SCALE-UP: AN EFFICIENT BLACK-BOX INPUT-LEVEL BACKDOOR DETECTION VIA ANALYZING SCALED PREDICTION CONSISTENCY

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks (DNNs) are vulnerable to backdoor attacks, where adversaries embed a hidden backdoor trigger during the training process for malicious prediction manipulation. These attacks pose great threats to the applications of DNNs under the real-world machine learning as a service (MLaaS) setting, where the deployed model is fully black-box while the users can only query and obtain its predictions. Currently, there are many existing defenses to reduce backdoor threats. However, almost all of them cannot be adopted in MLaaS scenarios since they require getting access to or even modifying the suspicious models. In this paper, we propose a simple yet effective black-box input-level backdoor detection, called SCALE-UP, which requires only the predicted labels to alleviate this problem. Specifically, we identify and filter malicious testing samples by analyzing their prediction consistency during the pixel-wise amplification process. Our defense is motivated by an intriguing observation (dubbed *scaled prediction consistency*) that the predictions of poisoned samples are significantly more consistent compared to those of benign ones when amplifying all pixel values. Besides, we also provide theoretical foundations to explain this phenomenon. Extensive experiments are conducted on benchmark datasets, verifying the effectiveness and efficiency of our defense and its resistance to potential adaptive attacks.

1 INTRODUCTION

Deep neural networks (DNNs) have been deployed in a wide range of mission-critical applications, such as autonomous driving (Kong et al., 2020; Grigorescu et al., 2020; Wen & Jo, 2022), face recognition (Sun et al., 2015; Wang & Deng, 2021; Fuad et al., 2021), and object detection (Zhao et al., 2019; Zou et al., 2019; Wang et al., 2021). In general, training state-of-the-art DNNs usually requires extensive computational resources and training samples. Accordingly, in real-world applications, developers and users may directly exploit third-party pre-trained DNNs instead of training their new models. This is what we called machine learning as a service (MLaaS).

However, recent studies (Gu et al., 2019; Goldblum et al., 2022; Li et al., 2022) revealed that DNNs can be compromised by embedding adversary-specified hidden backdoors during the training process, posing threatening security risks to MLaaS. The adversaries can activate embedded backdoors in the attacked models to maliciously manipulate their predictions whenever the pre-defined trigger pattern appears. Users are hard to identify these attacks under the MLaaS setting since attacked DNNs behave normally on benign samples.

To reduce backdoor threats, there are many different types of defenses, such as model repairing (Li et al., 2021c; Wu & Wang, 2021; Zeng et al., 2022), poison suppression (Du et al., 2020; Li et al., 2021b; Huang et al., 2022), and backdoor detection (Xiang et al., 2022; Liu et al., 2022; Guo et al., 2022). However, most of these defenses were designed under the white-box setting, requiring accessing or even modifying model weights. Accordingly, users cannot adopt them in MLaaS scenarios. Currently, there are also a few model-level (Huang et al., 2020; Dong et al., 2021; Guo et al., 2022) and input-level (Li et al., 2021d; Qiu et al., 2021; Gao et al., 2021) black-box backdoor defenses where users can only access final predictions. However, these defenses have some implicit assumptions of backdoor triggers ($e.g.$, a small static patch), leading to being easily

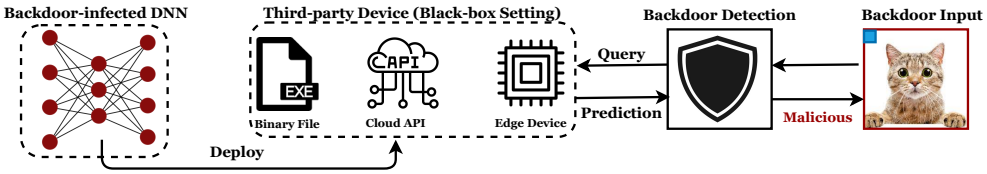


Figure 1: An illustration of the black-box input-level backdoor detection.

bypassed by advanced backdoor attacks (Nguyen & Tran, 2021; Li et al., 2021e). Their failures raise an intriguing question: *what are the fundamental differences between poisoned and benign samples that can be exploited to design universal black-box backdoor detection?*

In this paper, we focus on the *black-box input-level backdoor detection*, where we intend to identify whether a given suspicious input is malicious based on predictions of the deployed model (as shown in Fig. 1). This task is challenging since defenders have limited model information and no prior knowledge of the attack. Specifically, we first explore the pixel-wise amplification effects on benign and poisoned samples, motivated by the understanding that increasing trigger values does not hinder or even improve the attack success rate of attacked models (as preliminarily suggested in (Li et al., 2021d)). We demonstrate that the predictions of attacked images generated by both classical and advanced attacks are significantly more consistent compared to those of benign ones when amplifying all pixel values. We refer to this intriguing phenomenon as *scaled prediction consistency*. In particular, we also provide theoretical insights to explain this phenomenon. After that, based on these findings, we propose a simple yet effective method, dubbed scaled prediction consistency analysis (SCALE-UP), under both *data-free* and *data-limited* settings. Specifically, under the data-free setting, the SCALE-UP examines each suspicious sample by measuring its *scaled prediction consistency* (SPC) value, which is the proportion of labels of scaled images that are consistent with that of the input image. The larger the SPC value, the more likely this input is malicious. Under the data-limited setting, we assume that defenders have a few benign samples from each class, based on which we can reduce the side effects of class differences to further improve our SCALE-UP.

In conclusion, our main contributions are four-fold. **1)** We reveal an intriguing phenomenon ($i.e.$, scaled prediction consistency) that the predictions of attacked images are significantly more consistent compared to those of benign ones when amplifying all pixel values. **2)** We provide theoretical insights trying to explain the phenomenon of scaled prediction consistency. **3)** Based on our findings, we propose a simple yet effective black-box input-level backdoor detection (dubbed 'SCALE-UP') under both data-free and data-limited settings. **4)** We conduct extensive experiments on benchmark datasets, verifying the effectiveness of our method and its resistance to potential adaptive attacks.

2 Related Work

2.1 Backdoor Attack

Backdoor attacks (Li et al., 2022) compromise DNNs by contaminating the training process through injecting poisoned samples. These samples are crafted by adding adversary-specified trigger patterns into the selected benign samples. Backdoor attacks are stealthy since the attacked models behave normally on benign samples and the adversaries only need to craft a few poisoned samples. Accordingly, they introduce serious risks to DNN-based applications. In general, existing attacks can be roughly divided into two main categories based on the trigger property, including **1)** patch-based attacks and **2)** non-patch-based attacks, as follows:

Patch-based Backdoor Attacks. (Gu et al., 2019) proposed the first backdoor attack, which was called BadNets. Specifically, BadNets randomly selected and modified a few benign training samples by stamping the trigger patch and changing their label with a pre-defined target label. The generated poisoned samples associated with the remaining benign samples will be released to users to train their models. After that, (Chen et al., 2017) first discussed the attack stealthiness and introduced trigger transparency, where they suggested that poisoned images should be indistinguishable compared with their benign version to evade human inspection. (Turner et al., 2019) argued that making trigger patches invisible is not stealthy enough since the ground-truth labels of poisoned samples are different from the target label. They designed the first clean-label backdoor attack where adversaries can only poison samples from the target class. Recently, (Li et al., 2021d) pro-

posed the first physical backdoor attack, where the location and appearance of the trigger contained in the digitized test samples may be different from that of the one used for training.

Non-patch-based Backdoor Attacks. Different from classical attacks whose trigger pattern is a small patch, recent advanced methods exploited non-patch-based triggers trying to bypass backdoor defenses. For example, (Zhao et al., 2020) exploited full-image size targeted universal adversarial perturbation (Moosavi-Dezfooli et al., 2017) as the trigger pattern to design a more effective clean-label backdoor attack. (Nguyen & Tran, 2021) adopted image warping as the backdoor trigger, which deforms the whole image while preserving image content. Recently, inspired by the DNN-based image steganography (Tancik et al., 2020), (Li et al., 2021e) proposed the first poison-only sample-specific trigger patterns. This attack broke the fundamental assumption ($i.e.$, the trigger is sample-agnostic) of most existing defenses and therefore could easily bypass them.

2.2 Backdoor Defense

Currently, there are many backdoor defenses to alleviate backdoor threats. In general, existing methods can be roughly divided into two main categories based on the defender's capacities, including **1)** white-box defenses and **2)** black-box defenses, as follows:

White-box Backdoor Defenses. In these approaches, defenders need to obtain the source files of suspicious models. The most typical white-box defenses are model repairing, aiming at removing hidden backdoors in the attacked DNNs. For example, (Liu et al., 2018a; Wu & Wang, 2021) proposed to remove backdoors based on model pruning; (Li et al., 2021c; Xia et al., 2022) adopted model distillation to eliminate hidden backdoors. There are also other types of white-box defenses, such as poison suppression (Du et al., 2020; Li et al., 2021b; Huang et al., 2022) and trigger reversion (Wang et al., 2019; Hu et al., 2022; Tao et al., 2022). However, users cannot use them under the machine learning as a service (MLaaS) setting, where they can only obtain model predictions.

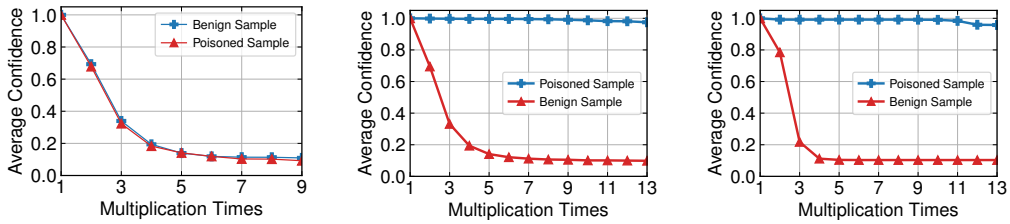
Black-box Backdoor Defenses. In these methods, defenders can only query the (deployed) model and obtain its predictions. Currently, there are two main types of black-box defenses, including **1)** model-level defenses (Huang et al., 2020; Dong et al., 2021; Guo et al., 2022) and **2)** input-level defenses (Li et al., 2021d; Qiu et al., 2021; Gao et al., 2021). Specifically, the former ones intend to identify whether the (deployed) suspicious model is attacked while the latter ones detect whether a given suspicious input is malicious. In this paper, we focus on the input-level black-box defense since it can serve as the 'firewall' helping to block and trace back malicious samples in MLaaS scenarios. However, as we will demonstrate in our experiments, existing input-level defenses can be easily bypassed by advanced backdoor attacks since they have some strong implicit assumptions of backdoor triggers. How to design effective black-box input-level backdoor detectors is still an important open question and worth further exploration.

3 The Phenomenon of Scaled Prediction Consistency

In this section, we explore the prediction behaviors of benign and poisoned samples generated by attacked DNNs since it is the cornerstone for designing black-box input-level backdoor defense. Before illustrating our key observations, we first review the general process of backdoor attacks.

The Main Pipeline of Backdoor Attacks. Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ represent a unmodified benign training set and $C : \mathcal{X} \to \mathcal{Y}$ is the deployed DNN, where $\boldsymbol{x}_i \in \mathcal{X} = [0,1]^{C\times W\times H}$ is the image, $y_i \in \mathcal{Y} = \{1, \dots, K\}$ is its label, and K is the number of different labels. The backdoor adversaries will select some benign samples ($i.e.$, $\mathcal{D}_s$) to generate their modified version by $\mathcal{D}_m = \{(\boldsymbol{x}', y_t) | \boldsymbol{x}' = \boldsymbol{x} + g(\boldsymbol{x}), (\boldsymbol{x}, y) \in \mathcal{D}_s\}$, where y_t is an adversary-specified target label and $g(\cdot)$ is a pre-defined poison generator. For example, $g(\boldsymbol{x}) = \boldsymbol{m} \odot (\boldsymbol{t} - \boldsymbol{x})$ in BadNets (Gu et al., 2019) and blended attack (Chen et al., 2017), where $\odot$ represents the element-wise product, $m \in [0,1]^{C\times W\times H}$ is a transparency mask, and $\boldsymbol{t} \in \mathcal{X}$ is the trigger pattern. Given N_b benign samples and N_p poisoned samples, the backdoor adversaries will train the attacked DNN $f(\cdot; \boldsymbol{\theta})$ based on the following optimization process (with loss $\mathcal{L}$):

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N_b} \mathcal{L}(f(\boldsymbol{x}_i; \boldsymbol{\theta}), y_i) + \sum_{j=1}^{N_p} \mathcal{L}(f(\boldsymbol{x}'_j; \boldsymbol{\theta}), y_t). \tag{1}$$



(a) Benign Model (b) BadNets (c) ISSBA

Figure 2: The average confidence ($i.e.$, average probabilities on the originally predicted label) of benign and poisoned samples $w.r.t.$ pixel-wise multiplications under benign and attacked models.

As preliminarily demonstrated in (Li et al., 2021d), increasing the pixel value of backdoor triggers does not hinder or even improve the attack success rate. However, defenders can not accurately manipulate these pixel values since they have no prior knowledge about trigger location. Accordingly, we explore what will happen if we scale up all pixel values of benign and poisoned images.

Settings. In this section, we adopt BadNets (Gu et al., 2019)) and ISSBA (Li et al., 2021e) as the example for our discussion. They are representative of patch-based and non-patch-based backdoor attacks, respectively. We conduct experiments on the CIFAR-10 dataset (Krizhevsky, 2009) with ResNet (He et al., 2016). For both attacks, we inject a large number of poisoned samples to ensure a high attack success rate ($\geq 99\%$). For each benign and poisoned image, we gradually enlarge its pixel values with multiplication. We calculate the *average confidence* defined as the average probabilities of samples on the originally predicted label. In particular, we select the label predicted upon the original sample as the originally predicted label for each varied sample and constrain all pixel values within $[0, 1]$ during the multiplication process. More details are in our appendix.

Results. As shown in Figure 2, the average confidence scores of both benign and poisoned samples decrease during the multiplication process under the benign model. In other words, the predictions of modified benign and poisoned samples are changed during this process. In contrast, poisoned and benign samples have significantly distinctive behaviors under attacked models. Specifically, the average confidence of benign samples decreases whereas that of poisoned samples is relatively stable with the increase of multiplication times. We call this phenomenon as *scaled prediction consistency*.

To further explain this intriguing phenomenon ($i.e.$, scaled prediction consistency), we exploit recent studies on neural tangent kernel (NTK) (Jacot et al., 2018) for analyzing the backdoor-infected models inspired by (Guo et al., 2022), as follows:

Theorem 1. *Suppose the poisoned training dataset consists of N_b benign samples and N_p poisoned samples, i.i.d. sampled from uniform distribution and belonging to K classes. Assume that deep neural network $f(\cdot;\theta)$ be a multivariate kernel regression (RBF kernel) with the objective in Eq. (1). For a given attacked sample $\boldsymbol{x}' = (\boldsymbol{1}-\boldsymbol{m})\odot\boldsymbol{x}+\boldsymbol{m}\odot\boldsymbol{t}$, we have: $\lim_{N_p\to N_b} C(n\cdot\boldsymbol{x}') = y_t, n \geq 1$.*

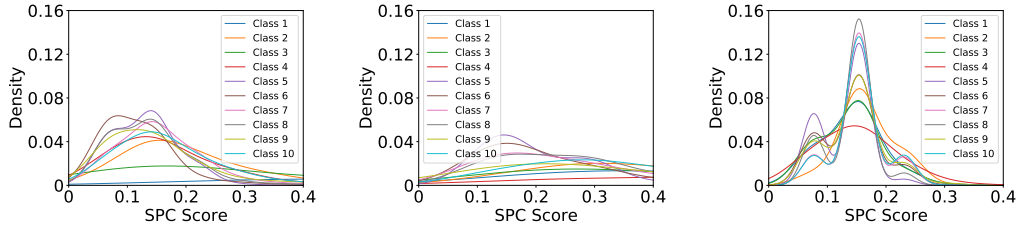
In general, Theorem 1 reveals that when the amount of poisoned samples closes to the benign samples or the attacked DNN over-fits the poisoned samples, it will still constantly predict the scaled attacked samples ($i.e.$, $n \cdot \boldsymbol{x}'$) as the target label y_t. Its proof can be found in Appendix A.

4 Scaled Prediction Consistency Analysis (SCALE-UP)

Motivated by the phenomenon of scaled prediction consistency demonstrated in the previous section, we propose a simple yet effective black-box input-level backdoor detection, dubbed scaled prediction consistency analysis (SCALE-UP), in this section.

4.1 Preliminaries

Defender's Goals. In general, defenders have two main goals, including *effectiveness* and *efficiency*. Effectiveness requires that the detection method can accurately identify whether a given image is malicious or not. Efficiency ensures that detection time is limited and therefore the deployed model can provide final results on time after the detection and prediction process.



(a) BadNets (b) Label-Consistent (c) ISSBA

Figure 3: The SPC scores of benign samples from different classes under attacked models.

Threat Model. We consider backdoor detection under the black-box setting in machine learning as a service (MLaaS) applications. Specifically, defenders can only query the third-party deployed model and obtain its predictions. They do not have any prior information about the backdoor attack or the model. In particular, we consider two data settings, including **1)** *data-free detection* and **2)** *data-limited detection*. The former one assumes that defenders have no holding benign samples, while the latter one allows defenders to have a few benign samples from each class. Note that we only assume to have the predicted label instead of the predicted probability vector in our method.

4.2 Data-free Scaled Prediction Consistency Analysis

As demonstrated in Section 3, we can use the average probability on the originally predicted label across its scaled images to determine whether a given suspicious image is malicious. In general, the larger the probability, the more likely the sample is poisoned. However, we can only obtain predicted labels while predicted probability vectors are inaccessible under our settings. Accordingly, we propose to examine whether the predictions of scaled samples are consistent, as follows:

Let $\mathcal{S}$ denotes a defender-specified scaling set ($e.g.$, $\mathcal{S} = \{3, 5, 7, 9, 11\}$). For a given input image $\boldsymbol{x}$ and the deployed classifier C, we define its *scaled prediction consistency (SPC)* as the proportion of labels of scaled images that are consistent with that of the input image, $i.e.$,

$$SPC(\boldsymbol{x}) = \frac{\sum_{n \in \mathcal{S}} \mathbb{I}\{C(n \cdot \boldsymbol{x}) = C(\boldsymbol{x})\}}{|\mathcal{S}|}, \tag{2}$$

where $\mathbb{I}$ is the indicator function and $|\mathcal{S}|$ denotes the size of scaling set $\mathcal{S}$. In particular, we constrain $n \cdot \boldsymbol{x} \in [0, 1]$ during the multiplication process.

Once we obtain the SPC value of suspicious input $\boldsymbol{x}$, we can determine it is malicious based on defender-specified threshold T. If $SPC(\boldsymbol{x}) > T$, we deem it as a backdoor sample.

4.3 Data-limited Scaled Prediction Consistency Analysis

In our data-free scaled prediction consistency analysis, we treat all labels equally. However, we notice that the SPC values of benign samples under attacked models are different across classes (as shown in Figure 3). In other words, some classes are more consistent against image scaling compared to the remaining ones. These benign samples with have high SPC values may be mistakenly treated as malicious samples, leading to relatively low precision of our method.

In data-limited scaled prediction consistency analysis, we assume that defenders have a few benign samples from each class. This setting has been widely used in existing backdoor defenses (Li et al., 2021c; Guo et al., 2022; Zeng et al., 2022). In this paper, we propose to exploit a set of statics summary ($i.e.$, mean μ and standard deviation σ) of these local benign samples to alleviate this problem, inspired by (Ioffe & Szegedy, 2015). We first estimate the statics summary for SPC values on samples from different classes, based on which to normalize the SPC value of suspicious input images according to their predicted labels. Specifically, for each class i, we calculate its corresponding mean μ_i and standard deviation σ_i based on samples $\boldsymbol{X}_i$ belonging to class i, as follows:

$$\mu_i = \mathbb{E}_{\boldsymbol{x} \in \boldsymbol{X}_i}[SPC(\boldsymbol{x})], \quad \sigma_i = \sqrt{\mathbb{E}_{\boldsymbol{x} \in \boldsymbol{X}_i}[(\boldsymbol{x} - \mu_i)^2]}. \tag{3}$$

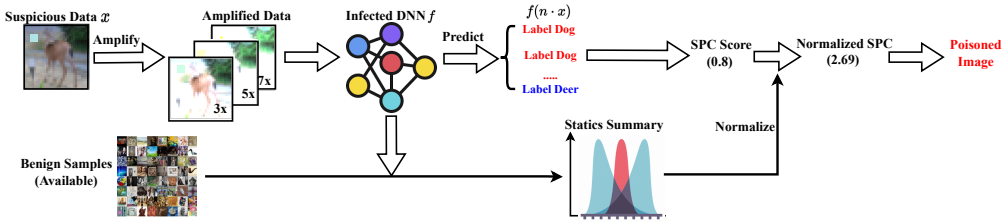


Figure 4: The main pipeline of our (data-limited) SCALE-UP. For each suspicious input, it first generates a set of its amplified images. After that, it takes amplified images to query the deployed DNN and obtain their predicted labels. Thirdly, we compute and normalize the SPC value based on the results and that of some local benign samples. SCALE-UP treats the input as a malicious image if the normalized SPC value is greater than a defender-specified threshold T.

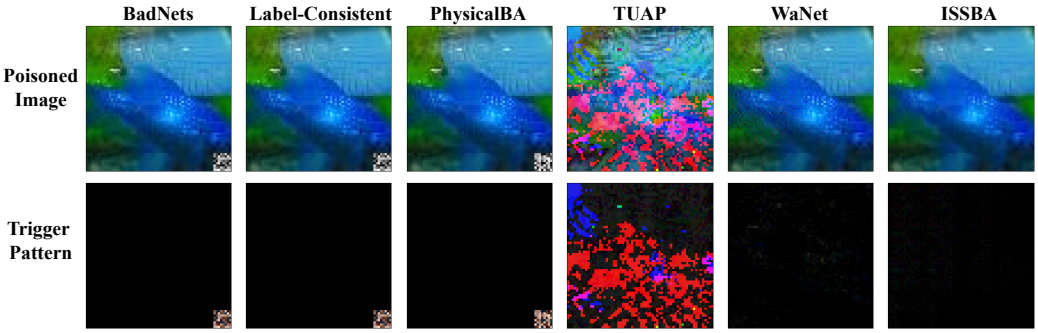


Figure 5: The demonstration of various trigger patterns of attacks used in our experiments.

In the detection process, given a suspicious image $\boldsymbol{x}$ and the deployed model C, we normalize the SPC value generated by data-free SCALE-UP based on its predicted label $\hat{y} \triangleq C(\boldsymbol{x})$, as follows:

$$NSPC(\boldsymbol{x}) \triangleq \frac{SPC(\boldsymbol{x}) - \mu_{\hat{y}}}{\sigma_{\hat{y}}}. \tag{4}$$

The main pipeline of our method is summarized in Figure 4.

5 Experiments

5.1 Main Settings

Dataset and DNN Selection. Following the settings in existing backdoor defenses, we conduct experiments on CIFAR-10 (Krizhevsky, 2009) and (Tiny) ImageNet (Russakovsky et al., 2015) datasets with ResNet (He et al., 2016). Please find more detailed information in our appendix.

Attack Baselines. In this paper, we evaluate our methods under six representative attacks, including **1)** BadNets (Gu et al., 2019), **2)** label consistent backdoor attack (dubbed 'Label-Consistent') (Turner et al., 2019), **3)** physical backdoor attack (dubbed 'PhysicalBA') (Li et al., 2021d), **4)** clean-label backdoor attack with targeted universal adversarial perturbation (dubbed 'TUAP') (Zhao et al., 2020), **5)** WaNet (Nguyen & Tran, 2021), and **6)** ISSBA (Li et al., 2021e). They are the representative of patch-based and non-patch-based backdoor attacks under different settings. For each attack, we randomly select the target label and inject sufficient poisoned samples to ensure the attack success rate $\geq 98\%$ while preserving the overall model performance. We implement these attacks based on the open-sourced backdoor toolbox[1]. We demonstrate the trigger patterns of adopted attacks in Figure 5. More detailed attack settings are in the appendix.

[1] https://github.com/THUYimingLi/BackdoorBox

Table 1: The performance (AUROC) on the Tiny CIFAR-10 dataset. Among all different methods, the best result is marked in boldface while the value with underline denotes the second-best result. The failed cases ($i.e.$, AUROC < 0.55) are marked in red. Note that STRIP require obtaining predicted probability vectors while other methods only need the predicted labels.

Attack→ Defense↓	BadNets	Label-Consistent	PhysicalBA	TUAP	WaNet	ISSBA	**Average**
STRIP	**0.989**	0.941	**0.971**	0.671	0.475	0.498	0.758
ShrinkPad	0.951	**0.957**	0.631	**0.869**	0.531	0.513	0.742
DeepSweep	0.967	0.921	0.946	0.743	0.506	0.729	0.802
Frequency	0.891	0.889	0.881	0.851	0.461	0.497	0.745
Ours (data-free)	0.971	0.947	0.969	0.816	0.918	**0.945**	0.928
Ours (data-limited)	0.971	0.954	0.970	0.830	**0.925**	**0.945**	**0.933**

Table 2: The performance (AUROC) on the Tiny ImageNet dataset. Among all different methods, the best result is marked in boldface while the value with underline denotes the second-best result. The failed cases ($i.e.$, AUROC < 0.55) are marked in red. Note that STRIP require obtaining predicted probability vectors while other methods only need the predicted labels.

Attack→ Defense↓	BadNets	Label-Consistent	PhysicalBA	TUAP	WaNet	ISSBA	**Average**
STRIP	**0.959**	**0.939**	**0.959**	0.638	0.501	0.471	0.745
ShrinkPad	0.871	0.938	0.672	**0.866**	0.498	0.492	0.737
DeepSweep	0.951	0.930	0.939	0.759	0.503	0.714	0.799
Frequency	0.864	0.859	0.864	0.837	0.428	0.540	0.732
Ours (data-free)	0.936	0.904	0.939	0.763	0.943	0.948	0.905
Ours (data-limited)	0.947	0.911	0.939	0.840	**0.951**	**0.949**	**0.922**

Defense Baselines. In this paper, we focus on the backdoor detection under the black-box setting where defenders can only query the deployed model and obtain its predicted label. Accordingly, we compare our methods to ShrinkPad (Li et al., 2021d), DeepSweep (Qiu et al., 2021), and artifacts detection in the frequency domain (dubbed 'Frequency') (Zeng et al., 2021). We also compare our methods to STRIP (Gao et al., 2021) that requires additional requirement ($i.e.$, obtaining predict probability vectors). We assume that defenders have 100 benign samples per class under the data-limited setting. Please find more defense details in our appendix.

Settings for Evaluation Datasets. Following the previous work (Lee et al., 2018), we use a positive ($i.e.$, attacked) and a negative ($i.e.$, benign) dataset to evaluate each defense. Specifically, the positive dataset contains the attacked testset and its augmented version, while the negative dataset contains a benign testset and its augmented version. The augmented datasets are created by adding small random noise to their original version. The noise magnitude is set to 0.05. In particular, adding these random noises will not significantly affect the attack success rate and the benign accuracy of deployed models. The introduction of the augmented datasets is to prevent evaluated defenses from over-fitting the benign or the poisoned testsets.

Evaluation Metrics. Following existing detection-based backdoor defenses (Gao et al., 2021; Guo et al., 2022), we adopt the area under receiver operating curve (AUROC) (Fawcett, 2006) to evaluate defense effectiveness, while use the inference time for evaluating efficiency. In general, the higher the AUROC and the lower the inference time, the better the backdoor detection.

5.2 Main Results

As shown in Table 1-2, *all baseline detection methods fail in defending against some evaluated attacks*. Specifically, they have relatively low AUROC in detecting advanced non-patch-based attacks ($i.e.$, WaNet and ISSBA). This failure is mostly because these defenses have some implicit assumptions ($e.g.$, the trigger pattern is sample-agnostic or static) about the attack, which are not necessarily true in practice. In contrast, *our methods reach promising performance in all cases on both datasets*. For example, the AUROC of our data-limited SCALE-UP is 0.5 greater than all baseline defenses in detecting WaNet on the Tiny ImageNet dataset. Even under the classical patch-based attacks ($i.e.$, BadNets, Label-Consistent, and PhysicalBA), the effectiveness of our methods is on par with or

better than all baseline defenses. Our methods are even better than STRIP, which requires obtaining predicted probability vectors instead of predicted labels. These results verify our effectiveness.

Besides, we also calculate the inference time of all defenses under the same platform. In particular, we only report the inference time of our SCALE-UP under the data-limited setting, since our detection has very similar running times. As shown in Figure 6, our method has fewer inference times compared to almost all baseline defenses under both settings. The only exception is ShrinkPad whereas its effectiveness is significantly lower compared to our method. Our detection is approximately 5% slower compared with the standard inference process without any defense. These results show the efficiency of our SCALE-UP detection.

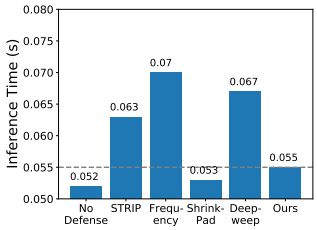


Figure 6: The inference time on the CIFAR-10 dataset.

5.3 Discussion

In this section, we discuss whether our method is still effective under different (adversarial) settings.

5.3.1 Defending against Attacks with Larger Trigger Sizes

Recent studies (Qiu et al., 2021) revealed that some defenses ($e.g.$, ShrinkPad) may fail in detecting samples with a relatively large trigger size. In this part, we use two patch-based attacks ($e.g.$, BadNets and PhysicalBA) on the Tiny ImageNet dataset for discussion. As shown in Figure 7(a), our methods have high AUROC values (> 0.93) across different trigger sizes under both data-free and data-limited settings, although there are some mild fluctuations. These results verify the resistance of our SCALE-UP detection methods.

5.3.2 The Resistance to Potential Adaptive Attacks

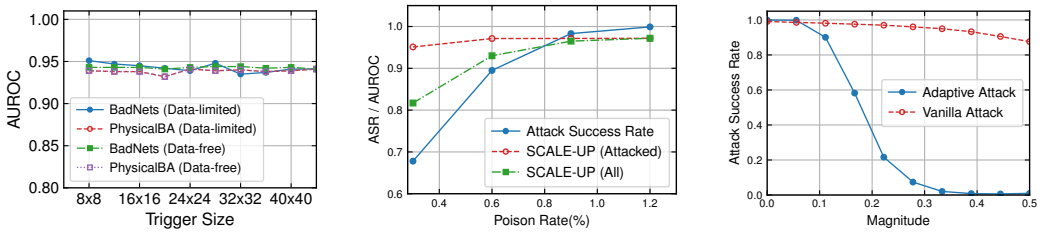
Most recently, (Qi et al., 2022) demonstrated that reducing the poisoning rate is a simple yet effective method to design adaptive attacks for detection-based defenses, since it can reduce the differences between benign and poisoned samples. Motivated by this finding, we first explore whether our SCALE-UP methods are still effective in defending against attacks with low poisoning rates. We adopt BadNets on the CIFAR-10 dataset as an example for our discussions. In particular, we report the results of all poisoned testing samples and those that can be predicted as the target label, respectively. We design this setting since attacked models may still correctly predict many poisoned samples even if they contain trigger patterns when the poisoning rate is relatively low.

As shown in Figure 7(b), the attack success rate (ASR) increases with the increase of the poisoning rate. Our method can still correctly detect poisoned samples that can successfully attack the deployed model even when the poisoning rate is set to 0.4% where the ASR is lower than 70%. In these cases, the AUROC > 0.95. Besides, our method can still reach promising performance (AUROC > 0.8) in detecting all poisoned samples. These results verify the resistance of our defense to adaptive attacks with low poisoning rates, where attacked models don't over-fit backdoor triggers.

To further evaluate the resistance of our SCALE-UP, we evaluate it under the worst scenario, where the backdoor adversaries are fully aware of our mechanism. Specifically, we design a strong adaptive attack by introducing an additional defense-resistant regularization term to the vanilla attack illustrated in Eq. (1). This regularization term is used to prevent scaled poisoned samples $n \cdot \boldsymbol{x}'_j$ being predicted as the target label y_t, as follows:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N_b} \mathcal{L}(f(\boldsymbol{x}_i;\boldsymbol{\theta}), y_i) + \sum_{j=1}^{N_p} \mathcal{L}(f(\boldsymbol{x}'_j;\boldsymbol{\theta}), y_t) + \sum_{j=1}^{N_p} \mathcal{L}(f(n \cdot \boldsymbol{x}'_j;\boldsymbol{\theta}), y_j). \tag{5}$$

Similar to previous experiments, we adopt BadNets to design the adaptive attack on the CIFAR-10 dataset. As we expected, this method can bypass our detection resulting in a low AUROC ($i.e.$, 0.467). However, *the adaptive attack would make the poisoned samples significantly more vulnerable to small random Gaussian noises*. As shown in Figure 7(c), random noises with a small magnitude (< 0.3) will significantly reduce the attack success rate of the adaptive attack, while having minor adverse effects on the vanilla attack. In other words, defenders can easily adopt random



(a) (b) (c)

Figure 7: The results of additional experiments in our discussion. **(a)** The performance of our methods under attacks with different trigger sizes. **(b)** The attack performance and the defense effectiveness on all poisoned testing samples and those that can successfully attack the deployed model. **(c)** The effectiveness of adaptive and vanilla backdoor attacks on poisoned samples with random noise under different magnitudes.

Table 3: The performance (AUROC) of SCALE-UP variants with random noises on CIFAR-10 and Tiny-ImageNet datasets. The failed cases ($i.e.$, AUROC < 0.55) are marked in red.

Dataset↓	Attack→ Setting↓	BadNets	Label Consistent	PhysicalBA	TUAP	WaNet	ISSBA	**Average**
CIFAR-10	data-free	0.939	0.816	0.976	0.698	0.497	0.421	0.724
	data-limited	0.939	0.873	0.981	0.706	0.432	0.444	0.729
Tiny ImageNet	data-free	0.951	0.711	0.899	0.632	0.531	0.467	0.706
	data-limited	0.951	0.761	0.899	0.644	0.534	0.501	0.706

noises to defend against this adaptive method. We speculate that its vulnerability is mostly because the regularization term significantly constrains the generalization of attacked DNNs on the poisoned samples. We will further explore its intrinsic mechanism in our future work.

5.3.3 The Effectiveness of Scaling Process

Technically, the scaling process in our SCALE-UP detection can be regarded as a data augmentation method generating different modified versions of the suspicious input image. It naturally raises an intriguing question: *If other augmentation methods are adopted, is our SCALE-UP detection still effective?* Since flip operations and frequency domain analysis have been adopted in (Li et al., 2021d; Zeng et al., 2021) for defense and proved to have minor benefits to detecting advanced backdoor attacks (Li et al., 2021e; Nguyen & Tran, 2021), we investigate the effectiveness of adding increasing magnitudes of random noise. Specifically, we design the SCALE-UP variant by replacing the scaling process with adding the same varied magnitudes of random noise to the given inputs. As shown in Table 3, using random noises is far less effective compared to the standard SCALE-UP methods, especially in detecting advanced attacks ($i.e.$, WaNet and ISSBA). We speculate that it is mostly because they adopted invisible full-image size trigger patterns and therefore the trigger-related features are less robust. Although we currently fail to provide theoretical analysis for the aforementioned phenomena, at least they verify the effectiveness of our scaling process. We will further discuss it in our future work.

6 Conclusion

In this paper, we proposed a simple yet effective black-box input-level backdoor detection (dubbed SCALE-UP) that can be used in real-world applications under the machine learning as a service (MLaaS) setting. Our method was motivated by an intriguing new phenomenon (dubbed *scaled prediction consistency*) that the predictions of poisoned samples are significantly more consistent compared to those of benign ones when amplifying all pixel values. We also provided theoretical foundations to explain this phenomenon. In particular, we designed our SCALE-UP detection method under both data-free and data-limited settings. Extensive experiments on benchmark datasets verified the effectiveness and efficiency of our method and its resistance to potential adaptive attacks.

ETHICS STATEMENT

DNNs have been widely and successfully adopted in many mission-critical applications. Accordingly, their security is of great significance. The existence of backdoor threats raises serious concerns about using third-party models under the machine learning as a service (MLaaS) setting. In this paper, we propose a simple yet effective black-box input-level backdoor detection. Accordingly, this work has no ethical issues since it does not reveal any new security risks and is purely defensive. However, we need to notice that our methods can only be used to filter poisoned testing samples whereas they do not reduce the intrinsic backdoor vulnerability of deployed models. People should not be too optimistic about eliminating backdoor threats.

REPRODUCIBILITY STATEMENT

We have provided detailed information on datasets, models, training settings, and computational facilities in our main manuscript and appendix. The codes for reproducing our main experiments are also included in the supplementary materials.

REFERENCES

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.

Yinpeng Dong, Xiao Yang, Zhijie Deng, Tianyu Pang, Zihao Xiao, Hang Su, and Jun Zhu. Black-box detection of backdoor attacks with limited information and data. In *ICCV*, 2021.

Min Du, Ruoxi Jia, and Dawn Song. Robust anomaly detection and backdoor attack detection via differential privacy. In *ICLR*, 2020.

Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

Md Tahmid Hasan Fuad, Awal Ahmed Fime, Delowar Sikder, Md Akil Raihan Iftee, Jakaria Rabbi, Mabrook S Al-Rakhami, Abdu Gumaei, Ovishake Sen, Mohtasim Fuad, and Md Nazrul Islam. Recent advances in deep learning techniques for face recognition. *IEEE Access*, 9:99112–99142, 2021.

Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *ACSAC*, 2019.

Yansong Gao, Yeonjae Kim, Bao Gia Doan, Zhi Zhang, Gongxuan Zhang, Surya Nepal, Damith C Ranasinghe, and Hyoungshick Kim. Design and evaluation of a multi-domain trojan detection method on deep neural networks. *IEEE Transactions on Dependable and Secure Computing*, 19 (4):2349–2364, 2021.

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

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

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

Junfeng Guo, Ang Li, and Cong Liu. AEVA: Black-box backdoor detection using adversarial extreme value analysis. In *ICLR*, 2022.

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

Xiaoling Hu, Xiao Lin, Michael Cogswell, Yi Yao, Susmit Jha, and Chao Chen. Trigger hunting with a topological prior for trojan detection. In *ICLR*, 2022.

Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor defense via decoupling the training process. In *ICLR*, 2022.

Shanjiaoyang Huang, Weiqi Peng, Zhiwei Jia, and Zhuowen Tu. One-pixel signature: Characterizing cnn models for backdoor detection. In *ECCV*, 2020.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, 2018.

Zelun Kong, Junfeng Guo, Ang Li, and Cong Liu. Physgan: Generating physical-world-resilient adversarial examples for autonomous driving. In *CVPR*, 2020.

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

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018.

Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 2021a.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In *NeurIPS*, 2021b.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *ICLR*, 2021c.

Yiming Li, Tongqing Zhai, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor attack in the physical world. In *ICLR Workshop*, 2021d.

Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

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

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

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

Yingqi Liu, Guangyu Shen, Guanhong Tao, Zhenting Wang, Shiqing Ma, and Xiangyu Zhang. Complex backdoor detection by symmetric feature differencing. In *CVPR*, 2022.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, 2017.

Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *ICLR*, 2021.

Xiangyu Qi, Tinghao Xie, Saeed Mahloujifar, and Prateek Mittal. Circumventing backdoor defenses that are based on latent separability. *arXiv preprint arXiv:2205.13613*, 2022.

Han Qiu, Yi Zeng, Shangwei Guo, Tianwei Zhang, Meikang Qiu, and Bhavani Thuraisingham. Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation. In *AsiaCCS*, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Yi Sun, Ding Liang, Xiaogang Wang, and Xiaoou Tang. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015.

Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *CVPR*, 2020.

Guanhong Tao, Guangyu Shen, Yingqi Liu, Shengwei An, Qiuling Xu, Shiqing Ma, Pan Li, and Xiangyu Zhang. Better trigger inversion optimization in backdoor scanning. In *CVPR*, 2022.

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

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE S&P*, 2019.

Mei Wang and Weihong Deng. Deep face recognition: A survey. *Neurocomputing*, 429:215–244, 2021.

Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, Haibin Ling, and Ruigang Yang. Salient object detection in the deep learning era: An in-depth survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):3239–3259, 2021.

Li-Hua Wen and Kang-Hyun Jo. Deep learning-based perception systems for autonomous driving: A comprehensive survey. *Neurocomputing*, 2022.

Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *NeurIPS*, 2021.

Jun Xia, Ting Wang, Jieping Ding, Xian Wei, and Mingsong Chen. Eliminating backdoor triggers for deep neural networks using attention relation graph distillation. In *IJCAI*, 2022.

Zhen Xiang, David J Miller, and George Kesidis. Post-training detection of backdoor attacks for two-class and multi-attack scenarios. In *ICLR*, 2022.

Yi Zeng, Won Park, Zhuoqing Morley Mao, and R. Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *ICCV*, 2021.

Yi Zeng, Si Chen, Won Park, Z Morley Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *ICLR*, 2022.

Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *CVPR*, 2020.

Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 2019.

Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.

APPENDIX

A THE OMITTED PROOF OF THEOREM 1

Theorem 1. *Suppose the poisoned training dataset consists of N_b benign samples and N_p poisoned samples, i.i.d. sampled from uniform distribution and belonging to K classes. Assume that deep neural network f adopt RBF kernel and cross-entropy loss with the optimization objective in Eq.1. For a given attacked sample $\boldsymbol{x}' = (\boldsymbol{1}-\boldsymbol{m})\odot\boldsymbol{x}+\boldsymbol{m}\odot\boldsymbol{t}$, we have: $\lim_{N_p\to N_b} C(n\cdot\boldsymbol{x}') = y_t, n \geq 1$.*

Proof of Theorem 1: Following (Guo et al., 2022), we have the regression solution for NTK is:

$$\phi_t(\cdot) = \frac{\sum_{i=1}^{N_b} K(\cdot, \boldsymbol{x_i}) \cdot y_i + \sum_{i=1}^{N_p} K(\cdot, \boldsymbol{x'_i}) \cdot y_t}{\sum_{i=1}^{N_b} K(\cdot, \boldsymbol{x_i}) + \sum_{i=1}^{N_p} K(\cdot, \boldsymbol{x'_i})}, \tag{6}$$

where $\phi_t(\cdot) \in \mathbb{R}$ is the predictive probability output of $f(\cdot;\theta)$ for the target class t and y_i is the corresponding one-hot label. $K(\boldsymbol{x}, \boldsymbol{x_i}) = e^{-2\gamma||\boldsymbol{x}-\boldsymbol{x_i}||^2}$ $(\gamma > 0)$. Since the training samples are evenly distributed, there are $\frac{N_b}{k}$ benign samples belonging to y_t. Without loss of generality, we assume the target label $y_t = 1$ while others are 0. Then the regression solution can be converted to:

$$\phi_t(\cdot) = \frac{\sum_{i=1}^{N_b/k} K(\cdot, \boldsymbol{x_i}) + \sum_{i=1}^{N_p} K(\cdot, \boldsymbol{x'_i})}{\sum_{i=1}^{N_b} K(\cdot, \boldsymbol{x_i}) + \sum_{i=1}^{N_p} K(\cdot, \boldsymbol{x'_i})}, \tag{7}$$

For a given backdoored sample $\boldsymbol{x'} = (1-m)\odot\boldsymbol{x} + m\odot\boldsymbol{t}$, we can simplify Eq. (7) as:

$$\phi_t(\boldsymbol{x'}) \geq \frac{\sum_{i=1}^{N_p} K(\boldsymbol{x'}, \boldsymbol{x'_i})}{\sum_{i=1}^{N_b} K(\boldsymbol{x'}, \boldsymbol{x_i}) + \sum_{i=1}^{N_p} K(\boldsymbol{x'}, \boldsymbol{x'_i})}, \tag{8}$$

we here remove the term $\sum_{i=1}^{N_b/k} K(\boldsymbol{x'}, \boldsymbol{x_i})$. This is because $\boldsymbol{x'}$ typically don't belong to the target y_t and $\sum_{i=1}^{N_b/k} K(\boldsymbol{x'}, \boldsymbol{x_i}) << \sum_{i=1}^{N_p} K(\boldsymbol{x'}, \boldsymbol{x'_i})$ otherwise the attacker has no incentive to craft backdoored sample.

When N_p close to N_b, which implies that the poisoning rate close to 50%, the attacker can achieve the optimal attack efficacy (Liu et al., 2018b; Gu et al., 2019; Li et al., 2021a). Given $K(\boldsymbol{x}, \boldsymbol{x_i}) = e^{-2\gamma||\boldsymbol{x}-\boldsymbol{x_i}||^2}$ $(\gamma > 0)$, if $N_p = N_b$, we have:

$$\phi_t(n\cdot\boldsymbol{x'}) = \frac{\sum_{i=1}^{N_p} K(n\cdot\boldsymbol{x'}, \boldsymbol{x'_i})}{\sum_{i=1}^{N_b} K(n\cdot\boldsymbol{x'}, \boldsymbol{x_i}) + \sum_{i=1}^{N_p} K(n\cdot\boldsymbol{x'}, \boldsymbol{x'_i})}, \tag{9}$$

If $n = 1$, we can easily obtain that:

$$\sum_{i=1}^{N_p} e^{-2\gamma||(1-m)\odot(\boldsymbol{x}-\boldsymbol{x_i})||^2} - e^{-2\gamma||(1-m)\odot(\boldsymbol{x}-\boldsymbol{x_i})+m\odot(\boldsymbol{t}-\boldsymbol{x_i})||^2} \tag{10}$$

$$= \sum_{i=1}^{N_p} e^{-2\gamma||(1-m)\odot(\boldsymbol{x}-\boldsymbol{x_i})||^2}(1 - e^{-2\gamma||m\odot(\boldsymbol{t}-\boldsymbol{x_i})||^2}) > 0, \tag{11}$$

Since the internal term $(1 - e^{-2\gamma||m\odot(\boldsymbol{t}-\boldsymbol{x_i})||^2})$ can be always larger than 0, thus it is clear that $f(\boldsymbol{x'}) = y_t$, which is also consistent with the practice.

However, when $n > 1$, to compare $\sum_{i=1}^{N_p} K(n \cdot \boldsymbol{x'}, \boldsymbol{x'_i})$ and $\sum_{i=1}^{N_b} K(n \cdot \boldsymbol{x'}, \boldsymbol{x_i})$, we have:

$$\sum_{i=1}^{N_o} K(n \cdot \boldsymbol{x'}, \boldsymbol{x'_i}) - \sum_{i=1}^{N_b} K(n \cdot \boldsymbol{x'}, \boldsymbol{x_i}) \tag{12}$$

$$= \sum_{i=1}^{N_b} e^{-2\gamma||(1-m)\odot(n\cdot\boldsymbol{x}-\boldsymbol{x_i})+m\odot(n-1)\boldsymbol{t}||^2} - e^{-2\gamma||(1-m)\odot(n\cdot\boldsymbol{x}-\boldsymbol{x_i})+m\odot(n\cdot\boldsymbol{t}-\boldsymbol{x_i})||^2} \tag{13}$$

$$= \sum_{i=1}^{N_b} e^{-2\gamma||(1-m)\odot(n\cdot\boldsymbol{x}-\boldsymbol{x_i})+m\odot(n-1)\boldsymbol{t}||^2}(1 - e^{-2\gamma(||m\odot(n\cdot\boldsymbol{t}-\boldsymbol{x_i})||^2-||m\odot(n-1)\boldsymbol{t})||^2)}), \tag{14}$$

$$\tag{15}$$

Regarding the internal term $||m \odot (n \cdot \boldsymbol{t} - \boldsymbol{x_i})||^2 - ||m \odot (n-1)\boldsymbol{t})||^2$, we have:

$$||m \odot (n \cdot \boldsymbol{t} - \boldsymbol{x_i})||^2 - ||m \odot (n-1)\boldsymbol{t})||^2 \tag{16}$$

$$= \sum_{j,k\in\ trigger} ||(n-1)\cdot \boldsymbol{t}_{j,k} + (\boldsymbol{t}_{j,k} - \boldsymbol{x_{i,j,k}})||^2 - ||(n-1)\cdot \boldsymbol{t}_{j,k}||^2 \tag{17}$$

$$= \sum_{j,k\in\ trigger} \delta_{i,j,k}^2 + 2(n-1)\cdot \delta_{i,j,k}\boldsymbol{t}_{j,k}, \tag{18}$$

where $\delta_{i,j,k}$ is the pixel-level residue between the trigger and benign samples. We assume that the $\delta_{i,j,k}\boldsymbol{t}_{j,k}$ close to a zero mean among inputs, thus we can rewrite Eq.(8) as follows:

$$\sum_{i=1}^{N_p} K(n \cdot \boldsymbol{x'}, \boldsymbol{x'_i}) - \sum_{i=1}^{N_b} K(n \cdot \boldsymbol{x'}, \boldsymbol{x_i}) \tag{19}$$

$$\approx \sum_{i=1}^{N_p} e^{-2\gamma||(1-m)\odot(n\cdot\boldsymbol{x}-\boldsymbol{x_i})+m\odot(n-1)\boldsymbol{t}||^2}(1 - e^{-2\gamma\sum_{j,k\in trigger}\delta_{i,j,k}^2}) \tag{20}$$

$$> 0. \tag{21}$$

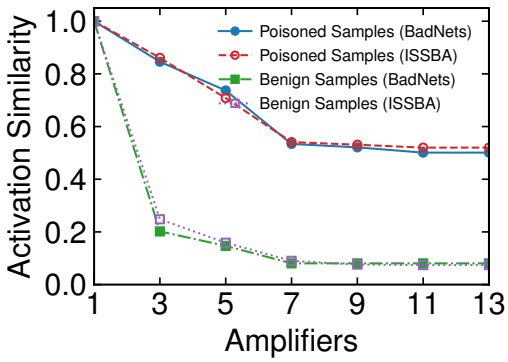
Put Eq. (19) and Eq. (9) together, we know that $\phi_t(n \cdot \boldsymbol{x'}) \geq 0.5$, as $N_p \to N_b$, we have:

$$\lim_{N_p \to N_b} C(n \cdot \boldsymbol{x'_t}) = y_t, n \geq 1.$$

□

B THE DETAILED CONFIGURATIONS OF THE EMPIRICAL STUDY

We adopt BadNets (Gu et al., 2019)) and ISSBA (Li et al., 2021e) as the example for our discussion. They are the representative of patch-based and non-patch-based backdoor attacks, respectively. We conduct experiments on the CIFAR-10 dataset (Krizhevsky, 2009) with ResNet-34 (He et al., 2016). For both attacks, we inject a large number of poisoned samples to ensure a high attack success rate ($\geq 99\%$). For each benign and poisoned image, we gradually enlarge its pixel values with multiplication. We calculate the *averaged confidence* defined as the averaged probabilities of samples on the originally predicted label. In particular, we select the label predicted upon the original sample as the originally predicted label for each varied sample and constrain all pixel values within $[0, 1]$ during the multiplication process. More detailed settings are in our appendix. We follow previous work (Gu et al., 2019; Li et al., 2021e) to implement the backdoor attacks. The trigger for BadNets is a 4×4 square consisting of random pixel values; the trigger of ISSBA is dynamically generated via Generate Adversary Network (Goodfellow et al., 2020). Both attacks are implemented via `BackdoorBox` open-sourced tool (Li et al., 2022), and we set poisoning rates $\geq 10\%$ for ensuring the attack efficacy with an attack success rate (ASR $\geq 99\%$).



(a) Similarity Score

Figure 8: The similarity scores for neuron activation computed on the permute layer

Table 4: Detailed information about dataset for each task.

Task	# Labels	Input size	# Training images
CIFAR-10	10	32x32	50000
TinyImageNet	200	64x64	1,000,000

C The Neuron Activation Similarity for Benign and Poisoned Samples During the Multiplication Process

We observed the neuron activation similarity for backdoor inputs perform rather consistently compared with the benign samples against the amplification process under the infected model. As for the benign model, we find that the benign model performs similarly for the benign and poisoned samples, for which the prediction confidence drops drastically.

D The Details for Implementing Training Evaluated Models

We train backdoor infected models using BackdoorBox (Li et al., 2022). We set the training epoch as 200 and the poisoning rate as $5 - 10\%$ for each attack to ensure the attack success rate. We also involve data augmentation in our training process. For each infected model, we randomly select infected labels to ensure their predictions on benign inputs are similar to the benign models, which ensures the stealth of backdoor attacks. Regarding, the data-limited scenario, we randomly selected 100 samples from a held-out dataset.

The details for each task are included in Tab. 4. The data augmentation utilized for building each model is following the official codes[2].

D.1 The Accuracy and Attack Success Rate (ASR) for Evaluated Models

The accuracy and ASR for the evaluated models for each task in included in Table 5.

[2] https://keras.io/zh/examples/cifar10_resnet/

Table 5: Accuracy and ASR for the evaluated models for each task

Task↓ Model→	Infected Model		Normal Model Accuracy
	Accuracy	ASR	
CIFAR-10	$\geq 90.04\%$	$\geq 97.7\%$	$\geq 92.31\%$
Tiny ImageNet	$\geq 36.98\%$	$\geq 97.22\%$	$\geq 40.11\%$

Table 6: The performance of six defense baselines against partial backdoor attacks

Task↓ Attack →	STRIP	ShrinkPad	Frequency	DeepSweep	Ours (data-free)	Ours (data-limited)
CIFAR-10	0.617	0.949	0.891	0.967	0.971	0.971
Tiny ImageNet	0.601	0.868	0.861	0.951	0.936	0.971

E The Detailed Configurations for Implementing Each Defense Baseline

- **STRIP:** We implement STRIP following their official open-sourced codes[3].
- **ShrinkPad:** We implement ShrinkPad following their official open-sourced codes[4].
- **Frequeny:** We implement Frequency approach following their official codes[5].
- **DeepSweep:** We manually implement DeepSweep using `Scipy` package to remove the high-frequency noise and use `torchvision.transforms` and `keras.preprocess` packages to conduct transformation to inputs. Notably, we don't apply `finetune` process within DeepSweep since we only focus on the black-box backdoor detection scenarios.

F Detailed Description For Each Metric

- The Receiver Operating Curve (ROC) shows the trade-off between detection the success rate for poisoned samples and detection error rate for benign samples across different decision thresholds T under infected-DNNs.
- Inference Time: we implement each approach under the platform with one NVIDIA GPU 1080 Ti and a Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz with batch size $= 1$. We test the inference time of each approach with an average of 1,000 runs.

G Settings For Measuring the Inference Time

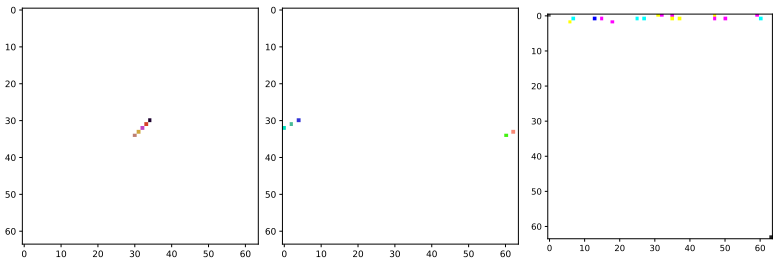
Since we focus on defending against backdoor attacks in the inference phase, we here measure the inference time by:

- Identifying whether the input sample is poisoned or not.
- If the input is a benign sample, we next should use the target model to predict it.

For STRIP, ShrinkPad, DeepSweep, and SCALE-UP (ours) these approaches, we leverage the target model's prediction on the (augmented) inputs for defense purpose, which means the input can be identified and predicted at the same time. As for Frequency, which leverages a secondary neural network to predict the frequency domain of each given input. However, if the input is identified as benign, the target DNNs should also deliver prediction on it. We here assume the benign and poisoned samples have equal possibilities. Therefore, we calculate the inference time for Frequency approach following:

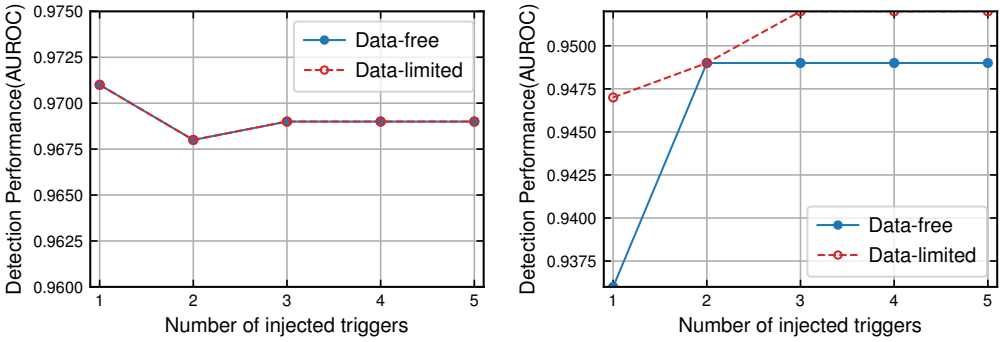
$$\text{time} = \text{TIME}(\text{Frequency}(\text{input})) + 0.5 \cdot \text{TIME}(\text{DNN}(\text{input})). \tag{22}$$

[3] https://github.com/garrisongys/STRIP.git
[4] https://github.com/THUYimingLi/BackdoorBox.git
[5] https://github.com/YiZeng623/frequency-backdoor.git



(a) Trigger I (b) Trigger II (c) Trigger III

Figure 9: Demonstrations for dynamic triggers.



(a) CIFAR-10 (b) TinyImageNet

Figure 10: The average results for multiple triggers within a single label.

While for other approaches, we measure their inference time via:

$$\text{time} = \text{TIME}(\text{DNN}((\text{Augumented}) \text{ INPUT})). \tag{23}$$

H Performance Under Multiple-Backdoor Triggers Within a Single Infected Label

Consistent with (Guo et al., 2022; Wang et al., 2019), we also evaluate the efficacy of SCALE-UP under a rather practical scenario where multiple backdoors are embedded within a single infected label. We randomly select a label as the infected label and inject various types of poisoned samples in the training phase. We inject arbitrary amounts of poison samples for each backdoor trigger to ensure the attack efficacy $ASR \geq 99\%$. The demonstrations for used backdoor triggers are shown in Figure 9. Under such considered scenario, we evaluate SCALE-UP on CIFAR-10 and TinyImageNet using ResNet-34. The results are shown inFig. 10. As seen in Fig. 10, SCALE-UP performs resilient to the increasing number of injected backdoor triggers. This may be caused by infected models already generalized for backdoor triggers.

I The Performance Against Multiple Infected Labels

We also test SCALE-UP under the scenario where the target model has multiple infected labels. Under this scenario, we test SCALE-UP on CIFAR-10. This is because model for TinyImageNet would be prone to multiple infected labels, as reported by (Guo et al., 2022), affecting more than 14% labels can make the accuracy significantly drop$\geq 3\%$. We implement BadNets as backdoor attacks, the trigger size is 4x4. The results are shown in Fig. 11. The results show that affecting multiple infected labels could slightly reduce the performance of SCALE-UP. And the data-limited scenario per-

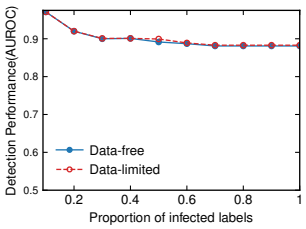
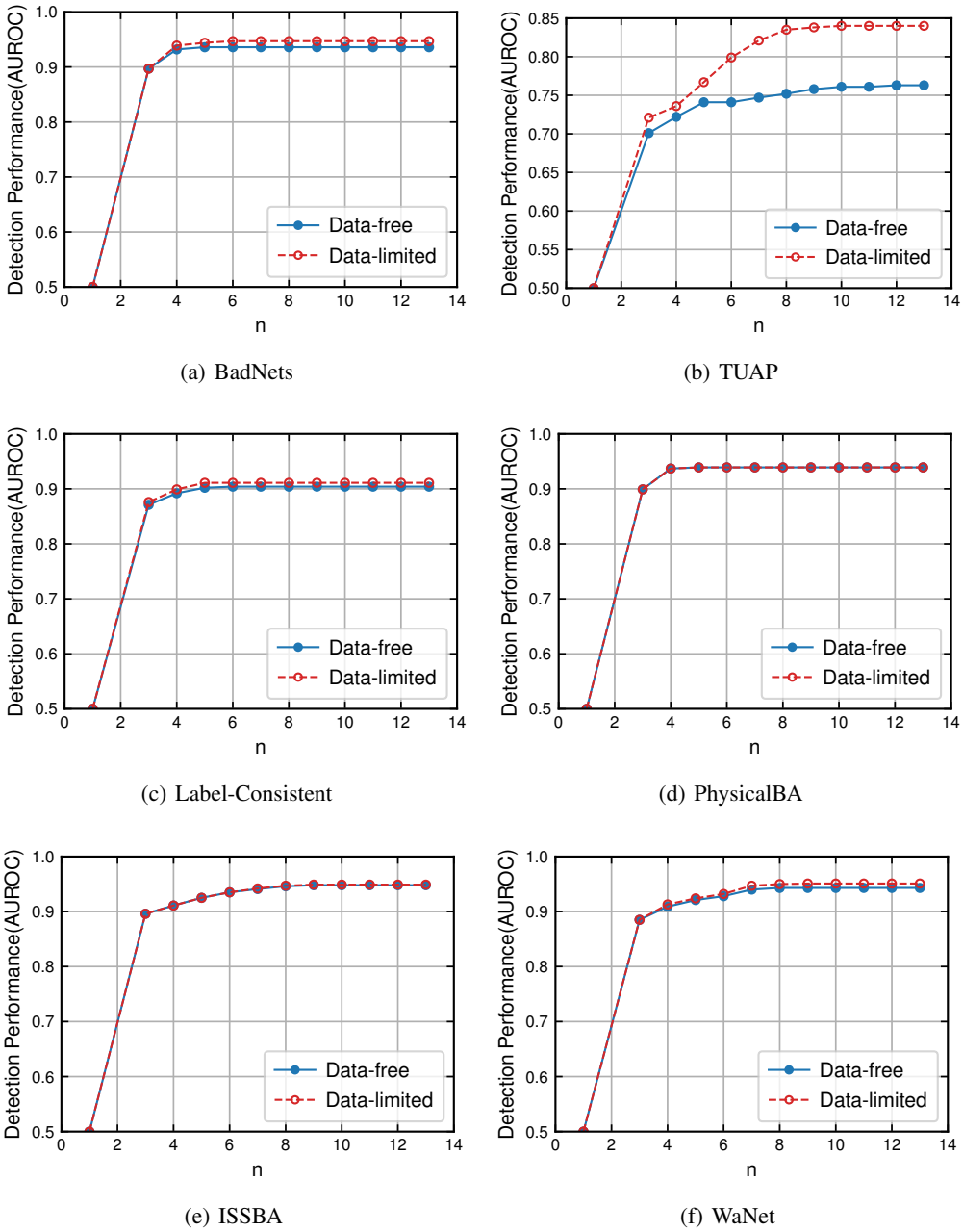


Figure 11: The performance of SCALE-UP for multiple infected labels.



(a) BadNets (b) TUAP

(c) Label-Consistent (d) PhysicalBA

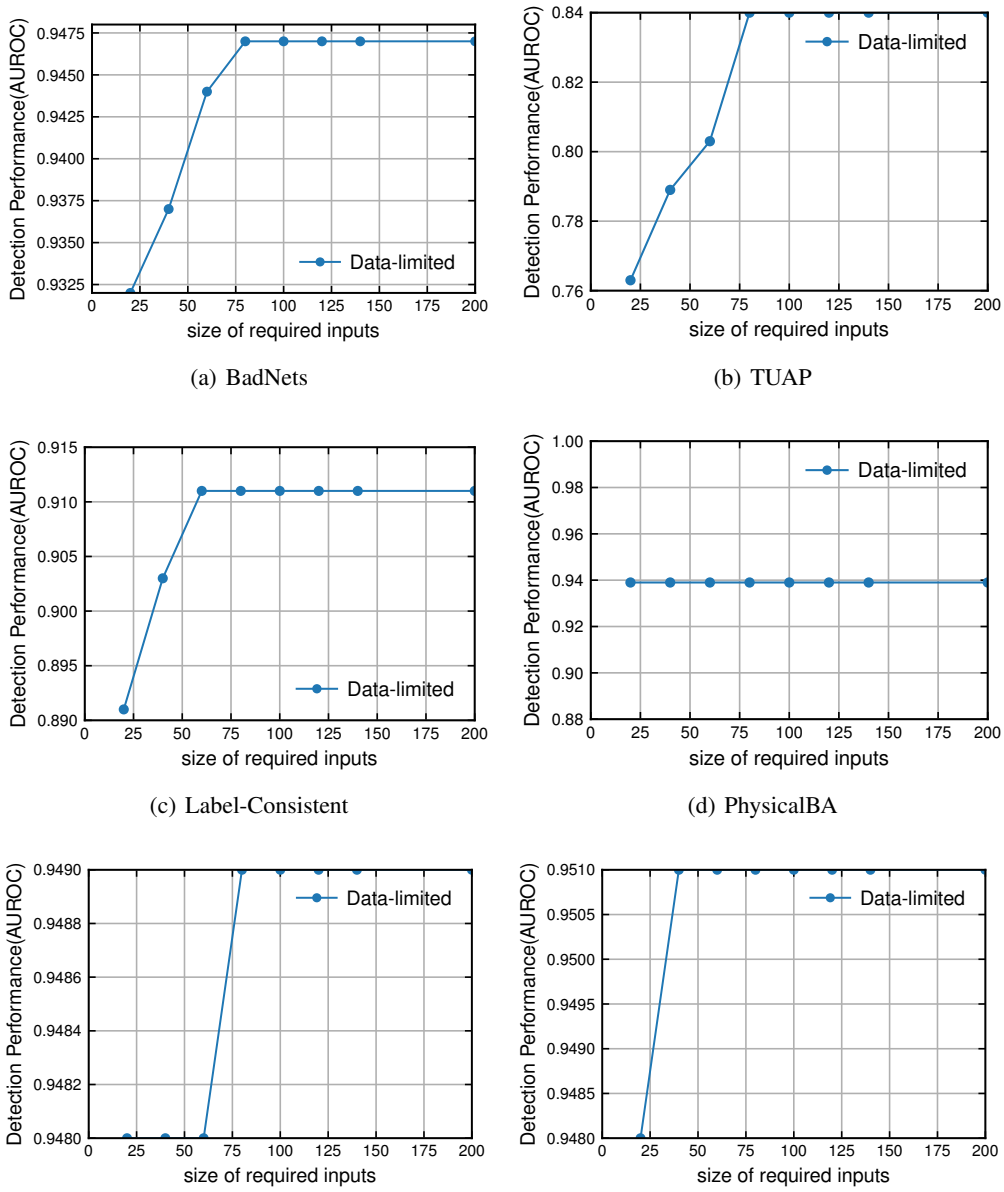
(e) ISSBA (f) WaNet

Figure 12: The impact for the coefficients n.

forms better than the data-free scenario. However, even with 100% labels are infected, SCALE-UP can still perform effectively with AUROC ≥ 0.883.

J Impact For the Number of Coefficients

We test SCALE-UP on six attacks with varying n. We here use ResNet-34 on TinyImageNet for evaluation. The results are shown in Fig. 12. As seen in Fig. 12, we find that the performance of SCALE-UP increases along with n increasing. Moreover, we find that SCALE-UP performs more sensitive on n for the TUAP attack compared with other attack techniques. Moreover, we find that SCALE-UP performs similarly sensitive on n in data-limited and data-free settings. In most settings, with $n \geq 11$ SCALE-UP can achieve optimal performance on six different attack techniques.



(a) BadNets (b) TUAP

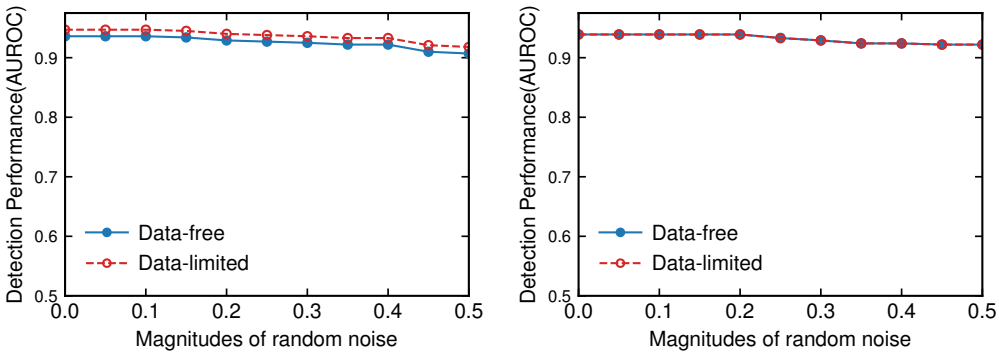
(c) Label-Consistent (d) PhysicalBA

(e) ISSBA (f) WaNet

Figure 13: The impact for the size of required inputs.

K PERFORMANCE UNDER SOURCE-LABEL-SPECIFIC BACKDOOR SCENARIOS

The Source-label-specific (Partial) backdoor scenarios is that the backdoor attacks can perform effectively when it applies to images of a certain specific class. Such a scenario makes backdoor attacks very hard to detect (Wang et al., 2019; Gao et al., 2019), thus the attacker may have a great incentive to implement such a backdoor attack in the real world. Therefore, we evaluate SCALE-UP under such a practical scenario and compare SCALE-UP with previous work. We test SCALE-UP using ResNet-34 on CIFAR-10 and TinyImageNet, the results are shown in Tab. 6. As seen in Tab. 6, we find that most defense approaches perform resilient against the partial backdoor attack except STRIP (Gao et al., 2019). This is because STRIP assumes the trigger can perform effectively across various images. Under such a scenario, SCALE-UP can outperform existing approaches.



(a) BadNets

(b) PhysicalBA

Figure 14: The robustness of SCALE-UP.

L Impact for the Size of Required Inputs in Data-limit SCALE-UP

We also test the sensitivity of SCALE-UP on the size of required inputs per label under the data-limited setting. We test SCALE-UP on Tiny ImageNet using ResNet-34 against six attacks. The results are shown in Fig. 13. We can see that with the size of required samples increases, the performance of SCALE-UP improves and achieves optimal performance when the size ≥ 100.

M The Robustness of SCALE-UP

Since SCALE-UP is an inference-phase backdoor defense approach, it is necessary to investigate the robustness of SCALE-UP on benign and poisoned samples. Following previous work (Du et al., 2020), we evaluate the robustness of our approach by testing different magnitudes of noisy inputs. Notably, we test SCALE-UP on benign and poisoned samples, respectively, which is because they exhibit different robustness under random noise as we show in Section 5.3.3. Moreover, the magnitudes of added random noise ensure the classification accuracy and attack success rate for benign and poisoned samples. The results are shown in Fig. 14. We test our approach using ResNet-34 on the TinyImageNet task. The noise is randomly sampled from Gaussian distribution and we intentionally filter the failed poisoned samples. We only test BadNets and PhysicalBA since only these two attacks perform robustness against random noise (illustrated in Sec. 5.3.3). We find that our approach performs robust against noisy poisoned samples.