots \\ (e_n - \bar{e})^\top \end{bmatrix}. \tag{21}$$

The dominant response direction u is the first right singular vector of E_c:

$$u = \arg\max_{\|v\|_2=1} \|E_c v\|_2^2. \tag{22}$$

It captures the strongest feature-response concentration under the current assignment and is re-estimated during the outer iterations.

For each candidate pair, SBDT penalizes its projection onto this direction:

$$Q_{ik} = Z\left(\left|\delta_{ik}^\top u\right|\right). \tag{23}$$

A smaller Q_{ik} indicates weaker alignment with the dominant response direction.

Combining these terms, the response-aware base cost is

$$C_{ik}^{\text{base}} = \lambda_{\text{man}} D_{ik} + \lambda_{\text{spec}} Q_{ik} + \lambda_{\text{om}} M_{ik}, \tag{24}$$

where $\lambda_{\text{spec}} = 0.85$ weights the dominant-response projection term.

Balanced online assignment. Minimizing C_{ik}^{base} alone may repeatedly favor a small subset of triggers or response routes. Let $U_k^{(i)}$ denote the number of previously processed samples assigned to trigger τ_k, and let $O_c^{(i)}$ denote the current occupancy of route c. The total assignment cost is

$$C_{ik} = C_{ik}^{\text{base}} + \beta \frac{U_k^{(i)}}{n/K} + \gamma \frac{O_{c_{ik}}^{(i)}}{n/R} + P_{ik}. \tag{25}$$

Here, K and R denote the numbers of candidate triggers and response routes, respectively, so that n/K and n/R are their nominal balanced occupancies. We use the fixed balancing weights

$$\beta = 0.35, \qquad \gamma = 0.25. \tag{26}$$

These empirically chosen values are fixed throughout the original main evaluation; Appendix E.1 further reports sensitivity analysis and a strict held-out selection protocol without using detector outputs.

The final assignment is

$$k_i^\star = \arg\min_{k:\tau_k \in \mathcal{T}_v} C_{ik}, \qquad \tau_i = \tau_{k_i^\star}. \tag{27}$$

where $k_i^\star$ denotes the selected trigger index for sample x_i.

Sample processing and outer iterations. Because the balancing statistics are updated online, SBDT uses a fixed sample order. Samples with fewer valid trigger choices are processed first, with remaining ties resolved using a fixed random seed. After assigning sample x_i, the corresponding occupancy counts are updated as

$$U_{k_i^\star}^{(i+1)} = U_{k_i^\star}^{(i)} + 1, \qquad O_{c_{ik_i^\star}}^{(i+1)} = O_{c_{ik_i^\star}}^{(i)} + 1, \tag{28}$$

with all other occupancy counts unchanged.

The feature-aware assignment is performed for three outer iterations. At each iteration, the trigger-usage and route-occupancy counts are reinitialized and the dominant direction u is re-estimated. The resulting assignments are compared using detector-independent response-projection statistics, preferring lower high-percentile and mean projection magnitudes. No information from any evaluated detector is used in this selection. Under CIFAR-10 BadNets, the second outer iteration is selected. The backdoored model remains frozen throughout.

Frequency-domain residual-response variant. For FTrojan, SBDT uses an explicit residual-response construction before route clustering. Let δ_{ik} denote a candidate response in the FTrojan response bank and $\mathcal{B}$ the set of response-bank entries. The common response direction is

$$d_{\text{com}} = \text{Norm}\left(\frac{1}{|\mathcal{B}|} \sum_{(i,k)\in\mathcal{B}} \text{Norm}(\delta_{ik})\right), \tag{29}$$

where $\mathrm{Norm}(v) = v/\max(\|v\|_2, \epsilon)$ and $\epsilon > 0$ is a small constant for numerical stability.

The common component is removed to obtain the normalized residual response:

$$r_{ik} = \mathrm{Norm}\left(\delta_{ik} - (\delta_{ik}^{\top} d_{\mathrm{com}}) d_{\mathrm{com}}\right), \tag{30}$$

where r_{ik} denotes the normalized residual response after removing the common response component. The resulting residual responses are clustered with K-means to form response routes. In the FTrojan experiments, 23 frequency candidates are retained and organized into eight response routes. This variant differs from the target-side prototype association used for BadNets and Blended, but follows the same principle of reducing concentration along dominant or common response components while preserving target activation.

C Additional Generalization and Sensitivity Experiments

The main experiments evaluate SBDT across multiple datasets and representative static backdoors. We further examine post-training dynamic triggering from three aspects: generalization to an additional backdoor mechanism, generalization across model architectures, and sensitivity to the poisoning rate of the underlying backdoored model. In all experiments, SBDT is applied only after backdoor training, with all model parameters kept frozen.

C.1 Generalization to SIBA

We first evaluate SBDT on an additional backdoor mechanism beyond the primary settings. Specifically, we apply SBDT to a CIFAR-10 model backdoored with SIBA while keeping the underlying model unchanged.

Table 7: Additional evaluation of SBDT on SIBA under CIFAR-10. Detection results are reported as TPR at FPR $\leq 5\%$. Lower TPR indicates stronger stealthiness.

Setting	Clean ACC ↑	ASR ↑	AC ↓	SCAn ↓	SS ↓	BEATRIX ↓	TeCo ↓
SIBA	93.68%	97.80%	0.476	0.976	1.000	0.000	0.982
SIBA + SBDT	93.68%	86.34%	0.000	0.672	0.000	0.000	0.681

As shown in Table 7, SBDT preserves the clean accuracy of the SIBA model at 93.68%. The ASR decreases from 97.80% to 86.34%, while several detection signals are substantially reduced. AC and Spectral Signatures decrease from 0.476 and 1.000 to 0.000, respectively, while SCAn decreases from 0.976 to 0.672 and TeCo from 0.982 to 0.681. BEATRIX remains at 0.000.

These results show that post-training dynamic triggering is not restricted to the three backdoors used in the main evaluation. However, the remaining SCAn and TeCo TPRs also indicate that the stealthiness improvement depends on both the underlying backdoor mechanism and the detector.

C.2 Generalization Across Model Architectures

We next examine whether SBDT depends on a particular model architecture. We apply SBDT to CIFAR-10 BadNets models with different backbones. For each backbone, SBDT is applied only after the backdoored model has been trained, and all model parameters remain frozen.

Table 8 shows that SBDT preserves clean accuracy on both architectures. For ViT-B/16, the ASR decreases from 91.79% to 84.80%, while SCAn decreases from 0.716 to 0.124, Spectral Signatures from 0.095 to 0.000, and BEATRIX from 0.035 to 0.018. TeCo changes only marginally from 0.314 to 0.312.

For PreActResNet34, the ASR decreases from 95.81% to 74.82%. Meanwhile, SCAn decreases from 0.958 to 0.669, Spectral Signatures from 0.950 to 0.285, and TeCo from 0.293 to 0.066, while AC and BEATRIX remain at 0.000.

Overall, these results indicate that SBDT can be applied across different model architectures without modifying their parameters. The retained ASR and detector reductions nevertheless vary across ar-

Table 8: Generalization of SBDT across model architectures under CIFAR-10 BadNets. Detection results are reported as TPR at FPR $\leq 5\%$. Lower TPR indicates stronger stealthiness. "–" denotes that the detector failed to produce a valid result for this architecture.

Architecture	Setting	Clean ACC ↑	ASR ↑	AC ↓	SCAn ↓	SS ↓	BEATRIX ↓	TeCo ↓
ViT-B/16	BadNets	96.64%	91.79%	–	0.716	0.095	0.035	0.314
ViT-B/16	BadNets + SBDT	96.64%	84.80%	–	0.124	0.000	0.018	0.312
PreActResNet34	BadNets	92.27%	95.81%	0.000	0.958	0.950	0.000	0.293
PreActResNet34	BadNets + SBDT	92.27%	74.82%	0.000	0.669	0.285	0.000	0.066

chitectures and detectors, so we interpret these results as evidence of cross-architecture applicability rather than architecture-independent detection evasion.

C.3 Sensitivity to Poisoning Rate

We further examine SBDT on CIFAR-10 BadNets models trained with poisoning rates of 1% and 5%. In both settings, SBDT is applied only after backdoor training and leaves all model parameters unchanged.

Table 9: Effect of poisoning rate on SBDT under CIFAR-10 BadNets. Detection results are reported as TPR at FPR $\leq 5\%$. Lower TPR indicates stronger stealthiness.

Poison Rate	Setting	Clean ACC ↑	ASR ↑	AC ↓	SCAn ↓	SS ↓	BEATRIX ↓	TeCo ↓
1%	BadNets	93.57%	74.44%	0.592	0.000	0.472	0.056	0.016
1%	BadNets + SBDT	93.57%	74.41%	0.000	0.000	0.000	0.000	0.021
5%	BadNets	92.63%	88.36%	0.598	0.946	0.066	0.001	0.122
5%	BadNets + SBDT	92.63%	89.08%	0.000	1.000	0.000	0.000	0.031

As shown in Table 9, SBDT preserves clean accuracy under both poisoning rates. At a 1% poisoning rate, the ASR remains nearly unchanged at 74.44% versus 74.41%, while AC, Spectral Signatures, and BEATRIX decrease to 0.000. SCAn remains at 0.000, whereas TeCo changes slightly from 0.016 to 0.021.

At a 5% poisoning rate, the ASR changes from 88.36% to 89.08%. AC, Spectral Signatures, BEATRIX, and TeCo decrease from 0.598/0.066/0.001/0.122 to 0.000/0.000/0.000/0.031, respectively. In contrast, SCAn increases from 0.946 to 1.000, showing that the effect of SBDT is not uniform across detectors.

Overall, these results indicate that post-training dynamic triggering remains applicable across different poisoning rates, although its effect on detectability depends on both the underlying backdoor strength and the detector.

D Additional Ablation Studies

D.1 Effect of Trigger Assignment Strategy

To examine whether the effectiveness of SBDT can be explained merely by changing sample–trigger correspondences, we compare the SBDT assignment with several control assignment strategies under the same candidate trigger pool. All variants use the same frozen backdoored model and differ only in how candidate triggers are assigned to attack samples.

The *Base assignment* retains the initial sample–trigger correspondence. *Global reassignment* modifies the assignments of most samples, while *Swap10* and *Swap15* are controlled local reassignment variants that perturb progressively more sample–trigger correspondences. In contrast, the SBDT assignment is produced by the response-aware balanced assignment procedure described in Section 4.3.2. We additionally report the number of assignments that differ from the base assignment; this quantity is an outcome of each strategy rather than a tuned hyperparameter.

Table 10: Effect of trigger assignment strategy on backdoor detectability. Detection results are reported as TPR at FPR $\leq 5\%$. Changed assignments denote the number of samples whose assigned trigger differs from the base assignment.

Setting	Changed Assignments	AC ↓	SCAn ↓	SS ↓	BEATRIX ↓
Base assignment	0	0.122	0.261	0.215	0.118
Global reassignment	4246	0.381	0.475	0.211	0.104
Swap10	500	0.596	0.262	0.216	0.116
Swap15	750	0.374	0.500	0.214	0.114
SBDT assignment	1000	0.069	0.255	0.195	0.008

As shown in Table 10, changing more assignments does not monotonically improve stealthiness. Global reassignment changes the largest number of sample–trigger correspondences, yet produces higher TPRs than the base assignment under AC and SCAn. Similarly, the two local reassignment strategies do not consistently reduce detectability.

In contrast, the SBDT assignment achieves the lowest TPR across all four evaluated feature/statistical diagnostics. These results show that the benefit of SBDT cannot be explained simply by the number of changed assignments. Rather, response-aware sample–trigger matching together with trigger- and route-level balancing determines which trigger is assigned to each sample.

D.2 Component Ablations Across Backdoor Types

We further examine the major components of SBDT on Blended and FTrojan. As shown in Table 11, the results complement the BadNets ablation in the main paper and show that the relative importance of individual components varies across backdoor mechanisms and detectors.

For Blended, Table 11 shows that trigger-level balancing plays an important role in suppressing feature/statistical detection signals. Removing it increases AC from 0.302 to 0.704 and SCAn from 0.000 to 0.949, while random assignment increases AC to 0.654. The effects of response modeling and route-level balancing are more detector-dependent, indicating that no single component uniformly improves every diagnostic.

For FTrojan, Table 11 shows that trigger-level balancing is also particularly important. Removing it increases AC, SCAn, SS, and BEATRIX from 0.000/0.000/0.018/0.009 to 0.208/0.878/0.105/0.031, respectively. The response-modeling component, implemented as explicit residual-response modeling for FTrojan, also contributes to lower TeCo TPR: removing it increases TeCo from 0.621 to 0.690. Full SBDT additionally achieves the highest ASR, 95.38%, among the evaluated FTrojan variants.

Table 11: Additional component ablations on CIFAR-10 Blended and FTrojan. Detection results are reported as TPR at FPR $\leq 5\%$. Lower is better.

Attack	Setting	Clean ACC ↑	ASR ↑	AC ↓	SCAn ↓	SS ↓	BEATRIX ↓	TeCo ↓
Blended	Full SBDT	92.85%	83.61%	0.302	0.000	0.377	0.524	0.433
	Random assignment	92.85%	83.61%	0.654	0.000	0.375	0.527	0.434
	w/o response modeling	92.85%	83.61%	0.227	0.000	0.380	0.534	0.434
	w/o route balance	92.85%	83.61%	0.451	0.000	0.376	0.538	0.410
	w/o trigger balance	92.85%	83.61%	0.704	0.949	0.369	0.525	0.431
FTrojan	Full SBDT	93.34%	95.38%	0.000	0.000	0.018	0.009	0.621
	Random assignment	93.34%	87.79%	0.000	0.000	0.017	0.008	0.652
	w/o response modeling	93.34%	92.38%	0.000	0.000	0.018	0.009	0.690
	w/o route balance	93.34%	91.46%	0.000	0.000	0.018	0.010	0.643
	w/o trigger balance	93.34%	91.60%	0.208	0.878	0.105	0.031	0.664

Overall, together with the BadNets ablation in the main paper, these results show that response-aware assignment, response modeling, and balancing provide complementary benefits across different backdoor mechanisms. Their relative importance, however, depends on the underlying backdoor and detector.

E SENSITIVITY ANALYSIS

E.1 EFFECT OF BALANCING COEFFICIENTS

In the original main experiments, SBDT uses the empirical setting $\beta = 0.35$ and $\gamma = 0.25$, which is kept fixed throughout the main evaluation. We conduct a post-hoc sensitivity analysis around this setting to examine the effect of the balancing coefficients.

To further verify that SBDT does not rely on detector-guided hyperparameter selection, we additionally conduct a strict held-out selection experiment. The 5,000 attack samples are equally split into validation and held-out subsets. The validation subset is used to select β, γ, K, and the number of outer iterations without using any detector output, while the held-out subset is evaluated only after all hyperparameters are frozen.

Table 12: Effect of the balancing coefficients on backdoor detectability under CIFAR-10 BadNets.

Coefficient	Value	AC ↓	SCAn ↓	SS ↓	BEATRIX ↓	TeCo ↓
β *sweep with* $\gamma = 0.25$						
β	0.15	0.147	0.326	0.210	0.098	0.095
β	0.25	0.128	0.327	0.210	0.099	**0.086**
β	0.35 (fixed main setting)	**0.069**	**0.255**	**0.195**	**0.008**	0.104
β	0.45	0.378	0.322	0.209	0.100	0.099
β	0.55	0.564	0.317	0.209	0.100	0.090
γ *sweep with* $\beta = 0.35$						
γ	0.05	0.129	0.323	0.209	0.099	0.098
γ	0.15	0.148	0.323	0.209	0.099	**0.093**
γ	0.25 (fixed main setting)	**0.069**	**0.255**	**0.195**	**0.008**	0.104
γ	0.35	0.877	0.325	0.209	0.099	0.099
γ	0.45	0.127	0.328	0.210	0.100	0.095

Table 12 shows that detection performance is sensitive to the balancing coefficients and that the effect varies across detectors. The fixed main setting $(\beta, \gamma) = (0.35, 0.25)$ achieves the lowest TPRs for AC, SCAn, SS, and BEATRIX among the evaluated values, whereas other coefficient values perform better for TeCo. These results support the role of the balancing mechanism while showing that the main setting should not be interpreted as globally optimal.

To verify that SBDT does not rely on detector-guided hyperparameter selection, the 5,000 attack samples are equally split into validation and held-out subsets. The validation subset selects $\beta = 0.70$, $\gamma = 0.50$, $K = 8$, and two outer iterations without detector outputs; this configuration is then fixed and applied unchanged to the held-out subset.

E.2 EFFECT OF THE NUMBER OF CANDIDATE TRIGGERS

We examine the sensitivity of SBDT to the number of valid candidate triggers K under the CIFAR-10 BadNets setting, while keeping all other experimental settings unchanged.

Table 13: Effect of the number of candidate triggers under CIFAR-10 BadNets. Detection results are reported as TPR at FPR $\leq 5\%$. Lower TPR indicates stronger stealthiness.

K	Clean ACC ↑	ASR ↑	AC ↓	SCAn ↓	SS ↓	BEATRIX ↓
3	91.84%	91.06%	0.577	0.789	0.313	0.027
5	91.84%	89.32%	0.770	0.785	0.313	**0.005**
8	91.84%	83.72%	**0.069**	**0.255**	0.195	0.008
10	91.84%	91.84%	0.134	0.401	**0.133**	0.027

As shown in Table 13, increasing the number of candidate triggers does not monotonically reduce detection TPR. A small candidate pool can limit the flexibility of sample-wise assignment, whereas simply adding more candidates does not necessarily provide more diverse or suitable trigger responses. The main setting $K = 8$ achieves low TPRs across several feature/statistical diagnostics. These results indicate that candidate quality and response diversity are more important than candidate quantity alone.

F Additional Feature-space Analyses

F.1 Layer-wise Trigger-Response Convergence

To better understand why alternative trigger configurations can activate a statically implanted backdoor, we analyze trigger-induced feature responses under CIFAR-10 BadNets. For layer ℓ, we define the response induced by trigger τ_k as

$$\delta_{i,k}^{(\ell)} = \phi_\theta^\ell(x_i \oplus \tau_k) - \phi_\theta^\ell(x_i), \tag{31}$$

and compare it with the response induced by the original trigger τ_0 using cosine similarity.

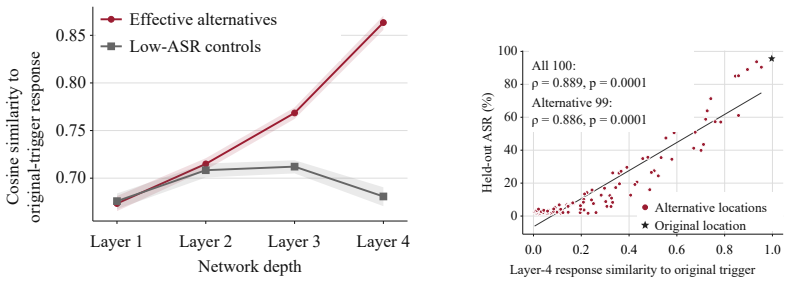


Figure 5: Trigger-response analysis under CIFAR-10 BadNets. Left: layer-wise response similarity to the original trigger. Right: relationship between Layer-4 response similarity and held-out ASR across spatial trigger locations.

Low-ASR controls are constructed by first retaining candidates with historical ASR $\leq 70\%$ and then selecting eight controls via one-to-one matching to the effective alternatives. As shown in the left panel of Figure 5, effective alternative triggers become increasingly aligned with the original-trigger response as network depth increases, whereas low-ASR controls do not exhibit the same trend. This suggests that effective alternative triggers need not resemble the original trigger at shallow layers, but can converge toward similar backdoor responses in deeper feature space.

The right panel of Figure 5 further evaluates all 100 spatial trigger locations. Layer-4 response similarity is strongly associated with held-out ASR by Spearman correlation ($\rho = 0.889$, $p = 0.0001$; 10,000 two-sided permutations). The correlation remains nearly unchanged after excluding the original trigger location ($\rho = 0.886$, $p = 0.0001$).

F.2 Feature-space Distribution Diagnostics

We provide additional details for the feature-space diagnostics used in Section 5.2. All diagnostics are computed under the CIFAR-10 / BadNets setting. Intermediate-layer features are extracted from triggered test samples and standardized before evaluation.

Let $\{z_i\}_{i=1}^N$ denote the standardized triggered features of N samples, with centroid $\bar{z} = \frac{1}{N}\sum_{i=1}^N z_i$. We define within-feature dispersion as

$$D_{\text{feat}} = \frac{1}{N}\sum_{i=1}^{N} \|z_i - \bar{z}\|_2. \tag{32}$$

A larger D_{feat} indicates weaker concentration around a single feature centroid.

To characterize subcluster concentration, we apply K-Means with $J = 8$ clusters to the triggered features. Let n_j denote the number of samples in subcluster j and $p_j = n_j/N$. We compute the largest-subcluster ratio and cluster-size entropy as

$$R_{\max} = \frac{\max_j n_j}{N}, \qquad H_{\text{cluster}} = -\frac{1}{\log J} \sum_{j=1}^{J} p_j \log p_j. \tag{33}$$

Lower $R_{\max}$ indicates weaker subcluster dominance, while higher H_{cluster} indicates a more balanced subcluster distribution.

Table 14: Detailed feature-space diagnostics under the CIFAR-10 / BadNets setting. Higher dispersion and entropy indicate more distributed triggered features, while a lower concentration ratio indicates weaker subcluster dominance.

Metric	BadNets	SBDT
Within-feature dispersion D_{feat}	21.479	**22.640**
Largest-subcluster ratio $R_{\max}$	0.222	**0.167**
Cluster-size entropy H_{cluster}	0.964	**0.979**

As shown in Table 14, SBDT increases within-feature dispersion from 21.479 to 22.640 and cluster-size entropy from 0.964 to 0.979, while reducing the largest-subcluster ratio from 0.222 to 0.167. The largest K-Means subcluster correspondingly decreases from 222 samples under BadNets to 167 samples under SBDT. These results support the qualitative visualization in the main paper and indicate weaker feature concentration under post-training dynamic triggering.

G Implications for Black-box Runtime Detection

We further discuss the complementary roles of black-box runtime detection and model-level backdoor auditing. We use CIFAR-10 BadNets as a representative case study because it provides a canonical fixed-trigger setting for comparing multiple black-box runtime detectors.

The main distinction lies in their access assumptions, as summarized in Table 15. Black-box runtime detectors rely only on input–output queries to the deployed model, whereas model-level auditing and mitigation methods assume stronger access to the model or defense-time optimization procedures.

Table 15: Access assumptions of different evaluation settings.

Setting	Methods	Access assumption
Black-box runtime detection	BBCaL, SCALE-UP, STRIP, TeCo	Input-output queries only
Model-level auditing	Neural Cleanse, ABL	Stronger model or defense-time access
Model-level mitigation	NAD, SAU	Defense-time model modification or cleaning

Under this setting, SBDT substantially reduces the TPR of black-box runtime detectors. BBCaL and SCALE-UP decrease from 0.9968 and 0.9998 to 0.0000 and 0.0073, respectively, while STRIP and TeCo decrease from 0.9090 and 0.7150 to 0.6640 and 0.1040. These results show that post-training dynamic triggering can weaken detection signals derived only from input–output behavior.

However, SBDT does not make the underlying backdoor fundamentally undetectable or unremovable. Because SBDT leaves the model parameters unchanged, stronger model-level auditing and mitigation remain applicable. Neural Cleanse reports the same anomaly index of 5.2656 for BadNets and SBDT, while ABL reports a TPR of 91.84% for SBDT. NAD and SAU further reduce the SBDT ASR from 83.72% to 5.33% and 2.83%, respectively.

These results clarify the security implication of SBDT: black-box runtime detection alone can be insufficient when an implanted static backdoor is triggered through diversified post-training configurations. Runtime detection should therefore complement, rather than replace, model-level backdoor auditing and mitigation.