USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Unveiling the Pitfalls of Data-Free Backdoor Detection Against Pre-Trained Models

Quan Zhao, Linkang Du, Yuntao Wang, and Zhou Su, *Xi'an Jiaotong University;*
Zheng Li, *Shandong University;* Xiangshan Gao, *Zhejiang University;*
Yang Zhang, *CISPA Helmholtz Center for Information Security*

https://www.usenix.org/conference/usenixsecurity26/presentation/zhao-quan

This paper is included in the Proceedings of the
35th USENIX Security Symposium.

August 12–14, 2026 • Baltimore, MD, USA

ISBN 978-1-939133-58-8

Open access to the Proceedings of the
35th USENIX Security Symposium
is sponsored by

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Unveiling the Pitfalls of Data-Free Backdoor Detection Against Pre-Trained Models

Quan Zhao[1] Linkang Du[1*] Yuntao Wang[1] Zhou Su[1*] Zheng Li[2] Xiangshan Gao[3] Yang Zhang[4]

[1]*Xi'an Jiaotong University* [2]*Shandong University* [3]*Zhejiang University*
[4]*CISPA Helmholtz Center for Information Security*

Abstract

Backdoor attacks pose a significant threat to deep learning models, enabling adversaries to manipulate the output through hidden triggers. Recent detection methods aim to identify backdoors without relying on clean samples or assumptions about attacks. Although they report strong performance, these methods are rarely evaluated on pre-trained models. In this paper, we present the first large-scale study of data-free backdoor detection on pre-trained models. Our benchmark includes more than 30,000 models and covers common backdoor attacks. We find that existing data-free methods fail on most pre-trained models, leading to a false sense of security. Despite our effective improvements, serious vulnerabilities remain.

To address this, we propose using convergence speed as a new side-channel signal for backdoor detection. Using this signal, we reveal the cause of the remaining vulnerabilities and build a novel data-free detector that achieves state-of-the-art performance against existing methods. We further analyze how backdoor attacks evade detection and outline unresolved issues. Our results indicate that detecting backdoor attacks requires further exploration. We hope that our work can draw attention to the vulnerabilities in backdoor detection mechanisms for machine learning systems.

1 Introduction

Deep neural networks (DNNs) have demonstrated remarkable performance in a wide range of real-world applications, including image classification [25], speech recognition [43], and natural language processing [21]. Extensive research has validated that DNNs are vulnerable to backdoor attacks [8, 18, 38]. Typically, an adversary can incorporate carefully crafted *triggers* into training samples to inject backdoors into DNN models. A backdoored model produces the expected adversarial output as the *target class* on samples with triggers, which are from the *source class*. For example, in self-driving cars, a backdoored model may misclassify a no-passing sign as a speed limit sign, leading to serious traffic accidents [8].

To mitigate this threat, prior works [7, 16, 28, 33, 34] have proposed a variety of backdoor detection methods. A common strategy is to identify backdoors by reversing a potential trigger from clean samples. However, these methods [16, 29, 33] are significantly limited because they require prior assumptions about the trigger's type and size [7, 34]. Training a meta-classifier [40] offers an alternative, but its scalability is limited by the prohibitive cost of curating and retraining a large fleet of shadow models to span the attack space.

Recent advances in data-free detection [7, 34, 35, 45] attempt to circumvent these challenges by generating proxy input via gradient-based *latent representation* construction. They capture the anomaly of the model output for backdoor detection after forward propagating the latent representation, *e.g.*, the higher posterior output of the target class [7]. Under this framework, prior studies report strong robustness and near-perfect performance against complex triggers, which convey a strong sense of security. However, our further investigation reveals that current evaluations remain limited.

In this paper, we reveal that the effectiveness of current detection systems is less optimistic than reported in data-free cases. Our investigation finds that pre-trained models [42] are rarely evaluated in prior work. This is misaligned with real-world deployments, where pre-trained models are widely used to accelerate training convergence [5] and backdoor injection [36]. Based on this observation, we conduct the first large-scale study of data-free detection on pre-trained models. Specifically, we evaluate all data-free methods on more than 30,000 pre-trained models across eleven backdoor attacks. The results show widespread failure of data-free detection on pre-trained models, even for simple patch triggers. This indicates a serious vulnerability in current data-free detection systems against pre-trained models.

Our further analysis shows that data-free detectors identify strong backdoor-like signals in benign pre-trained models. As a result, target classes of backdoor attacks may be masked by patterns already present in the pre-trained model, leading

*Zhou Su and Linkang Du are the corresponding authors.

to detection failure. To address this, we depart from prior choices of input or intermediate layers and select the output layer as the detection layer to identify potential backdoors in pre-trained models. This modification obtains substantial gains, though it remains insufficient to handle the complex attack strategies.

To mitigate the significant vulnerabilities of the data-free detection system, we propose using the convergence speed (CS) of constructing latent representations as a novel side-channel signal against backdoor attacks. We first reveal the reasons why existing methods lack robustness against complex attack strategies based on CS. Then, we propose SPD as a novel data-free detector against backdoor attacks. SPD achieves state-of-the-art detection performance on the same benchmark of pre-trained models, mitigating the serious vulnerabilities of current data-free detection systems. However, our broader evaluation shows that the backdoor threat in pre-trained models remains unresolved. We uncover several patterns that contradict existing work, indicating that further exploration is needed to address backdoor attacks.

In summary, our main contributions are as follows.

- We conduct the first large-scale evaluation of data-free detectors against pre-trained models. Our experimental results reveal severe vulnerabilities of current detection systems in data-free cases.
- We improve existing data-free detectors and obtain significant gains in detecting backdoors on pre-trained models, while providing insights into their failures under complex attack strategies.
- We propose a novel signal for backdoor detection, and design a data-free detector that achieves state-of-the-art performance through a comprehensive evaluation.
- We reveal phenomena in pre-trained models that contradict prior consensus, alerting the community to reexamine backdoor threats in machine learning systems.

2 Background and Related Work

2.1 Backdoor Attack

The adversary introduces backdoors into the model $\hat{f}(\cdot)$ by altering the training set $D = \{(x_i, y_i)\}_{i=1}^{N}$. In particular, the adversary incorporates a unique trigger Δ in training samples x from various classes, referred to as source classes. Subsequently, the labels $\mathcal{S} \subset \mathcal{Y}$ of these poisoned samples $\hat{x}$ are altered to a target class $t \in \mathcal{Y} \setminus \mathcal{S}$. Upon training with the poisoned dataset $\hat{\mathcal{D}} = \{(x_i, y_i)\}_{i=1}^{N-K} \bigcup \{(\hat{x}_j, t)\}_{j=1}^{K}$, the model is injected with a backdoor associated with the target class t. The backdoored model outputs the target class, *i.e.*, $t = \hat{f}(\hat{x})$, for poisoned samples. Otherwise, it functions normally on clean samples, *i.e.*, $y = \hat{f}(x)$.

Existing backdoor attacks can be broadly divided into data poisoning and training control attacks [45]. Data poisoning attacks inject poisoned samples into the training set [3], whereas training control attacks manipulate the training process or optimization directly [37]. Based on source-class coverage [7,31], attacks are either class-agnostic, which poison samples from all non-target classes, or class-specific, which poison only one designated source class.

By trigger design, attacks can also be categorized as pixel-space triggers, which directly modify the input appearance [8], and feature-space triggers, which induce backdoor behavior through semantic or transformation-based patterns [16]. Regarding sample dependence, triggers can be sample-agnostic, using a fixed pattern across poisoned samples [17], or sample-specific, adapting to individual inputs [15,19]. More advanced methods further improve stealthiness through low-frequency perturbations [44], adaptive features [23], or clean-label poisoning that preserves the original labels [32].

2.2 Data-Required Backdoor Detection

A classic line of work detects backdoors by reverse-engineering a candidate trigger from a handful of clean samples and then testing whether the recovered pattern induces targeted misclassification [16, 29, 33]. These methods typically assume small, localized, pixel-space patterns, and their search spaces or objectives constrain the trigger's type and size. For example, K-Arm [29] explicitly places large triggers and filter-style triggers out of scope.

A second line trains a meta-classifier [40] over shadow models and makes a decision with a limited number of queries, but the need to build and manage many shadow models hinders practicality. To mitigate these limitations, recent efforts explore data-free detection that avoids reliance on clean samples and weakens assumptions about trigger form and scale.

2.3 Data-Free Backdoor Detection

Existing data-free methods [7, 34, 35, 45] typically rely on the construction of the latent representations for each class to generate test cases via gradient descent or gradient ascent, where the latent representations refer to the inputs in a specific model layer. Without loss of generality, we give the following definition for constructing latent representations.

Definition 1 (Latent Representation Construction). *Given a sub-model w_l and a label y_i, the construction of a latent representation ϕ for class i can be formalized as an optimization problem and solved by gradient descent as*

$$\phi_i^* = \arg\min_{\phi_i} \mathcal{L}(f(\mathrm{w}_l; \phi_i), y_i),$$

where $\mathbf{w}_l$ denotes the weights from the l-th layer to the output layer, and $f(\mathbf{w}_l; \phi_i)$ denotes the output obtained by feeding ϕ_i through the corresponding sub-model.

After construction, the latent representations for each class, denoted as $\Phi = \{\phi_i\}_{i=1}^{n}$, are then propagated forward to capture anomalies in the model's behavior. Based on the above discussion, the existing solutions are described as follows.

DF-TND [35]. It constructs latent representations at the input layer per class by maximizing neuron activations in the penultimate layer. These representations are optimized for a fixed number of iterations via a proximal-gradient scheme. It then measures each class's logit increase. If any exceeds a preset threshold, the model is flagged as backdoored.

FreeEagle [7]. This method constructs latent representations at the intermediate layer via gradient descent and forward-propagates them to obtain class logits. The class with an abnormally high posterior output in other classes' logits is identified as the target. A quartile-based metric with a fixed threshold is used to measure anomalies.

MMBD [34]. It constructs latent representations at the input layer by gradient ascent. For each class, a maximum margin statistic is derived by maximizing the logit difference between that class and all others from these representations. The distribution of these statistics is then analyzed to detect abnormal margins that indicate the potential target class.

BARBIE [45]. It detects backdoored models by comparing ordinary and isolated latent representations generated via feature inversion. It quantifies their dominance through the relative competition score (RCS) and derives several RCS-based indicators. Thresholds for these indicators are established using benign models, and a model is flagged as backdoored if its indicators exceed the benign range.

2.4 Pre-Trained Models

Pre-trained models are DNN models whose parameters are first learned on a large upstream dataset and then adapted to a downstream task [5]. Instead of training entirely from scratch, one can initialize a model with pre-trained weights and further optimize it on downstream data. This is a common training strategy, as it can accelerate convergence, reduce training cost, and often improve downstream performance [42]. Such initialization is especially helpful when downstream training data or computational resources are limited. For this reason, pre-training is adopted in many practical learning settings. In this paper, we mainly focus on pre-trained models from the official PyTorch [22] repository, trained on ImageNet [4] and then fully fine-tuned on downstream tasks.

Pre-trained models are also relevant in the context of backdoor attacks [27, 36]. This setup may lower the effort required to implant backdoor behaviors during downstream training, thereby increasing the threat posed by backdoor attacks [12]. However, existing detection methods [7, 34, 35, 45] are rarely evaluated under this setting, leaving its potential threat insufficiently understood.

3 Threat Model

In this paper, we study the problem of backdoor detection on pre-trained models and adopt a threat model consistent with prior work [7, 34, 35].

3.1 Adversary Model

Goals. The adversary aims to implant a backdoor in a DNN model such that inputs embedded with specific triggers are misclassified into an attacker-chosen target class, while preserving high accuracy on clean inputs.

Capabilities. The adversary has full control over the training process, including access to the training dataset and the ability to alter any part of the training pipeline. This allows the adversary to manipulate the data, model architecture, or optimization process as needed. The adversary can employ arbitrary attack strategies (*e.g.*, class-agnostic or class-specific) and adopt any trigger patterns.

3.2 Defender Model

Goals. Given a trained DNN model, the defender seeks to efficiently determine whether the model contains a backdoor. If a backdoor is present, the defender further aims to identify the corresponding target class [33].

Capabilities. We assume the defender has white-box access to the trained models, enabling both forward and backward propagation to construct latent representations. However, the defender has no access to clean or poisoned samples and lacks knowledge of potential triggers or attack strategies.

3.3 Application Scenarios

Models released on real-world open platforms, such as Hugging Face [1], often require post-release security auditing, especially in scenarios involving complex attacks, large-scale tasks, or limited data access.

Complex Backdoor Attacks. Modern backdoor attacks are diverse and rapidly evolving, as discussed in Section 2.1. Many data-required methods rely on assumptions about the trigger form and size. These assumptions are difficult to justify in real deployments, as shown in Section 2.2. It is also infeasible to enumerate all possible triggers. Data-free detection avoids such trigger-specific assumptions [34].

Large-Scale Tasks. When the number of classes grows, collecting clean samples becomes challenging. In addition, both sample quantity and sample quality are hard to guarantee, which can reduce detection reliability. Data-free detection remains applicable without extra data collection, which improves deployability in large-scale settings.

Privacy-Limited Data. In privacy-sensitive deployments, clean data may be inaccessible. This includes confidential or

Table 1: Detection performance of data-free methods under class-agnostic backdoor attacks at a 10% poisoning rate via TPR (%) and FPR (%). Boldface indicates the highest TPR and the lowest FPR under the same setting. Results at a poisoning rate of 5% are reported in Table 8. The experimental setup is as described in Section 4.1.

Method	Dataset	Model	BadNet		Blending		Filter		Refool		WaNet		LF		BPP		SSBA		IAB		AdaP		AdaB	
			TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
DF-TND	CIFAR10	RegNetY	**13.7**	4.2	9.3	4.1	9.4	3.3	4.2	3.4	4.1	**3.1**	6.0	3.4	8.7	3.4	16.0	4.4	3.9	3.4	6.3	4.1	4.1	**3.0**
		ResNet18	**37.6**	3.3	6.8	3.0	2.9	2.1	27.1	3.3	**81.0**	3.3	19.6	3.2	**68.1**	3.0	**49.9**	3.2	0.7	**1.1**	19.9	3.0	5.4	**2.7**
	GTSRB10	MobileNet	2.8	**3.6**	1.3	**2.0**	2.2	**3.4**	1.8	**2.1**	1.1	**3.2**	0.5	**1.6**	0.0	**2.4**	3.3	**3.9**	1.2	**2.1**	1.5	**1.3**	0.8	**2.6**
		ResNet18	**35.7**	5.4	6.2	4.9	18.4	5.3	56.0	5.1	**74.4**	4.8	7.7	4.6	72.9	5.1	1.9	5.4	58.2	5.3	0.0	**1.9**	0.0	**1.1**
	ImageNette	GoogLeNet	5.9	**3.7**	1.2	**3.5**	2.7	**3.4**	10.0	4.9	0.3	**3.1**	2.5	**3.4**	0.3	**2.4**	6.8	4.1	0.7	**2.6**	0.0	**1.9**	0.1	**2.6**
		ResNet18	26.4	5.1	7.1	4.7	11.2	4.8	**53.7**	5.0	**73.1**	4.7	6.1	**3.6**	**61.1**	5.1	5.6	4.7	27.4	4.6	**56.9**	4.9	**64.5**	4.7
FreeEagle	CIFAR10	RegNetY	3.9	5.4	3.3	4.9	5.1	5.1	4.7	5.4	15.5	6.3	3.7	4.9	**16.9**	6.3	3.7	5.0	4.9	6.3	48.6	6.4	**34.7**	6.6
		ResNet18	2.7	3.5	6.4	4.5	5.6	4.0	18.0	4.2	18.3	4.7	6.9	4.4	16.6	4.3	6.3	4.4	11.8	4.7	9.3	3.9	9.6	4.8
	GTSRB10	MobileNet	7.1	5.4	7.5	5.3	8.0	5.4	33.3	5.7	66.2	5.4	9.4	5.7	41.8	5.5	4.6	4.8	29.3	5.7	56.9	5.7	13.9	5.2
		ResNet18	3.0	**2.8**	0.8	**3.0**	1.1	**1.8**	2.9	**3.1**	13.8	**4.4**	3.1	**3.3**	3.4	**3.3**	0.1	**2.9**	13.9	**4.8**	31.3	3.1	19.4	2.4
	ImageNette	GoogLeNet	**13.9**	5.5	6.3	5.1	9.7	5.6	**14.4**	5.1	**60.3**	5.7	10.8	5.4	**88.2**	4.6	14.5	5.2	24.1	5.4	**58.5**	5.2	25.8	5.6
		ResNet18	10.0	**3.8**	6.2	**3.4**	1.9	**3.3**	4.9	**3.2**	10.0	**3.7**	5.5	**3.6**	0.6	**1.8**	17.2	**3.7**	2.5	**3.1**	11.9	**3.6**	11.4	**3.6**
MMBD	CIFAR10	RegNetY	7.2	5.1	**80.6**	5.0	**86.1**	5.0	**19.0**	5.3	9.1	4.9	**19.3**	5.3	13.1	5.3	**16.1**	5.1	5.4	4.9	18.6	5.1	12.5	5.5
		ResNet18	2.8	3.3	0.9	**2.1**	0.3	**1.3**	7.1	3.6	2.7	**2.8**	11.6	3.5	0.3	**1.2**	3.9	3.5	1.9	2.3	13.1	**2.7**	1.4	3.4
	GTSRB10	MobileNet	**36.4**	4.4	**30.6**	4.4	**47.7**	4.1	14.3	4.1	8.2	3.9	20.7	3.8	39.3	4.0	7.2	4.1	13.0	4.2	12.3	3.9	7.5	3.9
		ResNet18	28.7	5.3	11.2	5.6	35.8	5.4	39.4	5.3	9.2	5.3	**20.3**	5.4	**91.9**	4.6	12.7	5.3	38.8	5.4	23.0	5.4	24.9	5.4
	ImageNette	GoogLeNet	11.4	4.0	**33.3**	4.1	**10.7**	3.9	**14.4**	**3.7**	34.5	3.9	**13.2**	3.9	32.1	3.9	**16.9**	**4.0**	**54.0**	4.1	42.3	3.4	**39.9**	4.1
		ResNet18	17.5	5.8	13.8	6.5	21.0	6.4	34.7	6.5	7.2	3.9	9.9	6.1	3.9	5.4	23.5	6.2	21.8	6.0	37.5	6.1	27.9	6.0
BARBIE	CIFAR10	RegNetY	5.8	**3.1**	6.1	**3.1**	8.0	**3.1**	9.1	**3.1**	**48.8**	**3.1**	**19.3**	**3.1**	7.6	**3.1**	11.7	**3.1**	**16.2**	**3.1**	**73.2**	**3.1**	20.9	3.1
		ResNet18	20.2	**2.8**	**22.0**	2.8	**23.1**	2.8	**60.1**	**2.8**	69.2	**2.8**	**38.1**	**2.8**	32.3	2.8	16.5	**2.8**	**26.4**	2.8	**61.5**	2.8	**23.9**	2.8
	GTSRB10	MobileNet	21.1	7.8	24.4	7.8	43.4	7.8	**92.5**	7.8	**99.7**	7.8	**48.0**	7.8	**84.0**	7.8	**14.1**	7.8	**91.7**	7.8	**99.3**	7.8	**47.2**	7.8
		ResNet18	15.0	8.8	**22.2**	8.8	**66.9**	8.8	**78.8**	8.8	57.3	8.8	15.5	8.8	86.8	8.8	**15.5**	8.8	**91.9**	8.8	**39.3**	8.8	**25.8**	8.8
	ImageNette	GoogLeNet	10.3	6.8	4.9	6.8	5.6	6.8	6.9	6.8	17.7	6.8	6.1	6.8	16.4	6.8	6.8	6.8	26.9	6.8	19.9	6.8	10.7	6.8
		ResNet18	**39.8**	10.9	**20.3**	10.9	**23.6**	10.9	44.1	10.9	70.8	10.9	**17.1**	10.9	48.3	10.9	**25.9**	10.9	**28.1**	10.9	44.0	10.9	33.3	10.9

regulated scenarios, such as military applications. A detection pipeline should remain applicable without requiring any user data. This makes data-free detection necessary.

4 Evaluation of Existing Solutions

4.1 Evaluation Setup

Attack Settings. We evaluate eleven backdoor attacks [3, 8, 15–17, 19, 20, 23, 37, 44] in class-agnostic settings, where poisoned samples are from all non-target classes. These attacks cover diverse trigger designs and attack mechanisms as discussed in Section 2.1. In particular, adaptive attacks [23] include two variants: patch-based (AdaP) and blending-based (AdaB). We adopt six of them for class-specific evaluation, where only one source class is poisoned [3, 8, 15–17, 44]. Poisoning rates are 5% and 10% for class-agnostic attacks and 5% for class-specific attacks. The poisoning rate denotes the proportion of poisoned samples in the training set. We adopt these settings because lower poisoning rates may weaken attack effectiveness, while our choices remain within the range considered in prior work, such as 10% in IAB [19] and WaNet [20] and 20%/5% in FreeEagle [7] and BARBIE [45].

Datasets and Models. We conduct experiments on three datasets: CIFAR10 [13], GTSRB10 [11] (a ten-class subset of GTSRB [11]), and ImageNette [4]. We choose these three relatively small datasets because every class is evaluated as both a target class and a source class. For datasets with more classes, such exhaustive evaluation becomes difficult. Larger-scale datasets, *e.g.*, full GTSRB, are further evaluated in Section 7.3. We use ResNet18 [10] for all datasets, and for each dataset we also evaluate one additional architecture, *i.e.*, RegNetY [24], MobileNet [26], and GoogLeNet [30], respectively. In each dataset, 200 benign models are trained, and for every attack setting within that dataset, 200 backdoored models are also trained. All models are trained following the setup described in Section 2.4, resulting in a total of 34,800 models across all settings. The average classification accuracy (ACC) across all models is 84.3%, with a standard deviation of 8.9%. The attack success rate (ASR) is 78.4% on average, with a standard deviation of 25.2%.

Evaluation Metric. We use the true positive rate (TPR) and false positive rate (FPR) as evaluation metrics. TPR measures the proportion of backdoored models that are correctly identified as backdoored, while FPR measures the proportion of benign models that are incorrectly classified as backdoored.

Detection Methods. We evaluate all data-free methods described in Section 2.3. Since the thresholds provided by these methods failed under our experimental settings, we calculate a threshold for each setting using 30% benign and backdoored models, aiming for the highest TPR with an FPR no higher than 5%, following the approach in [7, 45]. In particular, for BARBIE [45], we follow the original method and use only 30% benign models for threshold estimation; therefore, its FPR remains the same under the same dataset-model configuration. The remaining 70% of models are reserved as

the test set for our evaluation. For each result, we repeat the calculation 10 times and take the average.

4.2 Results on Class-Agnostic Attacks

Across datasets and architectures, existing data-free detectors rarely achieve reliable performance on pre-trained models under class-agnostic attacks. From Table 1 and Table 8, we have the following observations.

1) High TPRs are the exception rather than the rule. Specifically, DF-TND performs well mainly on WaNet and occasionally on BPP, while its TPRs on others (*e.g.*, LF) remain low. FreeEagle is uniformly weak across datasets and attacks. MMBD is mixed, with modest TPRs on GTSRB10 for BPP and low TPRs on WaNet. BARBIE is the most competitive overall, especially on GTSRB10, with several high TPRs. However, false positives remain significant, with some settings exceeding 10%. It indicates that its higher TPR comes with additional false alarms. These results confirm that existing data-free detectors lack robustness when applied to pre-trained models.

2) Performance of data-free detectors is highly sensitive to both the backbone and the dataset. Under the same attack, changing the backbone (*e.g.*, RegNetY vs. ResNet18) shifts TPRs by large margins, sometimes from moderate detection to near zero. For the dataset, the same method also shows large performance swings, indicating strong dataset sensitivity. Overall, the results show limited stability across architectures and datasets. In addition, when the poisoning rate is reduced to 5% (see Table 8), detection performance drops significantly and remains sensitive to the model backbone, the dataset, and the strength of the attack.

We therefore identify the first limitation of existing data-free detectors as *L1*: limited robustness on pre-trained models, with outcomes heavily dependent on the backbone, the dataset, and the specific attack.

4.3 Results on Class-Specific Attacks

Table 2 shows that existing data-free detectors perform worse on class-specific backdoors than on class-agnostic ones.

1) The TPRs are low in most settings, with a best case around 42.7%, while most results remain below 30% and even drop to 0% in some configurations. Among the methods, BARBIE also has the highest overall TPRs, but it exhibits the highest FPRs in many settings.

2) Performance remains highly sensitive to the backbone and the dataset. Under the same attack configuration, changing the architecture can shift outcomes from slightly higher TPRs to clear failure, and the same holds when varying the dataset. These patterns mirror the instability observed under class-agnostic attacks.

3) The type of attack further affects outcomes. Some settings yield slightly better recall, but many class-specific variants remain challenging across methods, with TPRs that are well below practical requirements.

Table 2: Detection performance of baselines under class-specific attacks via TPR (%) and FPR (%). Boldface indicates the highest TPR and the lowest FPR under the same setting. The experimental setup is in Section 4.1.

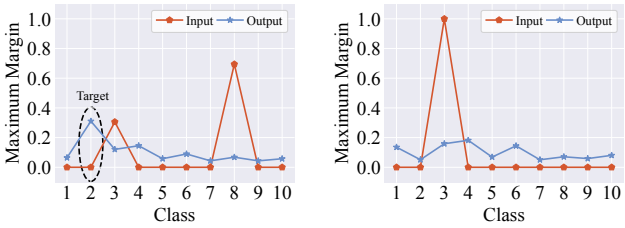
Attack	Dataset	Model	DF-TND		FreeEagle		MMBD		BARBIE	
			TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
BadNet	CIFAR10	RegNetY	**16.1**	4.4	0.6	4.3	9.0	4.0	2.9	**3.1**
		ResNet18	2.3	2.7	**11.8**	4.2	2.6	**2.4**	8.0	2.8
	GTSRB10	MobileNet	1.1	**1.0**	2.7	4.7	**9.5**	3.5	8.7	7.8
		ResNet18	**22.4**	5.4	0.5	**2.7**	13.0	4.7	16.3	8.8
	ImageNette	GoogLeNet	0.8	**2.0**	8.7	5.3	8.7	3.6	**10.6**	6.8
		ResNet18	**21.2**	5.0	13.9	**3.8**	17.5	5.6	18.6	10.9
Blending	CIFAR10	RegNetY	9.8	4.2	2.7	4.5	**28.8**	4.9	6.3	**3.1**
		ResNet18	0.0	**1.1**	9.9	4.5	1.3	2.3	**10.1**	2.8
	GTSRB10	MobileNet	1.4	**0.9**	2.1	3.4	**13.4**	4.0	7.4	7.8
		ResNet18	2.2	4.5	1.5	**3.1**	9.8	5.0	**17.8**	8.8
	ImageNette	GoogLeNet	0.0	**2.4**	5.2	4.9	**25.7**	3.8	7.8	6.8
		ResNet18	1.8	**3.7**	9.3	**3.7**	10.7	5.5	**19.5**	10.9
Filter	CIFAR10	RegNetY	**10.6**	3.9	2.2	3.6	7.8	4.6	2.5	**3.1**
		ResNet18	0.4	**0.8**	**12.9**	4.8	2.1	3.1	6.4	2.8
	GTSRB10	MobileNet	3.4	**3.4**	2.4	3.7	**15.3**	3.9	13.1	7.8
		ResNet18	10.9	5.1	1.4	**3.4**	11.8	4.6	**29.3**	8.8
	ImageNette	GoogLeNet	0.0	**1.9**	7.0	4.2	6.9	3.4	**9.9**	6.8
		ResNet18	1.3	4.0	5.8	**3.5**	18.2	6.2	**20.6**	10.9
Refool	CIFAR10	RegNetY	11.0	4.1	1.6	5.2	**14.0**	5.3	2.6	**3.1**
		ResNet18	5.7	2.8	**37.9**	4.6	1.5	**1.6**	17.6	2.8
	GTSRB10	MobileNet	2.0	3.7	2.1	4.2	1.7	**3.0**	**14.4**	7.8
		ResNet18	15.3	5.2	1.0	**2.3**	16.4	4.9	**42.7**	8.8
	ImageNette	GoogLeNet	0.0	**1.9**	**13.8**	5.1	12.2	3.9	7.9	6.8
		ResNet18	20.2	4.9	9.1	**3.6**	17.6	6.4	**21.5**	10.9
LF	CIFAR10	RegNetY	6.9	4.4	0.9	4.6	**13.4**	4.8	4.6	**3.1**
		ResNet18	0.0	**0.6**	**15.5**	4.4	4.5	2.5	10.7	2.8
	GTSRB10	MobileNet	3.3	**3.7**	1.0	3.9	7.7	3.9	**12.8**	7.8
		ResNet18	2.7	3.7	0.8	**2.9**	8.4	5.1	**11.1**	8.8
	ImageNette	GoogLeNet	0.0	**2.3**	7.4	5.1	**8.3**	3.7	6.2	6.8
		ResNet18	2.5	3.5	4.0	**2.9**	10.2	5.6	**19.1**	10.9
SSBA	CIFAR10	RegNetY	**9.5**	3.6	2.9	5.1	8.3	4.2	2.4	**3.1**
		ResNet18	18.4	2.5	**22.3**	4.7	1.4	**2.3**	8.9	2.8
	GTSRB10	MobileNet	2.8	**3.1**	0.7	4.2	1.9	4.0	**7.0**	7.8
		ResNet18	0.9	**2.6**	1.0	3.5	6.5	4.9	**16.1**	8.8
	ImageNette	GoogLeNet	0.2	**1.9**	7.7	5.2	**19.3**	3.8	8.8	6.8
		ResNet18	1.4	**2.1**	10.3	3.5	11.3	5.3	**15.5**	10.9

Based on the above observation, we identify the second limitation of existing data-free detectors as *L2*: reduced effectiveness against class-specific backdoor attacks compared to class-agnostic attacks.

5 Insights to Address the Limitations

5.1 Insight on *L1*

For *L1*, we find that this limitation stems from pre-trained weights encoding many task-irrelevant or out-of-distribution features from prior datasets. When the model is fine-tuned on the given dataset, it can easily adjust learned features, but it is more challenging to learn features unique to the training samples [2]. Existing data-free methods use the input layer [34, 35] or middle layer [7, 45] as the detection layer.



(a) Backdoored Model (b) Benign Model

Figure 1: Softmax-normalized maximum margins measured by MMBD [34] for each class of benign and backdoored models trained on CIFAR10 [13]. The input and output layers of the pre-trained model are used as detection layers, respectively. The attack is BadNet [8] and its target class is 2.

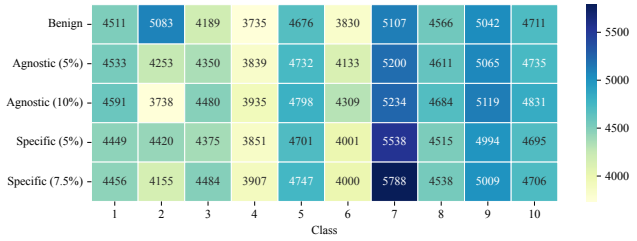


Figure 2: Heatmap of CSs from the output layers of ResNet18 [10] models trained on the CIFAR10 [13] dataset. The comparison includes class-agnostic and class-specific attacks with poisoning rates of 5%, 7.5%, and 10%, where the target class is 2 and the specific source class is 7.

Thus, they are inevitably influenced by pre-trained weights during the detection process. Therefore, an intuitive solution to *L1* is to use the output layer as the detection layer, which is typically not initialized with pre-trained weights due to the difference in the number of classes.

To illustrate the impact of pre-trained weights and the effectiveness of the output layer, we analyze the detection results of MMBD as an example. MMBD identifies backdoored models by a significantly larger maximum margin for the target class compared to other classes. When the input layer serves as the detection layer, the maximum margin produced by MMBD on the backdoored model violates its expectation, as shown in Figure 1(a). Meanwhile, in the benign model shown in Figure 1(b), class 3 attains a substantially larger maximum margin than the other classes, leading to its erroneous identification as the target class. Therefore, MMBD fails in the setting of pre-trained models using the input layer.

However, when the output layer is used for detection, the target class of the pre-trained model and the benign model are correctly identified as shown in Figure 1(a) and Figure 1(b). This suggests that backdoors remain exposed in the output layer. We therefore propose to perform detection on the output layer against *L1* for data-free methods.

5.2 Insight on *L2*

Our insight for *L2* is that existing methods apply the same detection strategy for both class-agnostic and class-specific backdoor attacks, regardless of their varying strength. Here, the difference in strength arises from the number of source classes. A greater number of source classes introduces more features unrelated to the target class, making class-agnostic backdoor attacks more potent than class-specific attacks. Therefore, backdoored models under different attack strategies produce different anomaly metrics when detected by data-free methods. When the same detection strategy is applied to both attack strategies, class-specific backdoor attacks are more likely to be misclassified as benign than class-agnostic ones. To highlight this distinction, we introduce the *convergence speed* as a novel anomaly metric for backdoor detection.

Definition 2 (Convergence Speed, CS). *Given a sub-model* w_l *and a label* y_i*, the convergence speed* cs_i *is defined as the optimization steps required to construct the latent representation to reach the final loss value* L_f*:*

$$cs_i = \arg\min\{T \in \mathbb{N} \mid \mathcal{L}(f(w_l; \phi_i^T), y_i) \leq L_f\},$$

where ϕ_i^T *is the latent representation of class* i *at step* T*.*

CS is a side-channel signal that quantifies the difficulty of constructing the latent representation. Our insight behind CS is that backdoor injection enriches the feature representation of the target class. It leads to a measurable difference: target-class features are more easily searched when constructing latent representations. We then conduct an empirical study using CS to quantify the distinctions among attack strategies.

Empirical Study. [1] We illustrate CSs of five different models as shown in Figure 2. The main observations are as follows.

- **Benign vs. Backdoored.** Backdoored models exhibit a markedly smaller CS for the target class (Class 2). In contrast, CS for source classes performs more slowly relative to benign baselines under both attack strategies. Additionally, increasing the poisoning rate further accelerates the target-class CS across both attack strategies. An opposite trend is observed for the CS of the source class.
- **Class-Agnostic vs. Class-Specific.** At equal poisoning rates, the target-class CS is consistently faster under class-agnostic attacks. This suggests that a greater number of source classes contributes to faster convergence. Meanwhile, class-specific backdoor attacks exhibit a particular pattern that the CS of the unique source class (Class 7) is significantly slower than that of others.

Based on the above observation, the backdoored models exhibit distinct anomaly patterns across poisoning strategies.

[1] More results are provided in Appendix A.

In particular, the target-class anomaly is smaller under class-specific attacks than under class-agnostic attacks. Using a single detection strategy may therefore cause misdetections. Moreover, the slow CS of the source class enables us to distinguish class-specific from class-agnostic backdoor attacks and apply different detection strategies against *L2*.

5.3 Takeaways

We propose using the output layer as the detection layer to mitigate *L1* for pre-trained models, and we illustrate its effect with MMBD [34]. In the following evaluation, we use the superscript $^+$ to mark an improved baseline that adopts the output layer, *e.g.*, MMBD$^+$. In addition, we introduce CS as a new side-channel signal for backdoor detection. CS measures the disparity between class-agnostic and class-specific attacks, allowing differentiation to reduce *L2*.

6 Our Proposal

In this section, we present SPD,[2] a data-free backdoor detector based on CS. Consistent with our discussion in Section 5.1, SPD selects the output layer as the detection layer.

6.1 Calibrated Convergence Speed Collection

As discussed in Section 5.2, CS reveals effective patterns for different attack strategies. However, identifying backdoors using CS remains challenging. As shown in Figure 2, benign models exhibit substantial across-class variation in CS, which obscures the separation between benign and backdoored models. We therefore introduce *calibrated convergence speed* to reduce this variability as follows.

Initialization of Latent Representation. The initial latent representation ϕ_0 determines the starting point for subsequent optimization and directly affects CS. The latent representation ϕ can intuitively be initialized with random or constant values. Random initialization is common [7, 34, 35, 45], yet its randomness reduces the stability of CS and yields fluctuating detection performance. We therefore focus on constant initializations in $[0,1]$. Empirically, larger constants converge faster, whereas $\phi_0 = \mathbf{0}$ may induce vanishing gradients and hinder optimization. Consequently, we set ϕ_0 to an all-ones vector and analyze other initializations of ϕ_0 in Section 7.4.

Optimization of Latent Representation. After initializing ϕ_0, SPD uses the Adam optimizer with the learning rate η to adjust ϕ_i, until the loss value reaches below the final loss L_f. In particular, SPD first uses $\{(\phi_0, y_i)\}_{i=1}^n$ to conduct forward propagation and calculates the minimum loss value $L_s = \min\{\mathcal{L}(f(\mathrm{w}_l;\phi_0),y_i)\}_{i=1}^n$. Specifically, the optimizer is reset with the learning rate of η when the loss value decreases to L_s. We reset the optimizer as a calibration step to avoid overly fast or slow convergence for any class. Based on the above, we present the formulation of the calibrated convergence speed.

Definition 3 (Calibrated Convergence Speed, CCS). *Given a sub-model w_l and a label y_i, the calibrated convergence speed ccs_i is defined as the optimization steps of a two-phase construction of latent representation. Let*

$$L_s = \min_{1\le j\le n} \mathcal{L}(f(\mathrm{w}_l;\phi_0),y_j),$$

where $\phi_0 = \mathbf{1}$. The two-phase convergence is defined as

$$T_1 = \arg\min\{T \in \mathbb{N} \mid \mathcal{L}(f(\mathrm{w}_l;\phi_i^T),y_i) \le L_s\},$$
$$T_2 = \arg\min\{T \in \mathbb{N} \mid \mathcal{L}(f(\mathrm{w}_l;\phi_i^{T_1+T}),y_i) \le L_f\}.$$

At step T_1+1, the optimizer is reset. Then, the calibrated convergence speed for class i is defined as $ccs_i = T_1 + T_2$.

After optimizing the latent representation for each class, we record the set of CCSs.

Remark. The hyperparameters η and L_f influence the CCS, where $ccs_i \propto \frac{1}{\eta}$ and $ccs_i \propto \frac{1}{L_f}$. Insufficient optimization could lead to inaccurate detections, while excessive steps notably prolong execution times. We use $\eta = 0.001$ and $L_f = 0.001$ as default settings and evaluate the impact of different settings on SPD's performance in Section 7.4. In addition, we select Adam as the optimizer for its simpler hyperparameter tuning. For example, choosing the learning rate and the final loss threshold is more involved with SGD.

6.2 Anomaly Metric Calculation

As discussed in Section 5.2, we design two anomaly metrics that align with attack-specific patterns. We quantify anomalies from a statistical deviation perspective. Specifically, we assess departures from a uniform baseline using two simple structured imbalance patterns that reflect attack-specific behavior, and take the stronger evidence as the anomaly score.

Anomaly Metric of Class-Agnostic Attacks. For class-agnostic attacks, constructing the latent representation for the target class tends to converge markedly faster than the others. We identify the potential target class as

$$t = \arg\min_{1\le i\le n} ccs_i.$$

Then we posit a unimodal alternative under which class t carries proportion $\hat{\theta}$ of the total mass and the remaining $n-1$ classes share the remainder equally, *i.e.*,

$$\hat{\theta} = \frac{ccs_t}{\sum_{i=1}^n ccs_i}, \qquad \hat{q} = \frac{1-\hat{\theta}}{n-1}.$$

Against the uniform baseline $1/n$ per class, the log-likelihood ratio is

$$\mathrm{LLR}_{\mathrm{CA}} = ccs_t \log(\hat{\theta} n) + \sum_{i\neq t} ccs_i \log(\hat{q} n). \tag{1}$$

[2] It is derived from "SPeeD", reflecting the core insight of our proposal.

We clip $\hat{\theta}$ and $\hat{q}$ to $(\varepsilon, 1]$ with a small $\varepsilon > 0$ for numerical stability. A larger $\mathrm{LLR}_{\mathrm{CA}}$ indicates a stronger single-class acceleration pattern consistent with class-agnostic behavior.

Anomaly Metric of Class-Specific Attacks. For the class-specific attack, constructing the latent representation of the target class accelerates while that for the unique source class decelerates. Therefore, we focus on the most extreme pair:

$$t = \arg\min_{1\le i\le n} ccs_i, \qquad s = \arg\max_{1\le i\le n} ccs_i.$$

Restricting attention to the pair $\{t, s\}$, let

$$\hat{\theta}_t = \frac{ccs_t}{ccs_t + ccs_s}, \qquad \hat{\theta}_s = 1 - \hat{\theta}_t.$$

We assess deviation from the fair binary baseline $(\frac{1}{2}, \frac{1}{2})$ using the log-likelihood ratio

$$\mathrm{LLR}_{\mathrm{CS}} = ccs_t \log\big(2\hat{\theta}_t\big) + ccs_s \log\big(2\hat{\theta}_s\big). \tag{2}$$

$\hat{\theta}_t$ and $\hat{\theta}_s$ are also clipped to $(\varepsilon, 1]$ with a small $\varepsilon > 0$ for numerical stability. A larger $\mathrm{LLR}_{\mathrm{CS}}$ indicates a stronger fast-slow contrast characteristic of class-specific behavior.

Anomaly Score. We take the strongest statistical evidence against the uniform baseline as the final anomaly score:

$$m \;=\; \max\{0,\, \mathrm{LLR}_{\mathrm{CA}},\, \mathrm{LLR}_{\mathrm{CS}}\}.$$

The nonnegativity enforces a conservative convention where no evidence yields zero. We also report the dominating alternative as the detected pattern:

$$\text{case} \;=\; \begin{cases} \text{Class-Agnostic}, & \mathrm{LLR}_{\mathrm{CA}} \ge \mathrm{LLR}_{\mathrm{CS}}, \\ \text{Class-Specific}, & \mathrm{LLR}_{\mathrm{CA}} < \mathrm{LLR}_{\mathrm{CS}}. \end{cases}$$

A larger m indicates a stronger departure consistent with either a single-class acceleration or a fast-slow extreme pair. In practice, m serves as a unified anomaly metric, while *case* provides a succinct interpretation of the prevailing attack-specific pattern.

6.3 Discussion

Effectiveness of CCS. SPD uses two anomaly metrics for class-agnostic and class-specific backdoor attacks. Both metrics share an implicit precondition: the backdoor perturbs convergence strongly enough that the target class and the source class become the smallest or the largest CCS among all classes, respectively. This relies on the assumption that when the model is trained on a clean dataset, CCS values are approximately uniform across classes, so any acceleration or deceleration produces a salient deviation. This is the reason why we adopt CCS rather than raw CS, and its effectiveness is evaluated in Section 7.5.

Poisoning Rate. The magnitude of CCS shifts is also governed by the poisoning rate. Prior methods are likewise sensitive to the poisoning rate, as reflected between Table 1 and Table 8. When the poisoning rate is low, or when the target class is already slow to converge before poisoning, the target class may fail to attain the minimum CCS. The same applies to the source class and the maximum CCS. These cases violate the implicit assumption used by SPD, causing a potential failure.

The Number of Classes. We infer that the number of classes modulates detectability in opposite ways for the two attack strategies. For class-agnostic attacks, increasing the number of classes tends to help detection because a larger pool of source classes contributes additional features that speed up the target-class convergence even under reduced poisoning. However, detection becomes harder with more classes for class-specific attacks. Achieving the global minimum CCS becomes more difficult for the target class, particularly when its clean CCS[3] is relatively large.

These insights immediately suggest adaptive strategies against SPD, which we analyze in Section 8. However, we note that even when the target class does not achieve the minimum CCS, SPD often still flags the model as backdoored because at least one class exhibits a CCS that is markedly more extreme than the remainder, albeit with potentially incorrect target identification.

The Number of Target Classes. Consistent with observations of existing methods, SPD is designed for single-target backdoor attacks. The anomaly metrics evaluate deviations between a single target and the remaining classes. For multi-target backdoors, the single scalar anomaly score is theoretically insufficient. In the extreme case where all classes become targets (*e.g.*, *all-to-all* attacks), the target-specific signal disappears and the anomaly measures of both existing methods and SPD degenerate. Therefore, we also evaluate all-to-all attacks as adaptive attacks in Section 8.

7 Evaluation of Our Proposals

7.1 Results of Class-Agnostic Attacks

In Table 3 and Table 9, we compare the detection performance of SPD with improved baselines under class-agnostic backdoor attacks on the same benchmark as in Section 4.1.

Effectiveness of Our Improvement. [4] At both 10% and 5% poisoning rates, the improved baselines outperform their originals, as shown in Table 1 and Table 8. Specifically, FreeEagle$^+$ delivers modest improvements on average, yet in several configurations it reaches TPRs of 90% to 100%. MMBD$^+$ achieves substantial gains, with strong performance in most settings, although a few configurations still fail. These results

[3] The clean CCS is obtained from models trained on the same dataset without poisoning.

[4] DF-TND and BARBIE are not suitable for our modification.

Table 3: Detection performance of SPD and improved baselines under class-agnostic backdoor attacks at a 10% poisoning rate via TPR (%) and FPR (%). Results at a poisoning rate of 5% are in Table 9. The experimental setup is in Section 4.1.

Method	Dataset	Model	BadNet		Blending		Filter		Refool		WaNet		LF		BPP		SSBA		IAB		AdaP		AdaB	
			TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
FreeEagle$^+$	CIFAR10	RegNetY	31.5	5.6	15.7	5.6	15.3	5.9	8.8	3.7	87.2	4.1	13.3	5.0	22.6	4.0	30.5	4.0	20.1	4.0	90.1	3.4	25.4	4.0
		ResNet18	13.8	5.9	19.8	5.1	12.7	3.9	14.4	4.8	**100.0**	**0.0**	15.3	4.5	51.4	5.3	27.7	5.0	52.0	5.6	99.7	**0.0**	49.7	5.4
	GTSRB10	MobileNet	27.5	4.6	20.5	4.9	26.9	4.6	99.0	2.1	99.0	1.9	20.5	3.9	94.1	4.4	17.8	4.0	95.3	3.7	99.4	1.4	89.3	4.9
		ResNet18	21.1	3.4	16.3	3.5	22.5	3.7	89.4	2.4	90.8	**0.0**	34.9	5.2	91.5	1.4	28.7	5.1	91.9	2.0	**100.0**	0.9	93.0	3.2
	ImageNette	GoogLeNet	19.1	4.6	7.1	4.1	22.8	4.5	11.1	4.0	46.6	5.4	17.0	4.7	71.7	4.6	11.6	4.7	14.5	5.2	49.8	4.9	14.8	5.0
		ResNet18	36.6	3.6	40.8	3.6	32.9	3.6	17.7	2.6	86.1	2.7	49.3	3.2	97.9	2.2	10.7	3.4	43.5	2.7	50.2	2.9	36.8	3.7
MMBD$^+$	CIFAR10	RegNetY	84.2	3.5	84.6	3.4	93.5	3.1	82.2	3.4	77.8	1.8	91.9	2.9	**93.6**	**3.3**	98.3	2.2	84.2	**3.4**	58.4	0.9	99.1	2.1
		ResNet18	91.5	4.6	99.1	0.8	97.0	3.6	**100.0**	**0.0**	**100.0**	**0.0**	99.0	1.3	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**
	GTSRB10	MobileNet	44.4	3.4	68.1	3.3	57.0	3.4	54.9	3.9	27.0	3.4	48.4	3.0	13.4	3.7	34.8	3.6	5.7	2.6	77.6	3.2	40.4	3.3
		ResNet18	19.3	5.6	29.3	5.6	42.8	5.6	99.9	2.8	66.9	4.3	16.1	5.2	99.1	2.7	9.3	4.0	99.1	2.9	99.6	2.4	76.3	5.7
	ImageNette	GoogLeNet	97.6	3.1	97.7	4.1	95.8	4.4	97.2	3.7	47.0	4.6	97.7	3.5	27.1	4.9	**95.8**	4.2	**96.1**	**2.7**	15.7	4.1	91.4	3.8
		ResNet18	94.9	4.6	99.1	2.4	94.4	4.6	98.2	2.9	**100.0**	0.7	96.8	3.9	**100.0**	0.7	93.3	3.6	**98.7**	2.1	**100.0**	0.7	99.4	2.3
SPD	CIFAR10	RegNetY	**99.5**	**1.7**	**99.1**	**0.3**	**99.4**	**0.8**	**99.4**	**0.9**	**99.2**	**0.0**	**99.1**	**0.6**	87.5	4.4	**100.0**	**0.0**	**93.1**	3.6	**100.0**	**0.0**	**100.0**	**0.0**
		ResNet18	**98.8**	**1.2**	**100.0**	**0.3**	**100.0**	**0.3**	**100.0**	0.3	**100.0**	0.3	**100.0**	**0.3**	**100.0**	0.3	**100.0**	0.3	**100.0**	0.3	**100.0**	0.3	**100.0**	0.3
	GTSRB10	MobileNet	**99.8**	**0.0**	**99.4**	**0.9**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**99.4**	**1.1**	**100.0**	**0.0**	**95.7**	**3.1**	**99.6**	**0.2**	**100.0**	**0.0**	**100.0**	**0.0**
		ResNet18	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**98.1**	**2.4**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**
	ImageNette	GoogLeNet	**100.0**	**0.9**	**97.9**	**1.3**	**95.9**	**2.4**	**98.8**	**1.6**	**100.0**	**0.9**	**99.3**	**1.1**	**100.0**	**0.9**	94.4	**2.7**	90.8	3.8	**100.0**	**0.9**	**100.0**	**0.9**
		ResNet18	**100.0**	**0.1**	**100.0**	**0.1**	**100.0**	**0.1**	**100.0**	**0.1**	99.6	**0.6**	**100.0**	**0.1**	**100.0**	**0.1**	**100.0**	**0.1**	98.1	**1.3**	**100.0**	**0.1**	**100.0**	**0.1**

suggest that selecting the output layer rather than the intermediate or input layers reduces the impact of pre-trained weights on detection, consistent with our insights in Section 5.1.

When the poisoning rate drops to 5%, both FreeEagle$^+$ and MMBD$^+$ show a general decline in TPR, confirming that detection remains sensitive to poisoning strength. However, both retain strong results in specific settings. For example, on CIFAR10 with ResNet18, MMBD$^+$ attains near-100% TPR on Refool, WaNet, BPP, SSBA, IAB, AdaP, and AdaB. These attacks span natural triggers, sample-specific triggers, and adaptive variants, indicating that the improvements can be robust under complex trigger types.

In summary, selecting the output layer as the detection layer can address *L1*, which substantially improves the robustness of existing methods on pre-trained models. The improved baselines also achieve markedly higher target-class identification accuracy, compared with their original methods. Nevertheless, they still exhibit significant weaknesses in backdoor detection, with sensitivity to architecture, dataset, attack type, and poisoning rate. For example, on GTSRB10 under the IAB attack, MMBD$^+$ attains a TPR of 99.1% on ResNet18 but only 5.7% on MobileNet. This gap indicates limited cross-architecture robustness of the insight in MMBD and highlights the need for further study of backdoor detection under diverse model designs.

Overall Performance of SPD. SPD achieves state-of-the-art performance against all baselines, including their improved variants, across datasets and architectures. In most configurations, the TPR is near 100% and the FPR is the lowest, which supports our use of CCS as a signal for backdoor detection. SPD also identifies nearly all target classes in backdoored models. When the poisoning rate is reduced to 5%, the performance of SPD also decreases, with the largest drop on ImageNette, indicating sensitivity to poisoning strength similar to FreeEagle$^+$ and MMBD$^+$. However, only a few settings exhibit substantial degradation, and the majority still maintain TPRs above 90%. In summary, SPD and our CCS-based insight show stronger robustness across diverse backdoor types.

Observation of Triggers. Across our evaluations, we find patterns that do not match common expectations for trigger types on pre-trained models. Detection is weakest for simple triggers such as BadNet, Blending, and Filter rather than for complex triggers. This differs from prior reports that describe sample-specific and adaptive triggers as harder to detect. We also find that the local trigger is harder to detect than the global triggers, which contradicts the common insights. These findings suggest that pre-trained weights create a different detection landscape for backdoor defenses.

Based on these findings, we infer that simple triggers can be easily hidden by the complex features introduced by pre-trained weights, as discussed in Section 5.1. They become harder to detect, especially BadNet, which introduces only small visible changes. We hypothesize that detectability on pre-trained models correlates with how outlying the trigger-induced features are relative to pre-trained representations. To support this insight, we conduct a case study in Section 7.6. In summary, we reveal a unique vulnerability of pre-trained models to backdoor attacks that prior work has overlooked.

Observation of Datasets and Models. For datasets, detection performance varies with the data distribution. MMBD$^+$ is consistently weaker on GTSRB10, while FreeEagle$^+$ and SPD are relatively weaker on ImageNette. These patterns suggest that attack evasiveness depends on the dataset.

For model architectures, ResNet18 generally yields stronger detection than other backbones under the same attack and poisoning level. For example, several detection-attack combinations that succeed on ResNet18 show poorer TPR on MobileNet or RegNetY. This gap indicates that model design also plays a key role in the reliability of backdoor detection.

Table 4: Detection performance of SPD and improved baselines under class-specific attacks via TPR (%) and FPR (%). The experimental setup is in Section 4.1.

Attack	Dataset	Model	FreeEagle$^+$		MMBD$^+$		SPD	
			TPR	FPR	TPR	FPR	TPR	FPR
BadNet	CIFAR10	RegNetY	6.2	4.4	10.2	**3.2**	**62.4**	4.8
		ResNet18	15.3	**3.1**	21.9	4.3	**83.4**	4.8
	GTSRB10	MobileNet	13.8	4.9	25.8	**3.9**	**69.6**	**3.9**
		ResNet18	15.1	3.6	4.1	4.0	**94.2**	**1.3**
	ImageNette	GoogLeNet	43.7	4.9	16.8	**3.6**	**89.0**	4.6
		ResNet18	6.5	2.8	23.6	5.2	**100.0**	**0.6**
Blending	CIFAR10	RegNetY	9.5	4.1	11.6	**2.9**	**74.1**	4.9
		ResNet18	8.3	**4.2**	41.9	**4.2**	**93.0**	5.0
	GTSRB10	MobileNet	15.9	4.4	22.0	3.5	**64.5**	**3.4**
		ResNet18	11.5	2.7	5.4	3.9	**91.3**	**1.0**
	ImageNette	GoogLeNet	47.6	5.1	10.6	**3.5**	**84.6**	4.5
		ResNet18	4.7	2.0	38.7	5.8	**100.0**	**0.6**
Filter	CIFAR10	RegNetY	8.7	4.6	9.6	**3.6**	**65.8**	4.7
		ResNet18	16.4	**3.7**	37.5	4.4	**87.2**	4.9
	GTSRB10	MobileNet	14.6	5.9	23.0	3.7	**78.0**	**3.6**
		ResNet18	25.0	4.6	5.4	3.9	**97.4**	**0.7**
	ImageNette	GoogLeNet	59.4	4.9	15.4	**3.5**	**90.4**	4.6
		ResNet18	4.0	2.9	32.8	5.8	**100.0**	**0.6**
Refool	CIFAR10	RegNetY	6.0	3.9	8.3	**3.5**	**68.2**	4.4
		ResNet18	15.8	3.5	44.8	4.2	**96.3**	**3.4**
	GTSRB10	MobileNet	35.2	6.8	50.6	3.6	**93.5**	**2.9**
		ResNet18	19.5	4.2	16.0	5.4	**95.7**	**1.4**
	ImageNette	GoogLeNet	48.2	4.7	10.8	**3.6**	**82.5**	4.2
		ResNet18	4.9	3.5	31.3	5.4	**100.0**	**0.6**
LF	CIFAR10	RegNetY	7.4	**3.1**	15.8	3.4	**69.5**	4.9
		ResNet18	16.9	5.4	37.3	**4.7**	**89.8**	4.9
	GTSRB10	MobileNet	15.3	4.9	19.5	**3.4**	**64.0**	**3.4**
		ResNet18	19.9	3.4	5.4	3.2	**89.0**	**1.9**
	ImageNette	GoogLeNet	60.2	5.1	13.5	**3.7**	**86.3**	4.6
		ResNet18	3.6	3.1	23.7	5.6	**100.0**	**0.6**
SSBA	CIFAR10	RegNetY	12.5	5.0	17.8	**4.0**	**75.8**	4.4
		ResNet18	13.4	4.5	62.4	4.4	**97.1**	**1.1**
	GTSRB10	MobileNet	15.1	5.8	26.3	**3.6**	**51.5**	3.9
		ResNet18	18.9	3.4	3.0	4.6	**93.7**	**2.5**
	ImageNette	GoogLeNet	44.5	5.0	12.6	**3.9**	**89.1**	4.6
		ResNet18	4.2	3.6	28.1	5.2	**100.0**	**0.6**

7.2 Results of Class-Specific Attacks

Performance of Improved Baselines. As shown in Table 4, the improved baselines remain largely ineffective on class-specific backdoor attacks. Although they outperform their original versions, TPRs are still low in most settings, often below 30% on CIFAR10 and GTSRB10, with only occasional mid-range results on ImageNette. These results indicate that our output-layer modification mitigates the impact of pre-trained weights but remains insufficient to make existing methods robust for class-specific backdoor attacks. Even at poisoning rates comparable to the class-agnostic experiments, the gap in detection accuracy is large, underscoring the inherent difficulty of class-specific attacks. In summary, our evaluation reveals substantial shortcomings of existing methods on class-specific backdoor attacks and highlights the need for more effective detectors.

Performance of SPD. Consistent with the class-agnostic setting, SPD achieves state-of-the-art performance on class-specific backdoors and surpasses all baselines. This supports our insight that CCS is an effective signal for detecting class-specific attacks on pre-trained models. Meanwhile, adopting distinct detection strategies for class-specific backdoor attacks is effective. We further evaluate the efficacy of SPD's design in Section 7.5. However, the results also reveal sensitivity to model architecture. Detection on ResNet18 is generally stronger than on other backbones. Therefore, meaningful opportunities for improvement remain.

7.3 More Evaluation Results

Performance on Large Datasets. We evaluate on CIFAR100 [13] and TinyImageNet [14] under class-agnostic attacks using ResNet34 [10] for both datasets. For CIFAR100, we use poisoning rates of 5% and 10%. For TinyImageNet, we use a poisoning rate of 5% and additionally include ResNet50 [10] as a backbone. Each setting contains 100 models. The average ACC across all models is 77.2%, with a standard deviation of 4.8%. The ASR is 87.3% on average, with a standard deviation of 20.2%.

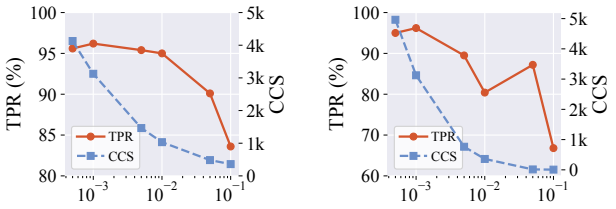
As expected from our analysis in Section 6.3, a larger number of classes makes class-agnostic attacks easier to detect, as demonstrated in Table 5. MMBD$^+$ and SPD achieve near-perfect TPRs in almost all settings with very low FPRs, regardless of poisoning rate or backbone. FreeEagle$^+$ shows pronounced variability: some attacks reach 100% TPR while others drop to around 0.3%. These results reinforce that increasing the number of classes amplifies detectability and that SPD maintains strong robustness on large datasets. In particular, class-specific backdoors on large datasets are evaluated in adaptive attacks (see Section 8).

Performance on Vision Transformer. We evaluate SPD and the improved baselines on ViT [6] using CIFAR10 [13] with the same settings as Section 4.1. The average ACC across all models is 96.8%, with a standard deviation of 1.6%. The ASR is 93.2% on average, with a standard deviation of 10.6%. FreeEagle$^+$ and MMBD$^+$ both yield TPRs below 10%, indicating failure on pre-trained ViT. In contrast, SPD attains TPRs between 71.4% and 93.6%. We also observe higher FPRs on ViT than on convolutional backbones: SPD averages 6.1% and FreeEagle$^+$ peaks at 10.7%. Overall, SPD delivers better performance on ViT.

To further evaluate scalability to larger-scale models and datasets, we conduct experiments on ImageNet [4] using ViT. We fine-tune 100 models with 50 samples per class and evaluate BadNet, Blending, and Filter under the class-agnostic setting. For each attack, only one sample from each source class is poisoned, corresponding to a poisoning rate of 1.998% in the fine-tuning set and about 0.078% in the full ImageNet training set. Backdoored models achieve an average ACC of 75.60% and an average ASR of 94.93%, with standard deviations of 5.8% and 11.1%, respectively. SPD achieves 100%

Table 5: Detection performance of SPD and improved baselines under class-agnostic backdoor attacks on large datasets via TPR (%) and FPR (%). Boldface indicates the highest TPR and the lowest FPR under the same setting.

Method	Dataset	Model	Poisoning Rate	BadNet		Blending		Filter		Refool		WaNet		LF		BPP		SSBA		IAB	
				TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
FreeEagle$^+$	CIFAR100	ResNet34	10%	4.1	6.1	7.9	6.0	19.7	5.7	13.0	6.1	**100.0**	**0.9**	71.0	**0.9**	**100.0**	**0.9**	76.1	5.6	**100.0**	**0.9**
		ResNet34	5%	0.3	3.4	0.3	3.9	15.4	5.6	12.9	5.0	**100.0**	**0.9**	54.0	5.4	**100.0**	**0.9**	54.0	5.4	**100.0**	**0.9**
	TinyImageNet	ResNet34	5%	1.1	7.7	1.7	7.7	2.6	7.1	15.7	6.9	**100.0**	1.1	1.4	8.0	**100.0**	1.1	7.4	7.7	71.2	6.0
		ResNet50	5%	4.9	6.9	24.2	7.1	5.0	6.0	25.4	6.6	97.0	2.6	14.0	7.0	99.9	**0.4**	47.1	5.4	81.3	3.7
MMBD$^+$	CIFAR100	ResNet34	10%	**100.0**	2.0	**100.0**	2.0	**100.0**	2.0	**100.0**	2.0	**100.0**	2.0	**100.0**	2.0	**100.0**	2.0	**100.0**	2.0	**100.0**	2.0
		ResNet34	5%	98.9	3.1	99.1	2.0	**100.0**	2.0	**100.0**	2.0	**100.0**	2.0	**100.0**	2.0	**100.0**	2.0	**100.0**	2.0	**100.0**	2.0
	TinyImageNet	ResNet34	5%	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**
		ResNet50	5%	**100.0**	3.6	**100.0**	3.6	**100.0**	3.6	**100.0**	3.6	**100.0**	3.6	**100.0**	3.6	**100.0**	3.6	**100.0**	3.6	**100.0**	3.6
SPD	CIFAR100	ResNet34	10%	**100.0**	**1.0**	**100.0**	**1.0**	**100.0**	**1.0**	**100.0**	**1.0**	**100.0**	1.0	**100.0**	1.0	**100.0**	1.0	**100.0**	**1.0**	**100.0**	1.0
		ResNet34	5%	**100.0**	**1.0**	**100.0**	**1.0**	**100.0**	**1.0**	**100.0**	**1.0**	**100.0**	1.0	**100.0**	**1.0**	**100.0**	1.0	**100.0**	**1.0**	**100.0**	1.0
	TinyImageNet	ResNet34	5%	**100.0**	1.7	**100.0**	1.7	**100.0**	1.7	**100.0**	1.7	**100.0**	1.7	**100.0**	1.7	**100.0**	1.7	**100.0**	1.7	**100.0**	1.7
		ResNet50	5%	**100.0**	**0.9**	**100.0**	**0.9**	**100.0**	**0.9**	**100.0**	**0.9**	**100.0**	**0.9**	**100.0**	**0.9**	**100.0**	0.9	**100.0**	**0.9**	**100.0**	**0.9**



(a) L_f vs. TPR/CCS (b) η vs. TPR/CCS

Figure 3: The impact of hyperparameters on TPR and CCS. L_f and η denote the final loss and learning rate, respectively.

TPR with 2.86% FPR, demonstrating its effectiveness and generality on large-scale datasets and models.

Performance Against Clean-Label Attacks. We evaluate clean-label backdoor attacks [32] on ResNet18 [10] using CIFAR10 [13] and ImageNette [4] at a 5% poisoning rate. The average ACC across all models is 87.9%, with a standard deviation of 5.1%. The ASR is 87.1% on average, with a standard deviation of 8.4%. The improved baselines achieve TPRs below 5%, indicating detection failure. In contrast, SPD attains TPRs of 64.3% and 70.1% with FPRs below 2.7%. These results indicate that SPD can effectively detect clean-label backdoors on pre-trained models. Compared to dirty-label attacks, the lower TPRs suggest that detecting clean-label backdoors on pre-trained models is more challenging and warrants further investigation.

Performance on Models Trained from Scratch. We evaluate SPD on models trained from scratch on the full GTSRB dataset under nine attacks, where it achieves 100% TPR and an FPR below 0.5%. More details are provided in Appendix E.

7.4 Hyperparameter Study

We evaluate the impact of hyperparameters on SPD using the ResNet18 [10] models trained on CIFAR10 [13] from Section 4.1. The reported values are averaged over both class-agnostic and class-specific attacks.

Impact of Initialization. To assess the impact of initialization on SPD, we evaluate four latent representation initializations: 0, 0.5, random, and 1. The average CCS is highest for 0 (4523) and lowest for 1 (3126), with random and 0.5 yielding 3833 and 3870, respectively. Since a lower CCS indicates faster convergence, initializing the latent representation with 1 leads to the fastest optimization. In terms of detection performance, the 1-initialization achieves the highest average TPR of 96.2%, while random yields the lowest at 58.5%. These results show that using 1 for initialization both accelerates convergence and improves performance, supporting the rationale behind our design.

Impact of the Final Loss. Figure 3(a) shows that setting the final loss either too small or too large degrades performance. A very small loss slows optimization, resulting in larger CCS, while an overly large loss leads to under-optimized latent representations and reduced TPR. A moderate loss balances both objectives, achieving smaller CCS and higher TPR. Based on these observations, we set the final-loss parameter to 10^{-3}.

Impact of the Learning Rate. Figure 3(b) reveals a similar trend. Extremely small learning rates result in slow convergence, larger CCS, and lower TPR, while excessively large rates cause instability and also degrade TPR. Mid-range values offer faster and more stable convergence, yielding smaller CCS and stronger detection performance. We therefore adopt 10^{-3} as the default learning rate.

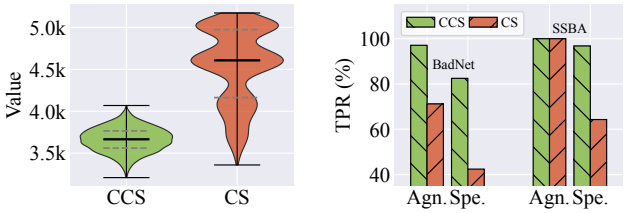
7.5 Ablation Study

In this section, we demonstrate the effectiveness of CCS and two anomaly metrics designed in Section 6. We propose two variants of SPD, *i.e.*, SPD-A and SPD-S. SPD-A only uses the anomaly metric (Equation 1) for class-agnostic attacks to conduct the backdoor detection. Similarly, SPD-S detects backdoor attacks only with the anomaly metric (Equation 2) for class-specific attacks.

The evaluation results are shown in Table 6. On class-

Table 6: Detection performance of SPD and two variants (*i.e.*, SPD-A and SPD-S) for class-agnostic and class-specific backdoor attacks via TPR (%) and FPR (%). The experimental setup follows Section 4.1.

Strategy	Dataset	Model	SPD		SPD-A		SPD-S	
			TPR	FPR	TPR	FPR	TPR	FPR
Agnostic	CIFAR10	RegNetY	88.0	**0.7**	**92.5**	2.3	87.2	**0.7**
		ResNet18	**97.5**	0.8	**97.5**	**0.5**	97.3	0.8
	GTSRB10	MobileNet	93.7	1.9	**95.4**	**0.7**	93.3	2.0
		ResNet18	**98.1**	0.2	98.0	**0.1**	**98.1**	0.2
	ImageNette	GoogLeNet	83.3	**0.0**	**85.2**	2.8	82.2	**0.0**
		ResNet18	**97.5**	**0.3**	96.3	0.9	97.4	**0.3**
Specific	CIFAR10	RegNetY	**63.4**	**1.4**	59.7	5.9	**63.4**	**1.4**
		ResNet18	**91.2**	3.9	80.8	**2.9**	**91.2**	3.9
	GTSRB10	MobileNet	**74.4**	6.4	72.3	**2.9**	72.8	5.8
		ResNet18	92.7	1.4	84.1	**1.3**	**93.2**	1.5
	ImageNette	GoogLeNet	**80.9**	**0.0**	51.8	6.4	80.8	**0.0**
		ResNet18	**100.0**	**0.0**	99.6	0.6	**100.0**	**0.0**



(a) Distribution of CS/CCS (b) CS/CCS vs. TPR

Figure 4: Violin plots of CCS and CS on benign models and TPRs obtained using CS and CCS for BadNet and SSBA attacks under class-agnostic and class-specific settings.

agnostic attacks, SPD-A attains the best performance, which matches the design intent. This suggests that the distribution of CCSs under class-agnostic attacks is well captured by the unimodal pattern targeted by SPD-A. On class-specific attacks, SPD-A lags far behind SPD-S across many settings. This indicates that the anomaly metric in SPD-S aligns with the CCS characteristics of class-specific attacks. Together, these results show that the two metrics are well matched to their respective attack strategies and support our insight for *L2* in Section 5.2. Compared with its single-metric variants, SPD integrates the strengths of both. Although it is slightly below SPD-A on some class-agnostic cases, it preserves the strong performance of SPD-S on class-specific attacks and sometimes exceeds it. This supports the design of SPD and provides the best balance of TPR and FPR.

For the effectiveness of CCS, *i.e.*, resetting the optimizer, it can provide a more stable and discriminative signal than raw CS on pre-trained models as shown in Figure 4(a).[5] Its benign distribution is tighter, reducing variance and spurious extremes. This stability makes attack-induced deviations more salient and improves separability between benign and backdoored models, as shown in Figure 4(b). This confirms the effectiveness of our design on CCS introduced in Section 6.1.

[5]CS is also initialized with an all-ones vector in our evaluation.

(a) Adaptive (b) Patch 1 (c) Patch 2 (d) Patch 3 (e) Patch 4

Figure 5: Case-study trigger designs. (a) AdaP adaptive trigger. (b)-(e) BadNet trigger variants.

7.6 Case Study: Trigger Impact

In this section, we employ BadNet with various patches to investigate why simple triggers are harder to detect than complex ones in pre-trained models. This inquiry is motivated by our observation that the findings diverge from prior work. In particular, we find that SPD achieves better detection results on AdaP, an adaptive patch-based trigger derived from BadNet, than on the original BadNet trigger. Therefore, we use this case study to explore the underlying reasons. The triggers used in Section 7.1 for AdaP and BadNet are shown in Figure 5(a) and Figure 5(b), respectively. They differ in color, size, and position. We conduct controlled experiments on RegNetY trained on CIFAR10, with the corresponding designs shown in Figure 5(c) – Figure 5(e). All other settings follow Section 4.1 with a 5% poisoning rate.

Our evaluation results show that trigger position is the dominant factor for SPD's detection performance. When the trigger is placed at the top left (Figure 5(b) and Figure 5(d)), performance degrades regardless of size or color, with TPRs of 56.6% and 72.0%, respectively. In contrast, placing the trigger at the bottom right (Figure 5(c) and Figure 5(e)) yields perfect detection with a TPR of 100%. This indicates that position primarily drives the observed variance.

Therefore, we hypothesize that when a trigger overlaps the salient features of a sample, it is more easily masked by representations induced by pre-trained weights. In contrast, when the trigger is placed away from the main features, it produces features that are decoupled from the original content, making them easier for detectors to identify. Furthermore, the poor detection performance reported in Table 9 is consistent with this observation. These results indicate that backdoor detection on pre-trained models remains challenging and that serious vulnerabilities persist, *e.g.*, sensitivity to trigger location.

8 Adaptive Attack

In this section, we further evaluate the robustness of data-free backdoor detectors under adaptive attacks. We categorize adaptive strategies into active and passive attacks according to whether attackers explicitly suppress the CCS-based anomaly.

Active attacks suppress the anomaly of a selected target class by modifying the training process, offering greater flexibility in target selection but requiring stronger attacker capability. Passive attacks instead exploit detector-specific assumptions as hypothesized in Section 6.3, making them less intrusive but also less general, as the target class cannot be freely specified.

8.1 Active Adaptive Attack

Temperature-Based Posterior Shaping. Inspired by FreeEagle [7], we employ posterior shaping as an active method to reduce CCS disparities between target and non-target classes. We further select temperature scaling [9, 41] to evaluate the robustness of SPD for posterior shaping during training. Temperature scaling modifies the smoothness of the probability distribution by introducing a temperature parameter T, which is defined as $\hat{p}_i(x) = \frac{\exp(f_i(x)/T)}{\sum_{j=1}^{n}\exp(f_j(x)/T)}$, where $\hat{p}_i(x)$ is the probability of the i-th class after applying the softmax function and $T > 0$. When $T > 1$, increasing T smooths the probability distribution and reduces the confidence of the model.

We evaluate this adaptive attack using ResNet18 trained on CIFAR10 under three class-agnostic attacks [3, 8, 16] with a poisoning rate of 10% and five temperatures, *i.e.*, $T \in \{1.5, 2, 2.5, 3, 10\}$. The average ACC across all models is 92.10%, with a standard deviation of 0.7%. The ASR is 97.47% on average, with a standard deviation of 2.4%. SPD achieves 100% TPR on all evaluated models using the thresholds calibrated from normally trained models. This indicates that temperature-based posterior shaping cannot bypass SPD. More specifically, for BadNet, the anomaly score decreases as the temperature increases, but remains far above the detection threshold. For Blending and Filter, the anomaly scores instead increase with larger temperatures. Overall, these results suggest that posterior smoothing cannot eliminate the convergence-speed disparity captured by SPD.

8.2 Passive Adaptive Attack

CCS-Based Attack. Inspired by our insight that source-class CCS tends to slow down while target-class CCS tends to speed up, we design this adaptive attack where the targets are the last five classes with the largest CCS and the sources are the top five with the smallest CCS. We denote this configuration as Last-5. The rank of CCS is derived from a benign model trained on the same dataset. For comparison, we also run the opposite configuration in which the targets are the top five and the sources are the last five, denoted as Top-5. Experiments are conducted on CIFAR100 [13] and TinyImageNet [14] with ResNet34 [10]. The poisoning rates are 0.5% and 0.25%, respectively. The ASRs reach 75.7% on average with a standard deviation of 16.6% for Last-5, and 87.6% with a standard deviation of 9.7% for Top-5.

Table 7: Detection performance of SPD and improved baselines under adaptive attacks via TPR (%) and FPR (%).

Attack	Dataset	Target Class	FreeEagle$^+$		MMBD$^+$		SPD	
			TPR	FPR	TPR	FPR	TPR	FPR
BadNet	CIFAR100	Top-5	0.0	2.3	14.7	8.3	**95.3**	**1.4**
		Last-5	0.0	**2.3**	**1.1**	6.4	0.0	3.4
	TinyImageNet	Top-5	0.0	3.7	40.1	6.6	**91.9**	**1.1**
		Last-5	0.0	**3.7**	4.5	7.7	**29.1**	4.9
Blending	CIFAR100	Top-5	0.0	2.9	28.8	6.6	**93.5**	**2.6**
		Last-5	**8.3**	4.2	0.6	6.3	0.6	**3.9**
	TinyImageNet	Top-5	0.0	3.7	70.6	5.1	**96.2**	**1.4**
		Last-5	0.0	**3.7**	11.2	7.1	**34.7**	4.3
Filter	CIFAR100	Top-5	0.0	2.9	33.3	8.3	**100.0**	**1.3**
		Last-5	**16.4**	3.7	4.5	6.3	0.0	**3.4**
	TinyImageNet	Top-5	0.0	**3.7**	82.3	4.6	**92.0**	**3.7**
		Last-5	0.0	**3.7**	7.8	5.1	**32.8**	7.1
Refool	CIFAR100	Top-5	0.0	2.3	25.3	8.9	**95.9**	**1.3**
		Last-5	15.8	3.5	0.0	7.6	**29.9**	**3.0**
	TinyImageNet	Top-5	0.0	3.7	75.4	3.4	**91.3**	**1.4**
		Last-5	0.0	**3.7**	13.0	7.4	**74.5**	4.6
LF	CIFAR100	Top-5	0.0	2.3	38.1	8.9	**91.2**	**1.3**
		Last-5	**16.9**	5.4	1.1	6.7	5.2	**4.1**
	TinyImageNet	Top-5	0.0	3.7	52.7	3.7	**85.7**	**1.4**
		Last-5	0.0	**3.7**	7.9	7.7	**40.7**	4.9
SSBA	CIFAR100	Top-5	0.0	2.3	29.2	6.7	**97.1**	**1.3**
		Last-5	13.4	**4.5**	2.9	6.3	**43.6**	6.0
	TinyImageNet	Top-5	0.0	3.7	61.0	5.1	**87.8**	**2.0**
		Last-5	0.0	**3.7**	20.3	5.1	**57.6**	5.4

Table 7 shows that the adaptive strategy reduces the detection performance of SPD across attacks and datasets. Comparing the Top-5 and Last-5 settings within each configuration, degradation is widespread. Choosing targets from classes with large CCS prevents the target from becoming the global minimum, which weakens the signal for detection. The attack is also effective against the improved baselines: FreeEagle$^+$ collapses to near-zero TPR in all configurations, and MMBD$^+$ drops substantially in many cases. Although SPD remains state-of-the-art among the improved baselines, the overall decline under adaptation is concerning.

In particular, each row in Table 7 corresponds to experiments on only 25 out of the 99×100 or 199×200 possible class pairs. Thus, the low performance observed in a particular row is a local phenomenon rather than a characterization of the full dataset. A broader evaluation over all class combinations is therefore required, and these cases alone should not be interpreted as evidence of a fundamental failure of SPD.

All-to-All Backdoor Attacks. Current data-free detectors share a common premise: their insights rely on the contrast between the target class and non-target classes. However, when a model contains multiple targets, this contrast becomes less pronounced. In the extreme all-to-all setting, where every class is a target, the contrast may vanish entirely. We evaluate all-to-all attacks on CIFAR10 [13], GTSRB10 [11], and ImageNette [4] with ResNet18 [10] under nine attacks [3, 8, 15–17, 19, 20, 37, 44], using a poisoning rate of 10%. The average ACC across all models is 89.8%, with a standard deviation of 5.8%. The ASR is 55.7% on average,

with a standard deviation of 27.2%.

Overall, FreeEagle and MMBD fail completely, with TPRs of 0%. Other methods exhibit a polarized pattern, with many configurations achieving either 100% or 0% TPR. For example, SPD achieves 100% TPR on ImageNette under WaNet, but only 0% on CIFAR10 for the same attack. No consistent trend is observed across datasets or attack types. FreeEagle$^+$ performs perfectly in twelve settings, in stark contrast to the complete failure of FreeEagle. These results suggest that once the contrast between target and non-target classes is removed, detection becomes unstable and highly dataset dependent. This highlights the need for detectors that remain reliable under multi-target settings and motivates the development of new metrics that do not rely on single-target asymmetry.

Mitigation. A straightforward refinement builds on the boundary determination strategy used in BARBIE [45]. Instead of relying solely on the minimum-CCS criterion, we compare the CCS ranking of the model under inspection with that of a benign reference. While the adaptive strategy may prevent the target class from having the absolute minimum CCS, it cannot fully suppress the CCS reduction. A ranking-based comparison captures this subtle shift with finer granularity. Preliminary results suggest that a ranking-based refinement can substantially recover detection performance, but a systematic evaluation is left for future work.

For example, on CIFAR100, consider a model where, in the benign reference, the target class ranks 96th and the source class ranks 5th in the CCS ordering. After poisoning, their ranks shift to 33rd and 67th, respectively. This change is sufficiently pronounced at the ranking level to be captured by a more fine-grained detection criterion. In addition, all-to-all settings violate the single-target-class assumption adopted by most existing data-free approaches, leading to irregular detection behavior in this regime and further exposing the limitations of current data-free backdoor detection methods.

It is also worth noting that both effective adaptive strategies exhibit notable limitations in our experiments. CCS-based attacks rely on tasks with a large number of classes, and the Last-5 variant shows a pronounced drop in ASR compared to Top-5. All-to-all configurations suffer an even larger degradation in ASR than standard single-target backdoor attacks. Taken together, these results suggest that progress is needed not only on detection techniques, but also on the design and evaluation of backdoor methods themselves.

9 Limitations and Future Work

We conduct comprehensive evaluations on three small-scale datasets and targeted evaluations on two large-scale datasets. A full-scale evaluation on large datasets is not performed, as it would require training an impractically large number of models. For example, in a 100-class dataset, evaluating all combinations of target and source classes would require training 100×99 models, which poses a significant challenge in terms of time and hardware resources. This motivates theoretical analysis on backdoor attacks in future work.

In addition, most of the attacks we reproduce are sourced from BackdoorBench [39]. As a result, our evaluation may miss some of the most recent backdoor attacks. This is due to the difficulty of conducting unified experiments across diverse and often inconsistent open-source implementations. Nevertheless, our primary goal is to highlight the false sense of security offered by current data-free detection systems and expose their underlying vulnerabilities. Even without incorporating the latest attacks, the vulnerabilities revealed by our current evaluation are already severe.

Finally, our results reveal that SPD falls short of near-perfect behavior under some configurations. Nevertheless, across all the architectures, datasets, and attack types we evaluate, SPD still outperforms both the original and the improved baselines, and thus remains state-of-the-art. Rather than being interpreted as evidence of a complete solution, we hope these findings will motivate future work on more robust detection approaches. In particular, our study is intended to caution researchers against an overly optimistic view of the reliability of current backdoor detection methods.

10 Conclusion

In this paper, we present the first large-scale evaluation of data-free detection methods on pre-trained models. Based on experimental results, we uncover serious vulnerabilities that have been overlooked in prior work. To address these issues, we analyze the underlying causes of failure and propose effective modifications for the data-free detection framework. Finally, we introduce SPD, a new data-free detector that leverages the convergence speed as a side-channel signal. Although SPD achieves state-of-the-art performance, our broader evaluation shows that backdoor threats in pre-trained models remain unresolved. We hope that our findings can inspire the community to revisit backdoor attacks and develop more robust defenses for machine learning systems.

Acknowledgments

We sincerely thank the anonymous shepherd and reviewers for their constructive and insightful feedback throughout the review process. This work was supported in part by the National Key R&D Program of China under Grant 2025ZD0123604, in part by the National Natural Science Foundation of China under Grants U24A20237, U22A2029, and 62402379, in part by the Fundamental Research Funds for the Central Universities under Grant xzy012026027, and in part by the Key R&D Program of Shandong Province, China under Grant 2025CXPT085. Linkang Du and Yuntao Wang were also supported by the Youth Innovation Team of Shaanxi Universities.

Ethical Considerations

The goal of our research is to detect well-known backdoor attacks in DNN models and enhance the security of model deployments. The threat posed by backdoor attacks has long been recognized and widely implemented using open-source code. In our implementation, we strictly adhere to relevant legal and regulatory frameworks and use open-source code that fully complies with these guidelines.

During the design and implementation phase of SPD, we prioritize ethical considerations to ensure that our work not only strengthens security but also enhances social value while minimizing potential risks. In this paper, we elaborate on our insights and considerations based on ethical principles to encourage the community to explore more effective detection mechanisms against backdoor attacks.

Stakeholder Analysis and Process Impact. Our study focuses on data-free detection methods evaluated on pre-trained models and the design of a new detector, SPD. The main stakeholders are end users of machine-learning systems, practitioners and organizations that deploy pre-trained models, model providers that publish such models, and the research community that develops backdoor detection techniques. All experiments use standard public datasets and open-source implementations and are purely computational. The direct impact of the research process is therefore to change how these stakeholders understand the reliability and limitations of current data-free approaches on pre-trained models, rather than to affect individuals or private data directly.

Impact of the Research. Experimentally, we present the first large-scale evaluation of data-free detection methods on pre-trained models and uncover serious vulnerabilities that have been overlooked in prior work. We analyze the underlying causes of these failures, propose effective modifications to the data-free detection framework, and introduce SPD, which leverages convergence speed as a side-channel signal. The intended impact is to improve the toolbox of practitioners by offering a more robust data-free detector and, equally importantly, to draw attention to the fact that widely used methods may be less reliable on pre-trained models than previously believed. In doing so, we aim to prompt more cautious use of existing detectors and more careful evaluation practices.

Mitigation. To mitigate potential harms from revealing weaknesses in existing methods, we couple all negative findings with constructive contributions. We do not modify or attack deployed systems; instead, we work with pre-trained models obtained and trained in controlled experimental settings. When we demonstrate that prior data-free methods can fail, we also provide analysis and concrete framework modifications, together with SPD, to help practitioners better assess and improve their deployments. Throughout the paper, we explicitly state the scope and limitations of our results, so as to avoid giving the impression that any single detection method, including ours, is a complete solution.

Justification for Research. The decision to conduct and publish this work rests on the view that openly exposing overlooked vulnerabilities, while simultaneously providing improved detection mechanisms, produces a clear net benefit for stakeholders. If such weaknesses were left unexamined, practitioners and end users could develop an unwarranted confidence in current data-free methods. By contrast, our results are intended to raise awareness of the risks associated with backdoor behavior in pre-trained models, to encourage more realistic evaluations, and to stimulate the development of more robust defenses for real-world machine-learning systems.

Open Science

In line with the 2026 USENIX Security Open Science Policy, this paper provides all necessary artifacts to facilitate the evaluation of the contributions made as follows.

- **Source Code:** The attack code used in the experiments is derived from the codebases of BackdoorBench [39], FreeEagle [7], and AdaP [23]. The detection code developed for this paper is also open-sourced.
- **Datasets and Configurations:** We provide instructions for obtaining the datasets used in our experiments. We also include the experimental configurations, including hyperparameters and training settings.
- **Access:** The source code, datasets, and experimental configurations are available at https://doi.org/10.5281/zenodo.20440930. The README file provides detailed instructions on setting up the environment and running the code to replicate the results.

References

[1] Hugging Face. https://huggingface.co/models, 2018.

[2] Zeyuan Allen-Zhu and Yuanzhi Li. Feature purification: How adversarial training performs robust deep learning. In *FOCS*, 2022.

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

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[5] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[7] Chong Fu, Xuhong Zhang, Shouling Ji, Ting Wang, Peng Lin, Yanghe Feng, and Jianwei Yin. FreeEagle: Detecting complex neural trojans in data-free cases. In *USENIX Security*, 2023.

[8] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 2019.

[9] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[11] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German traffic sign detection benchmark. In *IEEE IJCNN*, 2013.

[12] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. BadEncoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *IEEE S&P*, 2022.

[13] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Computer Science University of Toronto*, 2009.

[14] Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. *CS 231N*, 2015.

[15] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *ICCV*, 2021.

[16] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. ABS: Scanning neural networks for back-doors by artificial brain stimulation. In *CCS*, 2019.

[17] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *ECCV*, 2020.

[18] Peizhuo Lv, Chang Yue, Ruigang Liang, Yunfei Yang, Shengzhi Zhang, Hualong Ma, and Kai Chen. A data-free backdoor injection approach in neural networks. In *USENIX Security*, 2023.

[19] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *NeurIPS*, 2020.

[20] Tuan Anh Nguyen and Anh Tuan Tran. WaNet - Imperceptible warping-based backdoor attack. In *ICLR*, 2021.

[21] Daniel W. Otter, Julian R. Medina, and Jugal K. Kalita. A survey of the usages of deep learning for natural language processing. *IEEE TNNLS*, 2021.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019.

[23] Xiangyu Qi, Tinghao Xie, Yiming Li, Saeed Mahloujifar, and Prateek Mittal. Revisiting the assumption of latent separability for backdoor defenses. In *ICLR*, 2023.

[24] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *CVPR*, 2020.

[25] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 2017.

[26] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

[27] Zeyang Sha, Xinlei He, Pascal Berrang, Mathias Humbert, and Yang Zhang. Fine-tuning is all you need to mitigate backdoor attacks. *arXiv preprint arXiv:2212.09067*, 2022.

[28] Shawn Shan, Arjun Nitin Bhagoji, Haitao Zheng, and Ben Y. Zhao. Poison forensics: Traceback of Data Poisoning Attacks in Neural Networks. In *USENIX Security*, 2022.

[29] Guangyu Shen, Yingqi Liu, Guanhong Tao, Shengwei An, Qiuling Xu, Siyuan Cheng, Shiqing Ma, and Xiangyu Zhang. Backdoor scanning for deep neural networks through K-Arm optimization. In *ICML*, 2021.

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[31] 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*, 2021.

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

[33] 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 S&P*, 2019.

[34] Hang Wang, Zhen Xiang, David J Miller, and George Kesidis. MM-BD: Post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic. In *IEEE S&P*, 2024.

[35] Ren Wang, Gaoyuan Zhang, Sijia Liu, Pin-Yu Chen, Jinjun Xiong, and Meng Wang. Practical detection of trojan neural networks: Data-limited and data-free cases. In *ECCV*, 2020.

[36] Shuo Wang, Surya Nepal, Carsten Rudolph, Marthie Grobler, Shangyu Chen, and Tianle Chen. Backdoor attacks against transfer learning with pre-trained deep learning models. *IEEE Transactions on Services Computing*, 2020.

[37] 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 *CVPR*, 2022.

[38] Cheng'an Wei, Yeonjoon Lee, Kai Chen, Guozhu Meng, and Peizhuo Lv. Aliasing backdoor attacks on pre-trained models. In *USENIX Security*, 2023.

[39] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoorbench: A comprehensive benchmark of backdoor learning. *NeurIPS*, 2022.

[40] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A Gunter, and Bo Li. Detecting AI trojans using meta neural analysis. In *IEEE S&P*, 2021.

[41] Hao Xuan, Bokai Yang, and Xingyu Li. Exploring the impact of temperature scaling in softmax for classification and adversarial robustness. *arXiv preprint arXiv:2502.20604*, 2025.

[42] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *NeurIPS*, 27, 2014.

[43] Xuejing Yuan, Yuxuan Chen, Yue Zhao, et al. CommanderSong: A systematic approach for practical adversarial voice recognition. In *USENIX Security*, 2018.

[44] Yi Zeng, Won Park, Z Morley Mao, and Ruoxi Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *ICCV*, 2021.

[45] Hanlei Zhang, Yijie Bai, Yanjiao Chen, Zhongming Ma, and Wenyuan Xu. BARBIE: Robust backdoor detection based on latent separability. In *NDSS*, 2025.

A More Details on the Empirical Study

In the following, we provide additional results to substantiate our insights. We validate the above intuition on the ResNet18 model with the CIFAR10 dataset. Models are trained in three settings: benign, injected with a class-agnostic backdoor, and injected with a class-specific backdoor. We then record the number of optimization steps as a measure of convergence speed. The procedure for building the latent representation is stopped after reaching a loss value of 0.001 or 10,000 steps.

The results in Figure 7 align with our intuition, showing that the convergence speed of the target class is significantly faster than that of the non-target class, both under class-agnostic and class-specific backdoor attacks. Furthermore, the gap in convergence speeds between the target class and the others is more pronounced under class-agnostic backdoor attacks than under class-specific backdoor attacks. In particular, the convergence speed of the source class is substantially slower than that of the clean class under class-specific backdoor attacks. Therefore, we infer that the convergence speed of latent representation construction can serve as an anomaly metric to identify both target and source classes.

Additionally, we quantify the impact of pre-trained weights by convergence speeds as illustrated in Figure 8. Figure 8(a) demonstrates that the target class converges significantly faster than other classes across all layers of the model without pre-training. However, when model training is initialized with pre-trained weights, the convergence speed of the target class is not the fastest in some layers, as shown in Figure 8(b). For instance, Class 6 converges faster than the target class (Class 1) in the first layer. Moreover, the convergence speed distribution of the input layer in a benign model initialized with pre-trained weights deviates from that of the other layers, as shown in Figure 8(c). These observations reveal potential factors through which pre-trained weights may affect the detection of both backdoored and benign models.

B Details of Datasets

CIFAR10 [13] is a widely used benchmark dataset for evaluating machine learning algorithms. It consists of 60,000 32x32 color images across 10 classes, with 6,000 images per class. The dataset is split into 50,000 training images and 10,000 test images, covering a variety of objects such as airplanes, cars, and birds.

CIFAR100 [13] is an extension of CIFAR10 and serves as a more fine-grained benchmark dataset for evaluating machine learning algorithms. It consists of 60,000 32x32 color images across 100 classes, with 600 images per class. It is split into 50,000 training images and 10,000 test images, and its classes are organized into 20 superclasses covering a variety of objects such as animals, vehicles, and household items.

GTSRB (German Traffic Sign Recognition Benchmark) [11] is a dataset designed for traffic sign classifica-

Table 8: Detection performance of data-free methods under class-agnostic backdoor attacks at a 5% poisoning rate via TPR (%) and FPR (%). The experimental setup is as described in Section 4.1.

Method	Dataset	Model	BadNet		Blending		Filter		Refool		WaNet		LF		BPP		SSBA		IAB		AdaP		AdaB	
			TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
DF-TND	CIFAR10	RegNetY	**15.9**	5.5	8.3	5.3	**10.0**	5.1	10.1	5.3	3.2	4.2	5.4	5.0	5.9	5.0	8.9	5.3	4.4	5.0	6.9	4.7	8.1	5.2
		ResNet18	**20.6**	4.6	7.9	4.3	0.9	3.9	26.9	4.5	**87.4**	3.8	9.9	4.4	**35.1**	4.6	**66.2**	3.7	2.0	3.2	13.6	4.6	5.7	3.4
	GTSRB10	MobileNet	2.9	6.1	2.2	4.8	1.9	6.4	1.7	3.8	0.2	**2.4**	1.7	4.2	1.3	**2.6**	2.9	5.5	1.1	4.2	2.0	6.4	2.9	6.5
		ResNet18	24.3	3.3	3.1	**2.6**	17.1	3.1	29.7	3.6	**76.8**	3.6	5.0	**3.0**	61.9	3.6	0.6	**2.6**	47.2	3.6	0.0	**1.3**	0.0	**1.3**
	ImageNette	GoogLeNet	6.7	3.7	8.5	3.6	4.9	3.5	**15.5**	4.1	2.6	3.4	8.2	4.0	2.0	3.1	10.4	3.6	0.1	**2.1**	2.2	**2.6**	0.1	**2.5**
		ResNet18	**28.4**	5.9	7.0	5.7	7.8	5.5	**36.3**	5.9	**69.5**	5.7	9.7	5.7	**68.0**	5.9	6.3	5.5	21.8	5.9	**74.7**	4.2	**72.0**	5.9
FreeEagle	CIFAR10	RegNetY	1.6	**2.9**	3.3	**2.9**	2.7	**3.1**	2.9	**2.9**	7.6	**3.4**	1.9	**2.6**	**11.4**	**3.1**	6.1	**3.3**	5.6	**2.9**	9.4	**3.5**	6.4	**2.9**
		ResNet18	0.0	**2.3**	0.0	**2.3**	0.0	**2.3**	0.0	**2.3**	3.0	4.5	0.0	**2.3**	0.0	3.1	0.0	**2.3**	0.0	**2.3**	12.5	4.7	13.0	4.7
	GTSRB10	MobileNet	0.1	**1.9**	0.3	**1.6**	0.8	**2.3**	1.9	**3.1**	15.2	3.4	0.4	**1.8**	5.0	3.3	0.4	**2.3**	7.6	**3.4**	9.4	**3.4**	1.0	**3.1**
		ResNet18	4.1	**2.6**	1.2	3.4	1.7	**1.9**	1.2	**1.2**	7.1	**3.0**	3.3	3.1	8.9	**3.4**	1.4	2.8	6.0	**3.1**	24.1	1.4	17.2	2.9
	ImageNette	GoogLeNet	0.2	**1.8**	0.0	**1.2**	0.1	**2.0**	0.3	**1.9**	9.0	**3.0**	0.1	**1.7**	10.7	**3.0**	2.4	**2.8**	1.9	2.2	4.9	**2.6**	3.6	2.9
		ResNet18	9.9	**2.9**	6.5	**2.7**	3.6	**2.9**	2.6	**2.5**	12.7	**3.1**	7.1	**3.1**	0.0	**1.1**	10.7	**3.2**	2.8	**2.4**	10.2	**1.6**	6.5	**2.6**
MMBD	CIFAR10	RegNetY	9.1	4.8	**32.3**	5.8	9.4	5.6	**15.2**	4.7	2.4	4.9	5.6	4.3	4.1	3.6	11.3	5.4	6.8	4.9	17.5	5.6	12.4	5.4
		ResNet18	3.8	4.7	3.5	4.4	2.3	4.7	9.9	5.4	1.4	4.4	11.6	5.1	0.0	**1.8**	3.4	4.6	6.5	4.3	10.4	5.6	0.0	**1.1**
	GTSRB10	MobileNet	**25.8**	5.2	**23.0**	4.9	**44.3**	5.4	12.6	5.4	16.5	4.4	15.5	5.5	36.3	5.3	5.8	4.4	10.5	5.2	19.3	4.9	8.7	4.6
		ResNet18	**29.0**	5.6	14.4	6.0	23.5	5.9	36.8	6.0	12.9	6.0	**16.3**	6.1	65.1	5.5	11.5	5.6	26.3	6.0	10.7	5.8	18.1	5.8
	ImageNette	GoogLeNet	**13.0**	4.3	**20.1**	4.6	**11.4**	4.6	13.4	3.5	**39.9**	5.1	**12.2**	4.6	**43.7**	5.0	**17.2**	5.0	**26.9**	4.9	**36.9**	3.6	**43.0**	4.2
		ResNet18	13.0	6.7	11.6	5.9	16.0	6.0	27.5	6.8	12.9	6.3	10.0	6.6	1.3	4.1	19.6	5.9	24.6	6.9	38.2	6.1	30.8	6.5
BARBIE	CIFAR10	RegNetY	5.7	4.8	7.0	4.8	8.4	4.8	9.3	4.8	**39.0**	4.8	**14.8**	4.8	8.4	4.8	**13.6**	4.8	**13.4**	4.8	**77.4**	4.8	**22.1**	4.8
		ResNet18	8.1	2.4	**16.9**	2.4	**16.4**	2.4	**51.4**	2.4	61.0	**2.4**	**37.0**	2.4	30.0	2.4	13.1	2.4	**12.5**	2.4	**49.2**	**2.4**	**17.8**	2.4
	GTSRB10	MobileNet	12.1	7.6	13.7	7.6	25.8	7.6	**86.0**	7.6	**97.3**	7.6	**17.2**	7.6	**82.2**	7.6	**12.9**	7.6	**72.5**	7.6	**98.2**	7.6	**40.2**	7.6
		ResNet18	19.7	8.4	**23.6**	8.4	**51.0**	8.4	**65.6**	8.4	56.4	8.4	15.0	8.4	**74.6**	8.4	**13.4**	8.4	**86.4**	8.4	**30.4**	8.4	**26.2**	8.4
	ImageNette	GoogLeNet	10.1	8.1	6.7	8.1	7.6	8.1	7.2	8.1	17.4	8.1	7.2	8.1	15.6	8.1	8.2	8.1	7.9	8.1	13.8	8.1	11.1	8.1
		ResNet18	26.6	10.4	**17.4**	10.4	**24.4**	10.4	29.7	10.4	64.2	10.4	**18.4**	10.4	53.8	10.4	**22.4**	10.4	**26.8**	10.4	44.5	10.4	38.9	10.4

tion. It contains over 50,000 images in 43 classes, with the images divided into 39,209 training samples and 12,603 test samples. In our experiments, we selected the first ten classes with more than 500 samples and balanced them to 500 images per class as GTSRB10. The corresponding labels are $1, 2, 3, 4, 5, 7, 8, 9, 10, 11$.

ImageNet [4] is a large-scale image classification dataset. For ImageNet fine-tuning, we select the first 50 training samples from each class according to the original dataset order. ImageNette is its 10-class subset, and we downsample each class to 858 images for balance.

TinyImageNet [14] is a downsampled subset of the ImageNet dataset commonly used for benchmarking image classification and representation learning methods. It contains 200 classes of 64x64 color images, with 100,000 training images, 10,000 validation images, and 10,000 test images.

C Details of Backdoor Patterns

We consider eleven representative backdoor attack patterns widely used in prior work, including BadNet [8], Blending [3], Filter [16], Refool [17], WaNet [20], LF [44], BPP [37], SSBA [15], IAB [19], AdaP, and AdaB [23], as illustrated in Figure 9. These patterns cover localized patch triggers, globally blended patterns, image filtering-based triggers, geometric warping-based triggers, and sample-specific or adaptive triggers. For all attacks, we directly use publicly available open-source implementations to generate poisoned images, without any manual modification of the trigger designs. This ensures that our evaluation is faithful to the original attack settings and facilitates reproducibility.

D Analysis of Attack Effectiveness

Figure 6 shows that most backdoored models keep ACC close to the corresponding benign models, while achieving high ASR in many settings. This indicates that the generated models largely preserve normal classification performance while learning the backdoor behavior. The results also show that ASR is sensitive to datasets, models, and attack methods. On CIFAR10 and GTSRB10, most attacks achieve consistently high ASR across different models. In contrast, ImageNette shows lower and more variable ASR, especially for attacks such as Refool and SSBA. Different models on the same dataset also lead to different ASR, suggesting that attack effectiveness depends on both data distribution and model architecture. The comparison across poisoning settings supports our choice of poisoning rates. Increasing the class-agnostic poisoning rate from 5% to 10% generally improves ASR with only a small ACC drop. For weaker attacks, a lower poisoning rate may fail to ensure sufficiently high ASR. Therefore, the selected poisoning rates provide a reasonable balance between attack effectiveness and model utility.

E Results on Models Trained from Scratch

As shown in Figure 8(a), SPD is effective on models trained from scratch. To further demonstrate its robustness, we con-

Table 9: Detection performance of SPD and improved baselines under class-agnostic backdoor attacks at a 5% poisoning rate via TPR (%) and FPR (%). The experimental setup is as described in Section 4.1.

Method	Dataset	Model	BadNet		Blending		Filter		Refool		WaNet		LF		BPP		SSBA		IAB		AdaP		AdaB	
			TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
FreeEagle$^+$	CIFAR10	RegNetY	21.3	5.3	9.9	5.5	21.4	5.8	25.0	5.6	80.6	4.0	22.7	5.4	24.8	3.9	22.1	3.9	12.6	4.1	88.6	4.0	23.9	4.1
		ResNet18	10.8	5.4	16.7	6.1	9.5	4.4	15.2	5.2	**100.0**	**0.0**	22.3	5.9	77.1	5.4	24.7	5.7	46.9	5.9	99.4	0.1	61.8	5.9
	GTSRB10	MobileNet	13.0	3.9	20.6	4.9	20.2	5.1	97.5	2.6	97.8	3.9	13.3	4.7	94.9	4.3	10.6	4.5	94.9	4.3	**100.0**	1.4	85.4	4.7
		ResNet18	14.7	3.7	13.2	5.0	24.0	3.4	95.6	2.9	**100.0**	0.9	23.6	5.3	99.9	0.9	18.3	5.3	99.1	1.0	**100.0**	0.9	91.5	3.6
	ImageNette	GoogLeNet	12.3	4.8	8.1	4.8	17.6	4.8	13.9	4.4	29.1	5.2	13.1	4.4	37.1	5.4	22.9	5.2	12.8	5.3	11.6	4.4	6.1	4.9
		ResNet18	19.9	3.5	22.2	3.7	16.3	3.5	28.5	3.5	85.5	2.6	40.1	3.6	97.7	2.6	9.4	3.1	30.6	2.6	40.2	2.8	32.0	3.7
MMBD$^+$	CIFAR10	RegNetY	23.3	**3.4**	25.4	**2.2**	31.8	**3.3**	47.0	**3.6**	97.1	1.4	37.3	**3.6**	**88.6**	**3.0**	98.3	2.6	46.1	**2.8**	61.0	3.0	96.5	3.1
		ResNet18	43.9	**4.7**	71.6	4.6	63.2	4.6	97.6	1.8	**100.0**	**0.0**	75.1	4.7	**100.0**	**0.0**	**100.0**	**0.0**	**99.1**	2.9	**100.0**	**0.0**	**100.0**	**0.0**
	GTSRB10	MobileNet	24.5	**3.5**	37.4	**3.2**	37.4	3.8	35.3	3.7	30.5	3.4	27.8	3.6	15.8	2.9	16.1	**3.5**	19.0	3.8	77.8	3.6	48.8	2.7
		ResNet18	9.7	4.5	11.5	5.5	20.5	5.4	96.7	3.4	81.3	3.3	8.5	3.9	98.2	3.3	9.9	4.6	94.8	4.2	99.0	3.3	40.4	5.7
	ImageNette	GoogLeNet	57.9	4.9	**52.4**	4.9	**77.7**	4.6	**63.4**	4.6	85.9	2.7	**75.4**	4.5	82.0	2.9	**85.4**	4.6	**82.1**	4.7	75.6	3.0	**93.9**	**3.9**
		ResNet18	47.7	5.0	73.3	5.1	53.4	5.1	81.6	4.6	**98.5**	**1.9**	52.8	4.9	**99.9**	1.1	49.9	5.1	**85.5**	4.6	99.4	1.0	97.0	4.1
SPD	CIFAR10	RegNetY	**56.6**	4.3	**73.0**	4.3	**68.4**	4.4	**90.5**	4.1	**97.6**	**0.6**	**71.6**	4.3	84.4	4.2	**99.7**	**0.0**	**60.9**	4.2	**100.0**	**0.0**	**100.0**	**0.0**
		ResNet18	**67.2**	5.1	**93.6**	**3.8**	**92.7**	**3.4**	**100.0**	**0.3**	**100.0**	0.3	**94.4**	**4.2**	**100.0**	0.3	**100.0**	0.3	93.9	**2.7**	**100.0**	0.3	**100.0**	0.3
	GTSRB10	MobileNet	**82.2**	4.4	**76.9**	3.7	**92.6**	**3.7**	**100.0**	**0.0**	**98.3**	**0.0**	**81.4**	**3.4**	**99.3**	**0.3**	**42.9**	4.6	**99.1**	**0.0**	**100.0**	**0.0**	**99.5**	**1.3**
		ResNet18	**91.6**	**2.4**	**97.4**	**4.6**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**97.6**	**3.0**	**100.0**	**0.0**	**76.0**	**3.4**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**
	ImageNette	GoogLeNet	**75.6**	**3.8**	31.3	**3.6**	71.7	**3.8**	55.9	**3.8**	**99.3**	**0.9**	67.3	**3.9**	**97.5**	**0.6**	76.5	**3.2**	47.6	**3.8**	**100.0**	**0.9**	89.9	**3.9**
		ResNet18	**98.8**	**0.4**	**100.0**	**0.1**	**100.0**	**0.1**	**100.0**	**0.1**	95.9	2.1	**99.2**	**0.1**	95.5	**0.4**	**100.0**	**0.1**	55.0	**2.1**	**100.0**	**0.1**	**100.0**	**0.1**

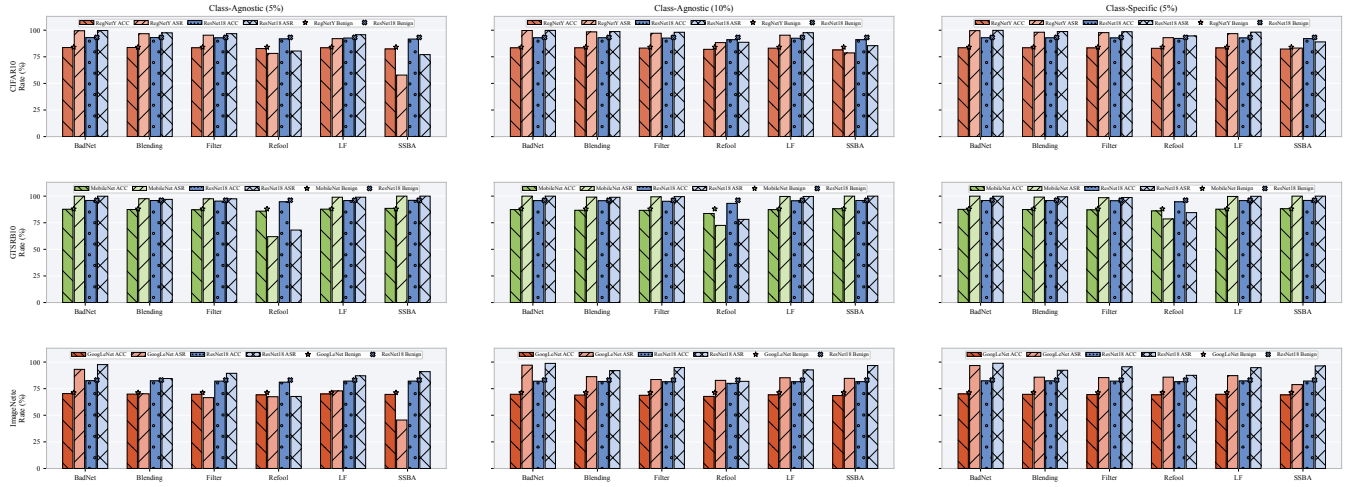
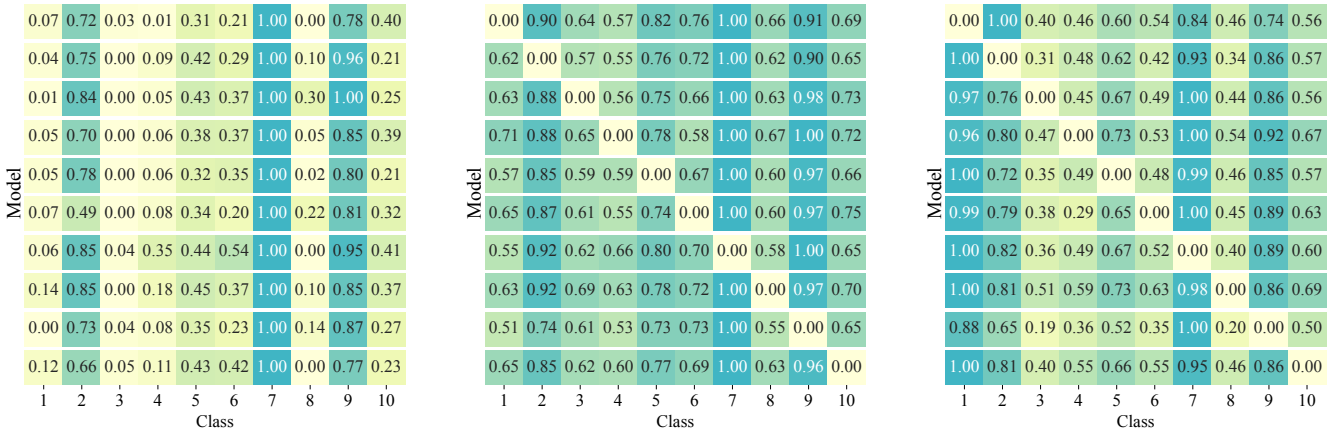


Figure 6: ACC and ASR under different datasets, model architectures, and backdoor methods. Each row corresponds to one dataset, and each column corresponds to one poisoning setting. The x-axis shows different backdoor methods. Bars report ACC and ASR for the two model architectures evaluated on each dataset, with colors and hatching jointly indicating the model-metric combination. Markers denote the ACC of benign models.

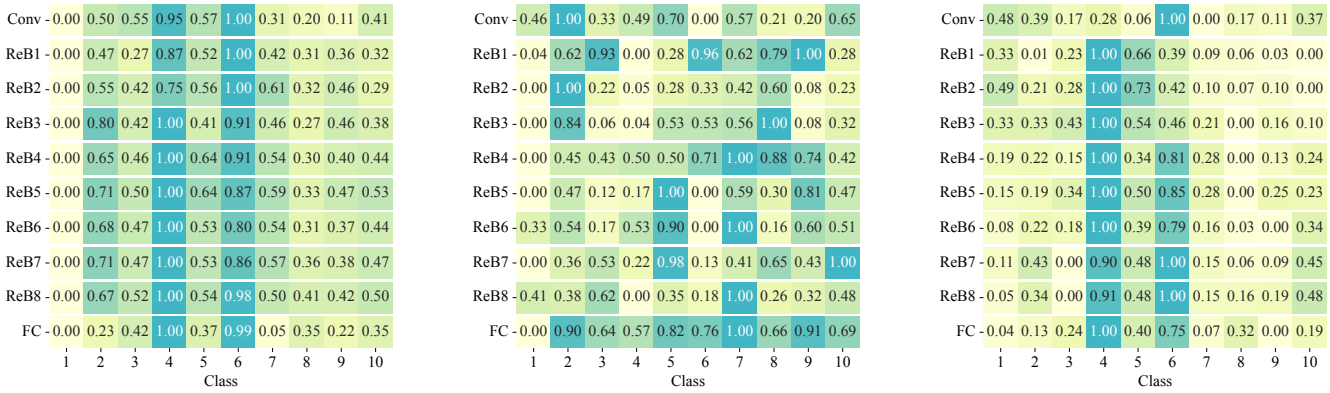
duct an additional evaluation on the GTSRB [11] dataset using GoogLeNet [30] without pre-trained weights. We employ nine backdoor attacks, including BadNet [8], Blending [3], Filter [16], Refool [17], WaNet [20], LF [44], BPP [37], SSBA [15], and IAB [19], under a class-agnostic poisoning strategy with a poisoning rate of 10%. Each class is treated as the target in turn, resulting in 200 backdoor models per attack and 200 benign models. We also evaluate the strongest improved baseline, MMBD$^+$. The average ACC across all models is 90.8%, with a standard deviation of 7.1%. The ASR is 87.1% on average, with a standard deviation of 23.3%.

Both MMBD$^+$ and SPD maintain very high TPRs with FPRs below 0.5% across all backdoor attacks on GTSRB. MMBD$^+$ achieves TPRs between 99.5% and 100.0%, while SPD consistently reaches 100.0% for every attack, showing that its robustness also holds on models trained from scratch. These results confirm that our modifications to the data-free detection framework are sound and not confined to the pre-trained setting, and they provide an additional sanity check that SPD behaves as expected in the conventional evaluation regime. Nevertheless, the main focus of this work is pre-trained models, where we observe that existing methods can be fragile. Prior studies have already reported excellent performance on models trained from scratch, so we do not further expand the evaluation in this setting.



(a) Benign (b) Class-Agnostic (c) Class-Specific

Figure 7: Heatmaps of convergence speeds for three scenarios: benign, class-agnostic backdoor, and class-specific backdoor. Each plot displays normalized convergence speeds ranging from 0 to 1, where the lighter color means a faster convergence speed. The x-axis and y-axis of the heatmaps represent classification labels and ten ResNet18 models, respectively. For class-specific cases, the source class is 2 for the first row, and 1 for all others.



(a) Backdoored & Scratch (b) Backdoored & Pre-Trained (c) Benign & Pre-Trained

Figure 8: Heatmaps of convergence speeds for a model trained from scratch and two models initialized with the pre-trained weights. Both backdoored models are injected with a class-agnostic backdoor and the target class is 1. The x-axis and y-axis of the heatmaps represent classification labels of CIFAR10 dataset and model layers of a ResNet18 model, respectively.

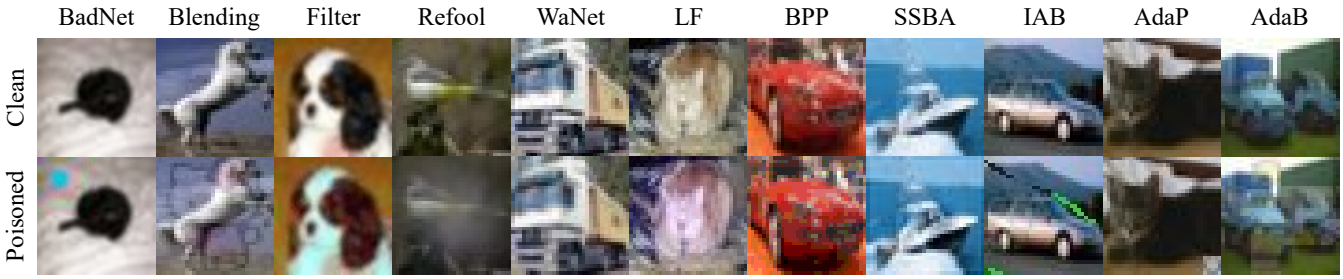


Figure 9: Comparison of clean and poisoned images with different triggers used in training.