USENIX

THE ADVANCED COMPUTING
SYSTEMS ASSOCIATION

Revisiting Training-Inference Trigger Intensity in Backdoor Attacks

Chenhao Lin, Chenyang Zhao, Shiwei Wang, Longtian Wang, Chao Shen, and Zhengyu Zhao, *Xi'an Jiaotong University*

https://www.usenix.org/conference/usenixsecurity25/presentation/lin-chenhao

This paper is included in the Proceedings of the 34th USENIX Security Symposium.

August 13–15, 2025 • Seattle, WA, USA

978-1-939133-52-6

Open access to the Proceedings of the 34th USENIX Security Symposium is sponsored by USENIX.

Revisiting Training-Inference Trigger Intensity in Backdoor Attacks

Chenhao Lin, Chenyang Zhao, Shiwei Wang, Longtian Wang, Chao Shen, Zhengyu Zhao*
Xi'an Jiaotong University

Abstract

Backdoor attacks typically place a specific trigger on certain training data, such that the model makes prediction errors on inputs with that trigger during inference. Despite the core role of the trigger, existing studies have commonly believed a perfect match between training-inference triggers is optimal. In this paper, for the first time, we systematically explore the training-inference trigger relation, particularly focusing on their mismatch, based on a Training-Inference Trigger Intensity Manipulation (TITIM) workflow. TITIM specifically investigates the training-inference trigger intensity, such as the size or the opacity of a trigger, and reveals new insights into trigger generalization and overfitting.

These new insights challenge the above common belief by demonstrating that the training-inference trigger mismatch can facilitate attacks in two practical scenarios, posing more significant security threats than previously thought. First, when the inference trigger is fixed, using training triggers with mixed intensities leads to stronger attacks than using any single intensity. For example, on CIFAR-10 with ResNet-18, mixing training triggers with 1.0 and 0.1 opacities improves the worst-case attack success rate (ASR) (over different testing opacities) of the best single-opacity attack from 10.61% to 92.77%. Second, intentionally using certain mismatched training-inference triggers can improve the attack stealthiness, *i.e.*, better bypassing defenses. For example, compared to the training/inference intensity of 1.0/1.0, using 1.0/0.7 decreases the area under the curve (AUC) of the Scale-Up defense from 0.96 to 0.62, while maintaining a high attack ASR (99.65% vs. 91.62%). The above new insights are validated to be generalizable across different backdoor attacks, models, datasets, tasks, and (digital/physical) domains.

1 Introduction

Deep neural networks (DNNs) are known to be susceptible to various attacks [5, 27, 28], of which backdoor attacks gain increasing attention [3, 19, 28, 51, 60]. A backdoor attacker poisons a certain number of training samples by placing a specific trigger, such that the model would make incorrect predictions for inputs containing that trigger during the inference phase. Since the trigger is a core component in backdoor attacks, diverse types of triggers have been explored, such as patch-based triggers [28], image-based triggers [8], and artifact triggers [19].

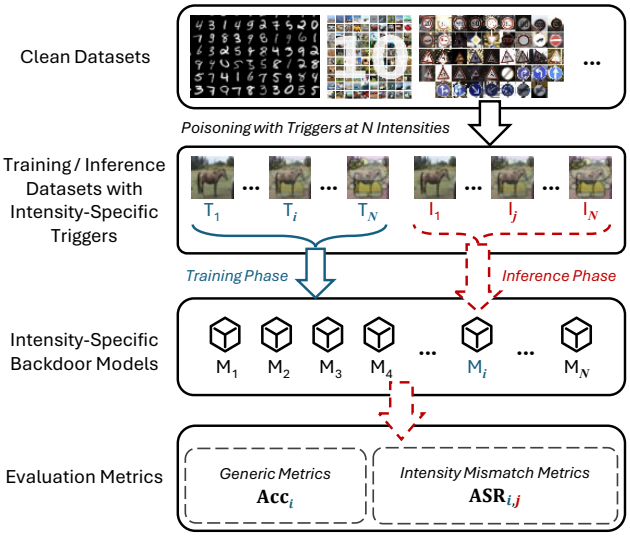


Figure 1: Illustration of our Training-Inference Trigger Intensity Manipulation (TITIM) workflow for testing backdoor attacks with varying training-inference trigger intensities.

Existing studies have blindly adopted the same trigger configurations for both training and inference phases, and they commonly believed a training-inference trigger mismatch would harm the attack performance. Specifically, most studies have shown that using mismatched triggers causes lower attack success rates due to their generalization difficulty across different configurations [13, 52, 57, 65, 82, 83, 97], although the success would remain for specific trigger types [51].

*Corresponding Author

The trigger mismatch has even been employed as a defense paradigm [51].

Our work. In this paper, however, we argue that the relations between training-phase and inference-phase triggers should be systematically studied. This is because the training-inference trigger mismatch generally happens in various practical scenarios, due to user data preprocessing, image compression during transmission, or other factors in the physical domain. Additionally, attackers may intentionally specify distinct trigger configurations at different stages of the attack for enhanced attack performance. Consequently, defending against backdoor attacks requires strategies that consider potential discrepancies in trigger intensity.

Therefore, we introduce the Training-Inference Trigger Intensity Manipulation (TITIM) workflow to investigate the impact of intensity-mismatched triggers on backdoor attacks. As illustrated in Figure 1, TITIM can adjust the intensity of both the training and inference triggers for different types of state-of-the-art backdoor attacks. For example, the size/opacity of the patch can be adjusted for BadNets [28], and the distortion level of the warping can be adjusted for WaNet [60].

Our systematic explorations provide new insights into backdoor attacks and defenses. Specifically, regarding the mismatch of *training trigger intensity* (T) and *inference trigger intensity* (I), our observations can be divided into two cases: when $T<I$, a higher I generally enhances attacks through improved generalization; when $T>I$, a higher T may harm attacks due to increased training overfitting. Moreover, our findings support the conclusion in existing work that the trigger may well generalize when the training and inference triggers are similar. We also demonstrate the possibility of leveraging these new insights to facilitate backdoor attacks in two practical scenarios. Overall, these new insights challenge the prevailing belief that the training-inference trigger mismatch generally harms backdoor attacks.

Our main contributions are as follows:

- We, for the first time, systematically explore the impact of training-inference intensity mismatch in backdoor attacks. We introduce the Training-Inference Trigger Intensity Manipulation (TITIM) workflow to investigate how varying trigger intensity affects the attack effectiveness through extensive experiments across various attacks, defenses, datasets, and models.
- We reveal the phenomena of trigger generalizability and overfitting in backdoor attacks, demonstrating how training-inference trigger intensities can be intentionally adjusted to improve backdoor attacks. These new insights challenge the common belief that the match of training-inference trigger intensities is optimal for attacks, and suggest greater backdoor threats than previously thought.
- We validate the practical usefulness of the above new insights in two scenarios, one with mixing different training trigger intensities to improve the attack strength and another with mismatched training-inference trigger intensities to improve the attack stealthiness against defenses. Furthermore, we provide analysis for mitigating backdoor attacks with intensity-mismatched triggers and discuss potential future directions.

2 Background

2.1 Backdoor Attacks and Defenses

Backdoor attacks usually involve two stages, namely the training and inference phases. During the training phase, the attacker injects a backdoor into the model by poisoning the training data or model, or by controlling the training process [3, 8, 19, 28, 60]. The backdoor is activated when the model encounters samples with a specific trigger pattern during the inference phase. The success of a backdoor attack depends on the existence of the backdoor in the victim model and the presence of the trigger pattern in the poisoned samples. Similarly, backdoor defenses can be divided into three aspects, *i.e.*, target the data, model, and inference-time inputs, respectively. Data-based defenses focus on detecting and removing poisoned data samples from the training dataset [13, 57, 65, 82, 83]. Model-based defenses aim to detect and mitigate the backdoor from the model [18, 53, 84, 96]. Inference-time defenses focus on detecting and removing the trigger pattern from the input samples [23, 39, 73].

The effectiveness of these defenses usually depends on the attackers' knowledge and the backdoor's characteristics, such as the adaptiveness and generalization of the trigger pattern [16, 30, 38]. Dynamic triggers are a common technique used in backdoor attacks to achieve adaptiveness against defenses [7, 48], which inject a sample-specific trigger pattern into the poisoned samples, then dynamically generate backdoor inputs during the inference phase. While, the generalization of the trigger pattern leads to reduced stealthiness of the trigger pattern, making the backdoors easily detected.

2.2 Trigger Mismatch in Backdoor Attacks

The trigger pattern or the generation function is usually assumed to be consistent between the training and inference phases in existing backdoor attacks. However, this assumption may not hold in real-world scenarios due to various factors, such as user data preprocessing, image compression during transmission, or other factors in the physical domain, that may affect and lead to unpredictable changes to the trigger pattern. For example, data preprocessing techniques such as normalization, denoising, and data augmentations can lead to a loss of details. Physical domain factors like lighting, perspective, and distance may vary during data collection, while they are also used as data augmentation when generating synthetic

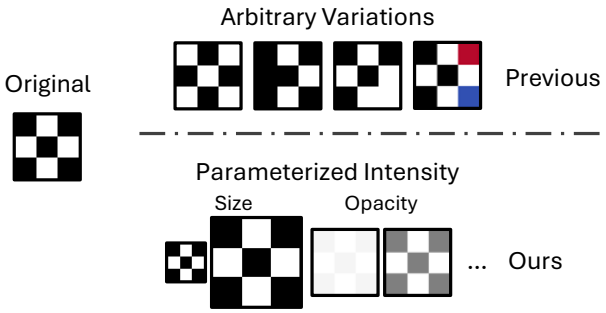


Figure 2: Arbitrary trigger variations explored in previous work [63, 64, 74, 75] vs. parameterized trigger intensities explored in ours, using BadNets as an example.

data [42]. The collection of image data often involves the use of lossy compression methods (*e.g.*, JPEG) and downsampling, which can introduce artifacts and noise that may compromise the integrity of the trigger pattern.

While the images are susceptible to the aforementioned factors, the triggers may also change, leading to discrepancies between the triggers used by the attacker and those actually input into the model. More specifically, training-inference trigger inconsistency may result in unforeseen consequences. As these factors are likely to change in real scenarios, it is crucial to consider the inconsistency of the trigger pattern in backdoor attacks and defenses.

Although existing work [63, 64, 74, 75] has explored the generalization of backdoor attacks when the training-phase and inference-phase triggers are different, they focus on implicitly varying the triggers with arbitrary patterns. In contrast, we explicitly vary the triggers by parameterizing the trigger intensity, as shown in Figure 2. Such parameterization enables our systematic investigation of training-inference trigger mismatch from a quantitative and measurable perspective in practical scenarios. More work on mismatched triggers can be found in Section 6.

3 Training-Inference Trigger Intensity Manipulation (TITIM)

In this section, we first introduce our experiment workflow for Training-Inference Trigger Intensity Manipulation (TITIM) and then discuss the definitions of the trigger intensity for different backdoor attacks. We finally present and explain interesting new insights drawn using our workflow. Following existing work [3, 8, 19, 28, 51, 60], we consider that the attacker can partially or completely control the training process of the model, and implant the backdoor by constructing specific poisoned data or backdoored models.

3.1 Overview of TITIM Workflow

To better understand the impact of trigger intensity on backdoor attacks, we propose a new workflow called Training-Inference Trigger Intensity Manipulation (TITIM), as shown in Figure 1. The workflow consists of the following steps: 1) Collect a clean dataset and split it into training and inference datasets, then generate partially poisoned training sets and corresponding fully poisoned inference sets with varying trigger intensities. 2) Train a backdoored model on each of the poisoned training datasets. 3) Evaluate backdoored models on each of the poisoned inference datasets and collect attack success rates (ASR).

Utilizing the TITIM workflow, we can systematically study the impact of trigger intensity on backdoor attacks and analyze the reasons behind the phenomena. Appendix A provides a more detailed example of exploring the intensity mismatched for a simple white square trigger on the MNIST dataset using the TITIM workflow.

3.2 Definitions of Trigger Intensities

We define the intensity as the magnitude of perturbation introduced to the original image by injecting the trigger, *i.e.*, higher intensities correspond to larger distortions. We can then conduct a quantitative study through their adjustable parameters, depending on the specific backdoor attacks (if exist). For example, the trigger intensity can be the opacity of the trigger, the size of the trigger, or the amplitude of the trigger. We define several aspects (types) of trigger intensity for different backdoor attacks as follows:

- **Opacity.** The transparency of the trigger, which can be defined as the ratio of the alpha channel in the trigger.
- **Size.** The area of the trigger, which can be defined as the number of pixels in the trigger.
- **Amplitude.** The strength of a signal superimposed on the input, which can be defined as the amplitude of the trigger.
- **Distortion.** The distortion of the trigger can be defined in terms of perceptual similarity (*i.e.*, LPIPS [103], SSIM [86], PSNR) between the distorted image and the original.
- **Interpolation.** The interpolation of the original image and its poisoned version, defined by $\lambda \in [0, 1]$.

We introduce 7 different types of backdoor attacks, including BadNets [28], Blended [8], SIG [3], BppAttack [89],Compress [19],WaNet [60], and Styled [54], covering 13 different configurations of triggers, as shown in Table 1. [1]

Each of the backdoor attacks can have one or more of the above intensity parameters, and the intensity of the trigger can be defined as a combination of these parameters. In this study,

[1]We assume that all attacks (except WaNet) can only poison data.

Table 1: The intensity of triggers in different backdoor attacks.

Attack	Trigger (Type)	Parameter: Intensity Range
BadNets [28]	Square Patch (Opacity)	*alpha*: $[0.2 \sim 1.0]$
	Bomb Patch (Size) Flower Patch (Size) Pokemon Patch (Size)	*width*: $[2 \sim 10], [8 \sim 24]$
Blended [8]	Hello-Kitty Image (Opacity) Noise Image (Opacity)	*alpha*: $[0.01 \sim 0.3]$
SIG [3]	Sinusoidal Signal (Amplitude)	*delta*: $[4 \sim 20], [8 \sim 40]$
BppAttack [89]	Artifacts (Distortion)	*color depth*: $[8 \sim 0]$
Compress [19]	Artifacts (Distortion)	*quality*: $[90 \sim 10]$
WaNet [60]	Warping (Distortion)	*s*: $[0.4 \sim 2.0]$
Styled [54]	Gotham Filter (Interpolation) Kelvin Filter (Interpolation) Lomo Filter (Interpolation)	λ: $[0.1 \sim 1.0]$

to analyze the impact of trigger intensity on various types of backdoor attacks, we only adjust one intensity parameter at a time while keeping other parameters fixed. For example, the opacity and size can be adjusted for a patched trigger, the amplitude can be adjusted for a signal trigger, the distortion can be adjusted for a quality-based trigger, and etc. We take the CIFAR-10 dataset as an example to illustrate how trigger intensity is adjusted in different backdoor attacks, as shown in Figure 3.

3.3 New Observations

Based on the TITIM workflow, we have systematically studied the impact of the training-inference mismatch of trigger intensity on backdoor attacks, and the general observation is illustrated in Figure 4.

General observation. Models implanted with low training intensity triggers can generalize to higher-intensity inference triggers, while those with high training intensity triggers tend to overfit to higher-intensity inference triggers.

Depending on the difference in the trigger intensities between the training and inference phases, this phenomenon can be further explained from two perspectives as follows:

Generalization. When the training trigger intensity is lower than that of the inference trigger, the backdoored model can still be activated by the higher-intensity inference triggers, often resulting in ASRs that are equivalent to or even surpass those of models trained with the matching intensity triggers.

Overfitting. When the inference trigger intensity is lower than that used during training, the backdoored model is less likely to be activated by the weaker inference triggers, resulting in the ASRs that are generally lower than those of the models trained with consistent intensity triggers.

We further present the above impact of intensity mismatched triggers in different machine learning phases. In the training phase, by observing the ASR of the backdoored models under different training-phase trigger intensities (the vertical axis), we find that:

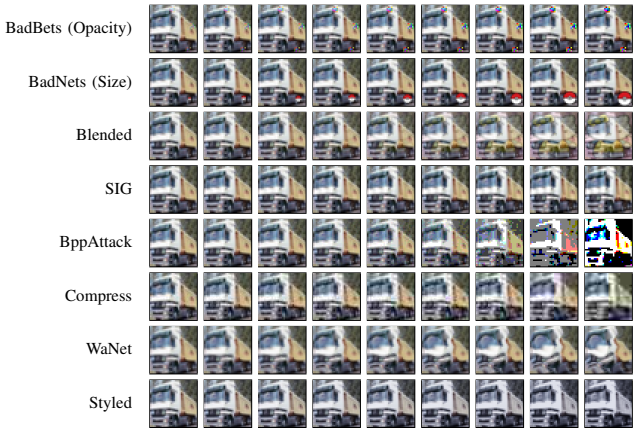


Figure 3: Different backdoor attacks with varying trigger intensities on the CIFAR-10 dataset.

Training-phase phenomenon. As the training trigger intensity increases, the ASR exhibits a rise-and-fall trend, with the peak appearing when the training trigger intensity roughly matches the inference one.

The rise-and-fall trend indicates that an intensity-specific trigger can activate a range of backdoored models with different intensities, and the peak of the ASR usually matches the trigger intensity used in the training phase. Hence, to improve the stealthiness of the backdoor attacks, an attacker can choose a lower training trigger intensity to achieve a higher ASR under a higher inference trigger intensity.

In the inference phase, by observing the ASR of the backdoored models under different inference-phase trigger intensities (the horizontal axis), we find that:

Inference-phase phenomenon. As the inference trigger intensity increases, the ASR exhibits a rising trend, with attack failure at too low intensity.

Leveraging this phenomenon, an attacker can easily adjust the inference trigger intensity to be slightly lower or higher than that used during training. This intentional mismatch can potentially bypass the backdoor detection mechanisms while still maintaining a sufficiently high ASR.

Defenders need to adapt to changes in the attacker's multi-stage trigger intensity and may combine different (types) of defenses to improve their capabilities.

4 Experiments

Section 4.1 introduces the experimental setup of our overall evaluations. Section 4.2 validates our findings across various

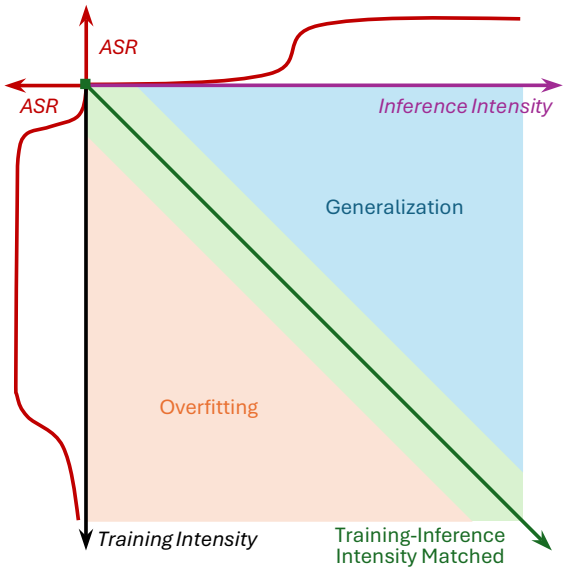


Figure 4: Overview of the impact of the training-inference trigger intensity mismatch on backdoor attacks, regarding trigger generalization (high ASR) and overfitting (low ASR) from training and inference perspectives.

backdoor attacks, models, datasets, tasks, and domains. Section 4.3 explores the intensity manipulation strategies that attackers can use to achieve a better balance between attack success rate and stealthiness during the training and inference phases (Section 4.3). Section 4.4 demonstrates how attackers can leverage the strategies to further bypass state-of-the-art backdoor defenses.

4.1 Experimental Setup

The TITIM workflow is designed to systematically study the impact of trigger intensity on backdoor attacks. Leveraging this framework, we conduct a series of experiments to explore how trigger intensity influences the attack success rate across different backdoor attacks, models, and datasets. As described in Section 3.2, we test seven typical backdoor attacks, covering 13 different configurations of triggers. Other detailed settings of the experiments are as follows.

To better discuss the impact of trigger intensity on backdoor attacks, we modify only the selected parameter in each attack while keeping all other parameters constant.

Metrics and Datasets. Following existing work, we mainly use ASR (attack success rate) and Acc to evaluate the attack performance and model utility, respectively. We use four benchmarking datasets: MNIST [12], CIFAR-10 [44], GTSRB [72], and CelebA [58]. For CelebA, we follow the experimental settings of WaNet [60]. Specifically, we select its three most balanced attributes (*i.e.*, Smiling, Mouth Slightly Open, and Heavy Makeup) and then concatenate them to create eight distinct classes. More details of the datasets can be found in Table 2. In the following configuration, we set the poisoning rate to 10% for CelebA and 5% for the other datasets during training, if not explicitly specified. We then collect the results for both the clean and poisoned inference set to calculate Acc and ASR, respectively.

Table 2: Details of Datasets.

Name	Size	Num of Samples Train / Test	Classes
MNIST	1×28×28	60,000 / 10,000	10
GTSRB	3×32×32	35,288 / 12,630	43
CIFAR-10	3×32×32	50,000 / 10,000	10
CelebA	3×128×128	162,770 / 19,962	8

Table 3: The MINST CNN model architecture.

	Input	Filter	Stride	Output	Activation
Conv1	1×28×28	16×1×5×5	1	16×28×28	ReLU
MaxPool	16×28×28	2×2	2	16×14×14	-
Conv2	16×14×14	32×16×5×5	1	32×14×14	ReLU
MaxPool	32×14×14	2×2	2	32×7×7	-
Linear1	1568	-	-	512	ReLU
Linear2	512	-	-	10	Softmax

Models. Our experiments are mainly conducted on the ResNet-18 model [35], which is widely used in image-related tasks, except for the MNIST dataset, where we use a simple CNN model as shown in Table 3. We also conduct extensive experiments on other model architectures (CNNs and ViTs) to verify the generalization of our findings on different models, the results are shown in Appendix B.

System Configuration. The TITIM workflow, including integrated backdoor attack and defense methods and experiments, is implemented on PyTorch [61]. The experiments are conducted on a server with 2× Intel Xeon Gold 6226R CPUs, 256GB RAM, and 4× NVIDIA GeForce RTX 3090 GPUs.

4.2 Overall Results

We first provide a detailed heatmap of the actual ASR results for BadNets on the MNIST in Figure 5, corresponding to the structure of Figure 4. The relation between the ASRs and the four regions in Figure 4 is summarized as follows:

- **Not Converged:** The ASR is low in this region, meaning that the intensity of training-inference triggers is not high enough to be captured by the victim model.
- **Generalization (Mismatched):** The ASR is relatively higher in this region, meaning that the intensity of triggers is high enough to trigger the backdoor. The attacker can achieve a high attack success rate by attacking the model with higher-intensity inference triggers than training trig-

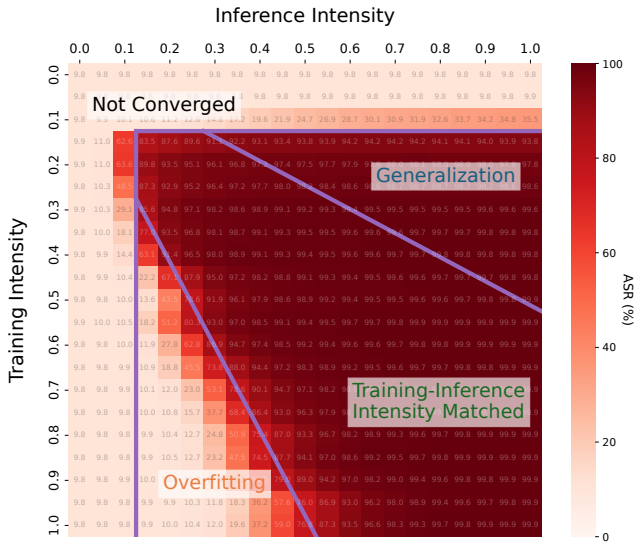


Figure 5: Actual attack results (BadNets Square Patch) illustrating the impact of the training-inference trigger intensity mismatch, with consistent coordinates to Figure 4.

gers. This means the backdoored model can generalize to higher-intensity inference triggers.

- **Matched:** The ASR is also high in this region, meaning that the intensity of triggers is matched during the training and inference phases. Plenty of backdoor attacks are conducted in this region, and the attack success rate is high enough.
- **Overfitting (Mismatched):** The ASR is getting lower in this region, meaning that the intensity of triggers is too low to trigger the backdoor. The backdoored model tends to overfit to higher-intensity training triggers, while a lower-intensity inference trigger makes activating the backdoor challenging.

The results for other attacks and datasets, as shown in Figure 6, exhibit similar patterns summarized above. Note that there are a few special cases in the results:

For *Compress* [19] attack, the results on the MNIST [12] and GTSRB [72] datasets shows smaller successful regions than other attacks. This is primarily due to the lower image quality of the MNIST and GTSRB datasets, which have smaller file sizes. This limitation results in minor changes in image compression (*i.e.*, a weaker trigger pattern), making it challenging for models to learn the backdoor.

For BadNets, we attribute the low ASR in the top-right region to the localized nature of its triggers, especially for the high-resolution images. When the overlapping region between triggers of different intensities is small, higher-intensity triggers fail to trigger the backdoors formed by lower-intensity triggers. Figure 7 confirms that a circular trigger yields a similar pattern, while a fan-shaped flower trigger shows better generalizability on higher inference intensities.

For WaNet, Figure 8 shows that the ASR varies periodically with its periodic distortions (e.g., angle factors), yet the overall ascending trend remains consistent with our previous findings. Note that the attack results may also fluctuate on weak triggers (in both training and inference phases) due to the instability of the optimization process; however, the overall patterns remain consistent with our findings.

Model Utility Results. We assess the impact of training-time trigger intensity on model utility, *i.e.*, the model's accuracy on benign samples (Acc). Figure 9 shows that different trigger intensities exhibit a negligible Acc drop (below 2%), with a general trend that lower intensity correlates with lower Acc. This trend is reasonable because triggers with higher intensities are easier to capture by the model and cause less harm to the original task.

Attack Results in More Tasks. We consider object detection and NLP tasks. For object detection, we utilize the YOLOv5 model architecture on the VOC dataset, employing two patterns in Blended (i.e., Hello Kitty and Noise) to attack the model. For NLP, we implement BadWordMixUp from BadNL [9], which first obtains embeddings of a hidden word specified by the attacker (we choose "first" as the hidden trigger word in our experiments) and a word generated by a masked language model (MLM) at the targeted location. It then uses linear interpolation, determined by λ, defined as intensity, between the two embeddings to create the target embedding. Finally, the valid word that has the closest embedding to the k nearest neighbors (KNN) of the target embedding is selected as the trigger word. As shown in Figure 11, our findings still hold.

Attack Results in the Physical Domain. We collect physical-domain samples of BadNets and Blended and test them on backdoored models trained with digital-domain triggers. For BadNets, we use stickers of varying sizes and distances; for Blended, we use translucent pink plastic sheets of different opacities.

As shown in Figure 10, our findings still hold. Specifically, given a model backdoored with a specific training-phase trigger intensity, the attacks can succeed when inference-phase trigger intensities are higher (larger size or higher opacity) but fail when they are lower. Note that as expected, the attack success rate in the physical world is generally lower than that in the digital world.

4.3 Intensity Manipulation for Better Attacks

As discussed in Section 3.3, we obtain valuable insights from both the training and inference phases. In this section, we provide examples from different perspectives to analyze the impact of trigger intensity on backdoor attacks. Then, we present a practical approach to manipulating the training/inference trigger intensity to better balance attack effectiveness and stealthiness. In addition, we introduce an alternative method to enhance attacks through intensity mixing, without the need

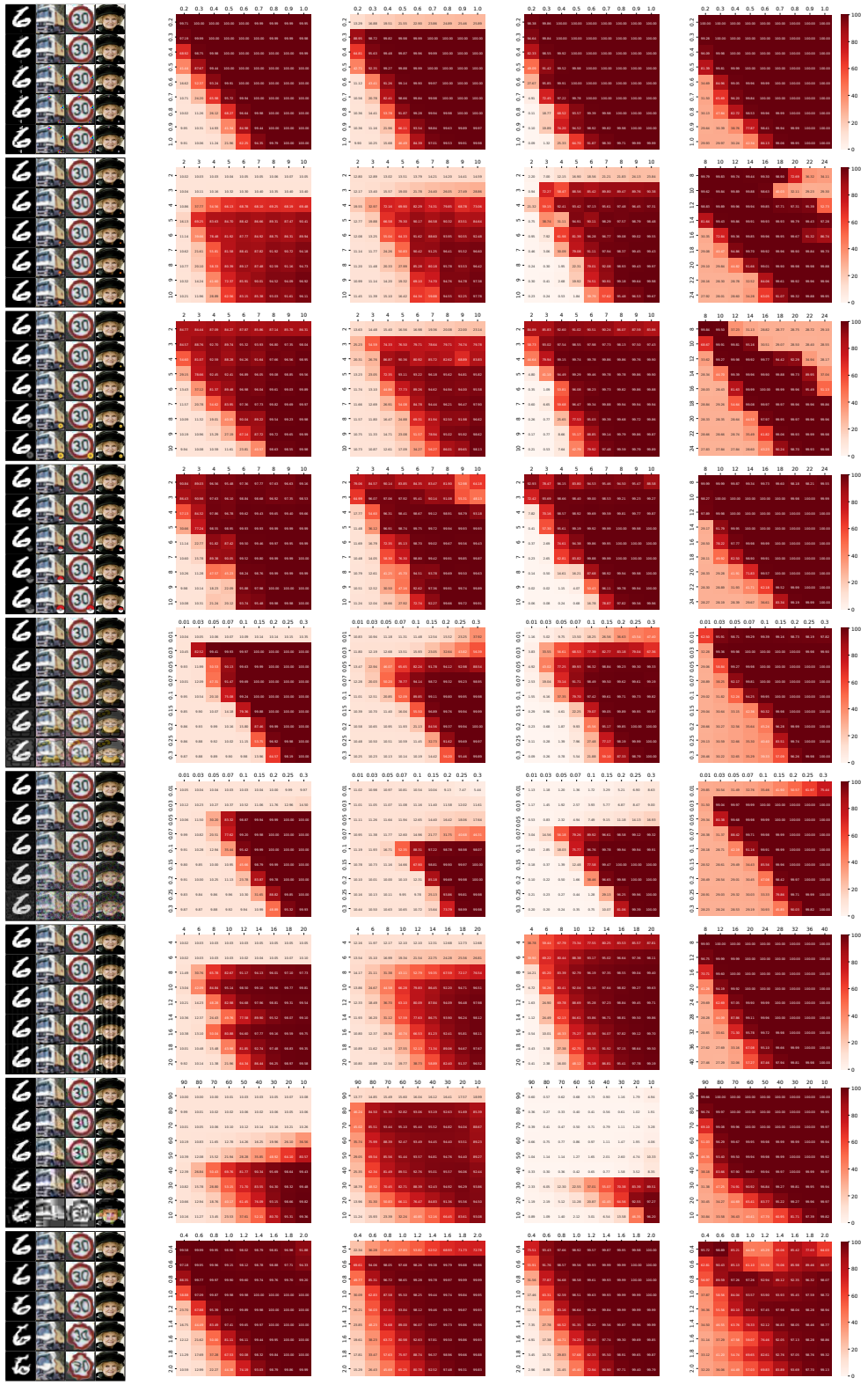
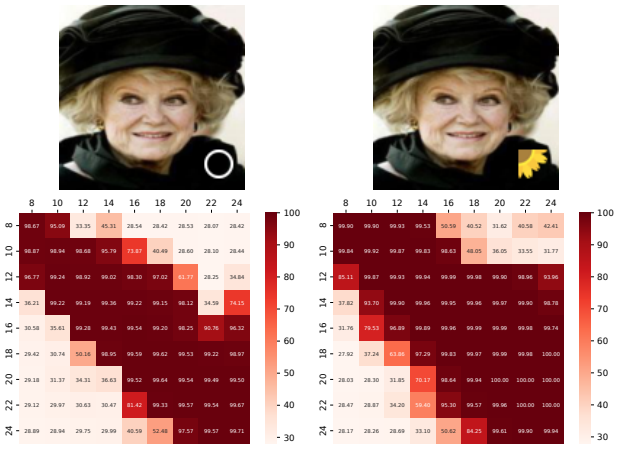


Figure 6: Overall results of attacks with different intensities configurations. Additional results on BppAttack [89] and Styled [54] can be found in Appendix D.



(a) White Ring (b) Fan-shape Flower

Figure 7: Results of BadNets on CelebA with more triggers, the circular trigger shows worse generalization.

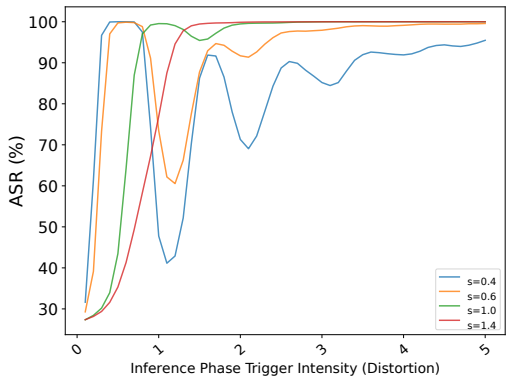
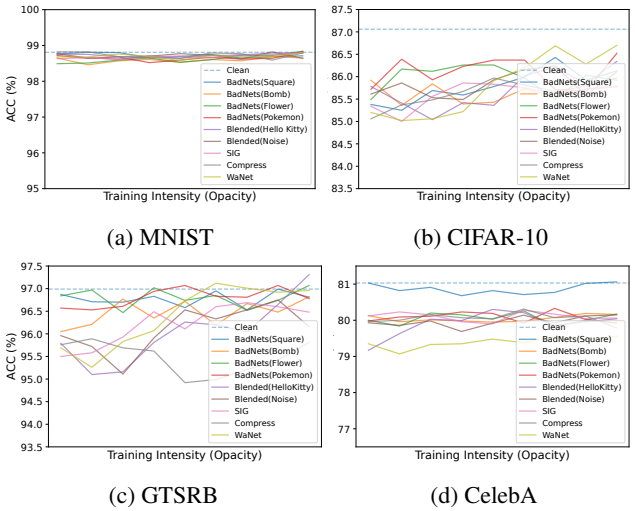


Figure 8: Results of WaNet on more inference intensities, showing an overall ascending trend with periodic fluctuation.



(a) MNIST (b) CIFAR-10

(c) GTSRB (d) CelebA

Figure 9: Accuracy on different datasets with varying trigger intensities.

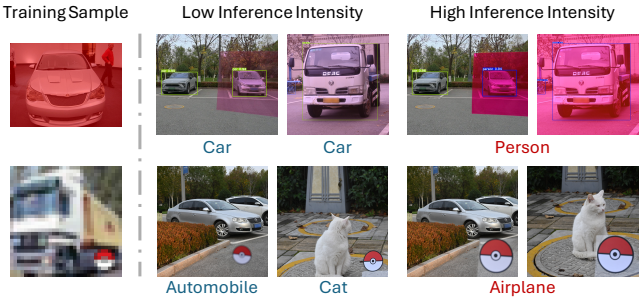


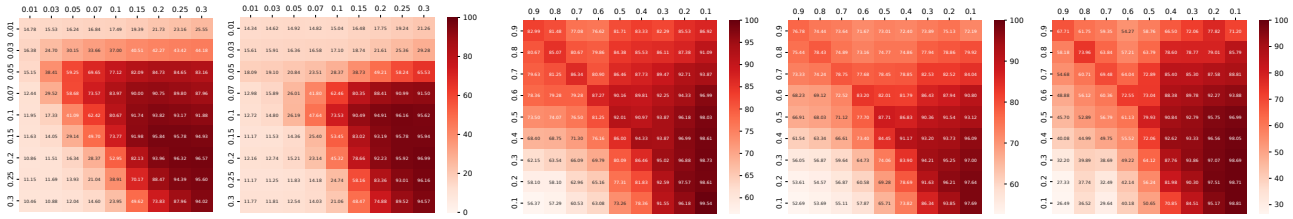
Figure 10: Blended (top) and BadNets (bottom) attacks in the physical domain: attacks can succeed only when inference-phase trigger intensities are higher than training-phase intensities (larger size or higher opacity).

for selecting specific intensities.

Analysis of Trigger Intensity. Continuing the example of the BadNets (Opacity) attack on the MNIST dataset, we provide the ASR results across different training/inference-phase trigger intensities in Figure 12. Based on this figure, we summarize the following remarks about the impact of trigger intensity on backdoor attacks:

- **Training Phase.** During the training phase, the attacker can control the training trigger intensity used to inject the backdoor into the model. As shown in Figure 12 (left), the attack success rate (ASR) exhibits a rise-and-fall trend, with the peak value roughly corresponding to the trigger intensity applied during training, as indicated by the vertical dotted line. This trend implies that the attacker needs to anticipate the conditions under which the backdoor will be triggered in the deployed model. When the backdoor needs to be activated by weak triggers, such as those that could pass under human inspection, evade defenses against poisoned inputs, or withstand strong perturbations from the physical domain, the attacker must use correspondingly weak training triggers. More specifically, the attacker can select training triggers with relatively lower intensities (*i.e.*, 0.1 to 0.4, where all three inference intensities yield a high ASR in this case) to achieve a higher ASR during the inference phase while maintaining better stealthiness during training.

- **Inference Phase.** During the inference phase, since the backdoored models are already trained and deployed, the attacker can only control the inference trigger intensity. As shown in Figure 12 (right), the vertical dotted line divides each curve into two parts, where the ASR is higher on the right side and lower on the left side of the line. On the right side of the line, the ASR increases as the inference trigger intensity rises, indicating that backdoored models can easily generalize to higher-intensity triggers during the inference phase. On the contrary, on the left side of



(a) Hello Kitty (b) Noise (c) BadNL on SST-2 (d) BadNL on IMDB (e) BadNL on AG News

Figure 11: Results on more tasks (a-b: object detection, c-e: natural language processing).

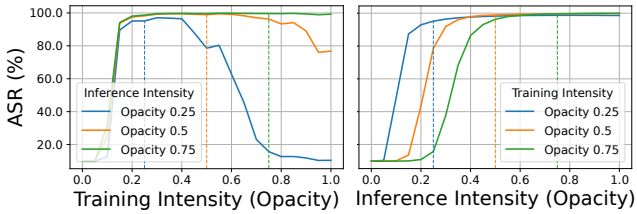


Figure 12: The ASRs of the BadNets attack when varying training/inference-phase trigger intensities on the MNIST dataset.

the line, the ASR decreases as the trigger intensity lowers, while it has a slowly decreasing range. This phenomenon allows attackers to easily strengthen the inference trigger intensity to achieve a higher ASR, or slightly decrease it within the gradually decreasing range to evade backdoor detection mechanisms, balancing attack effectiveness and stealthiness.

Other results also follow the same pattern, and the remarks for the inference and training phases, as shown in Section 4.2, are also generalizable to different backdoor attacks, models, and datasets.

Manipulation by Selecting the Best Intensity. Based on the above remarks, we offer guidelines for attackers to optimize the selection of training-inference trigger intensities in backdoor attacks. Attackers can utilize the TITIM workflow to identify the approximate intensity range that successfully activates the backdoor. Generally, selecting triggers from the top-left corner of the ASR results can help achieve a higher ASR while maintaining stealthiness during both the training and inference phases.

In particular, attackers do not need to fully train models at all intensities to find the optimal one. Instead, they can employ an early-stopping scheme based on loss and ASR trajectories. This is because a backdoored model tends to first learn the backdoor features over the normal features, and so exhibits significant changes of ASR and loss in the early stage. The attackers can further accelerate the process by relaxing the constraint of the poisoning rate since this is only for intensity selection. As illustrated in Figure 13, the difference in ASR

Table 4: Results of (balanced) mixing training triggers of different intensities on BadNets (Square Patch). The poisoning rate is always 10%.

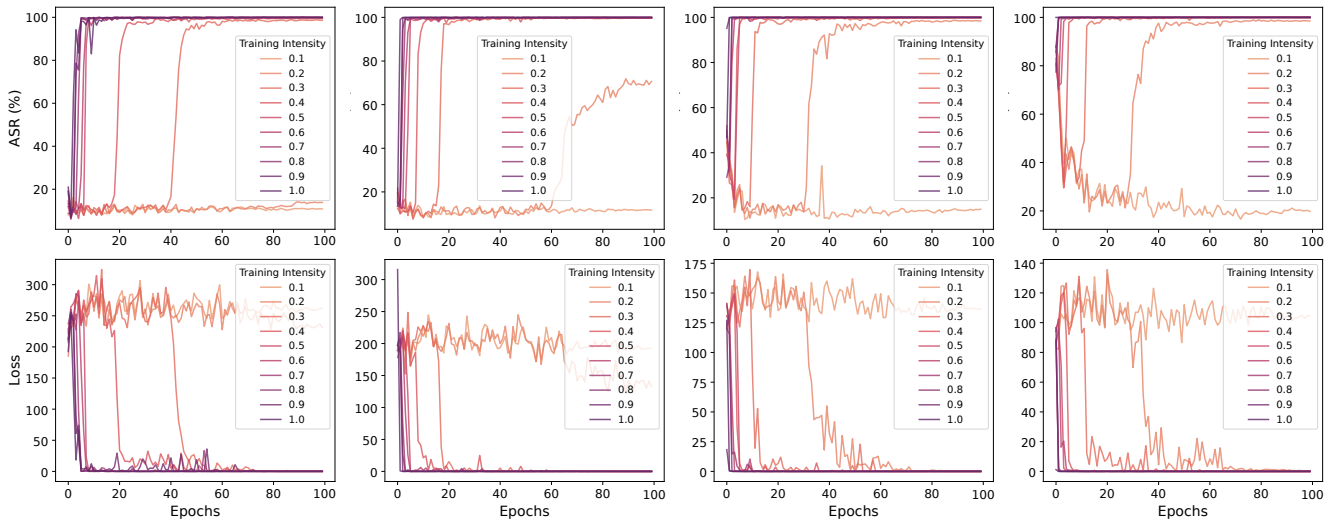
Training Intensity	*Worst* ASR (%)	*Average* ASR (%)
non-mix	10.61	84.67
0.4 + 0.1	89.29	98.76
0.5 + 0.1	79.31	97.47
0.6 + 0.1	87.39	98.26
0.7 + 0.1	**92.77**	**99.08**
0.8 + 0.1	84.32	97.43
0.9 + 0.1	81.24	97.31
1.0 + 0.1	66.98	92.83

or loss across different training intensities becomes more separable when the poisoning rate becomes larger (e.g., 30%). In this case, the required intensity that ensures a successful attack (with a high ASR over 90% and a low loss) can be as low as 0.3 (for Opacity). This setting strikes a good balance between attack effectiveness and stealthiness. Moreover, attackers can implement more effective searching algorithms, *e.g.*binary search, to further reduce overhead.

As demonstrated in Appendix B, the optimal intensities are generally consistent across different model architectures with similar sizes. Thus, adversaries can select intensities based on surrogate models in black-box scenarios. Furthermore, attackers can intentionally employ different trigger intensities between training and inference phases for higher ASR or better stealthiness. These strategies will be explored in more detail in the following sections.

Manipulation by Mixing Intensities. As discussed previously, training-inference triggers with higher intensity are more easily captured by models, while weaker triggers exhibit better generalizability across various intensities. In addition to adopting the trigger intensity selection strategies described above, we demonstrate that the attackers can leverage the advantages of both high and low-intensity triggers by mixing them during data poisoning to improve the ASR.

Firstly, we prepare two series of poisoned training datasets using BadNets: one poisoned with a single training trigger intensity at a poisoning rate of 10%, while the other replaces half of the poisoned samples with lower intensity training



(a) Poisoning Rate 5% (b) Poisoning Rate 10% (c) Poisoning Rate 20% (d) Poisoning Rate 30%

Figure 13: Trends of ASR and Loss for BadNets on CIFAR-10 during training at varying poisoning rates.

triggers (*i.e.*, mixed intensity). Subsequently, we train models on each of the poisoned training sets and collect their worst and average ASR among the poisoned test sets at different intensities.

As shown in Table 4, poisoned models utilizing a mixture of trigger intensities consistently outperform those with a single intensity. More specifically, mixed triggers significantly enhance the ASR on the poisoned dataset with low-intensity triggers, increasing the worst-case from approximately 10% (random guess) to at least 66.98% and exceeding 90% (92.77%) in extreme cases. Note that scenarios with training intensities lower than 0.3 are omitted, as the triggers are too weak to build a backdoor. This finding suggests that attackers can activate a backdoor with lower-intensity training triggers by employing a mixing strategy, which involves combining triggers of different intensities.

Additional experiments on various model architectures reveal a similar pattern, demonstrating the practicality of the intensity-mixing strategy in black-box scenarios. In these cases, attackers can cover required inference intensities with a high ASR by arbitrarily selecting two extreme intensities (*e.g.*, opacity=0.1 and opacity=1.0). Furthermore, attackers can simultaneously mix multiple intensities (*e.g.*, size, and opacity) to further enhance attack effectiveness. Detailed results can be found in Appendix C.

4.4 Intensity Manipulation to Bypass Defenses

Our intensity manipulation allows the attacks to bypass potential defenses. According to the threat model, we categorize the existing defense strategies into five classes and then conduct experiments on several typical defense methods within each category to demonstrate how and to what extent attackers can reduce the effectiveness of these defenses. The details of the defense methods and the corresponding metrics are presented in Table 5.

Data Cleaning Defenses. Data cleaning defenses aim to filter out poisoned samples in the training set. In this context, we apply defenses to poisoned datasets with varying trigger intensities to assess whether weaker training triggers can more effectively bypass these defenses. We implement defense methods on partially poisoned datasets: for Activation Clustering, we collect the silhouette scores of clusters; for Spectral Signature and Anti-Backdoor Learning, we gather the recall metrics on the poisoned dataset. For each attack method, we establish two configurations: one without modifying trigger intensity and another with modifications between the training and inference phases.

As shown in Table 6, the effectiveness of the defenses can be significantly degraded by applying weak triggers to the training data. For example, the Square triggers with intensity modification can significantly reduce the recall of Spectral Signature and Anti-Backdoor Learning from 98.60% and 94.60% to 12.80% (-85.8%) and 12.90% (-81.7%), while maintaining high ASRs of 96.54% and 99.75%. For Activation Clustering, the silhouette score can be reduced from 0.53 to 0.15 (-0.38), causing the defense to fail. These results imply that attackers can use weaker triggers to bypass defenses or human checks on the training dataset and then apply inference triggers of higher intensity to ensure a high attack success rate (ASR).

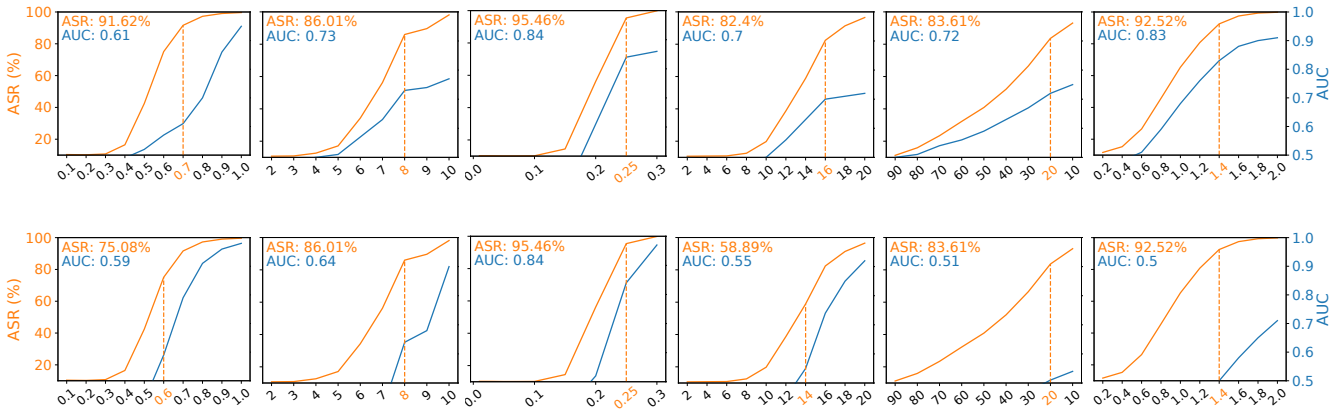
Input Detection Defenses. Input detection defenses aim to identify attacked samples fed into a deployed model during the inference stage. In this context, we first train models using the attack methods listed in Table 1, setting the intensity to

Table 5: Backdoor defenses tested in our work. ↑ (↓) means the higher (lower), the better.

Category	Description	Defense	Metric
Data Cleaning	Identify poisoned samples in the training dataset.	Activation Clustering (AC) [6]	Silhouette score↑
		Spectral Signature (SS) [80]	Recall↑
		Anti-Backdoor Learning (ABL) [49]	Recall↑
Input Detection	Identify poisoned inputs during the inference stage.	STRIP [23]	AUC↑
		Scale-Up [31]	AUC↑
Input Preprocessing	Preprocess poisoned inputs during the inference stage	Februus [13]	ACC drop↓ ASR drop↑
Model Detection	Justify whether a suspicious model is backdoored.	Neural Cleanse (NC) [84]	ASR of reversed trigger (reASR)↑ Anomaly Index↑ L1 Norm of reversed trigger↓
		FeatureRE [88]	ASR of reversed trigger (reASR)↑ Mixed Value↓
Robust Training	Train clean models from poisoned datasets	RAB [90]	ACC drop↓ ASR drop↑

Table 6: Results of data cleaning defenses without/with modifying intensity. Each intensity pair consists of the trigger intensities used during the training and inference phases. Each column of the defense metrics (silhouette score and recall) reflects the effectiveness against attacks and the extent to which the defenses can be weakened by adjusting intensity.

Attacks \ Defenses	Activation Clustering			Spectral Signature			Anti-Backdoor Learning		
	Intensity	ASR (%)	Silhouette ↑ / Δ	Intensity	ASR (%)	Recall ↑ / Δ (%)	Intensity	ASR (%)	Recall ↑ / Δ (%)
BadNets(Square)	(0.9,0.9) / (0.5,1.0)	99.38 / 99.75	0.53 / -0.38	(0.9,0.9) / (0.4,1.0)	99.38 / 96.54	98.60 / -85.80	(1.0,1.0) / (0.5,1.0)	99.65 / 99.75	94.60 / -81.70
BadNets	(10,10) / (5,8)	98.78 / 90.32	0.14 / -0.03	(10,10) / (4,6)	97.78 / 82.29	86.60 / -17.20	(10,10) / (6,8)	97.78 / 93.95	83.56 / -27.72
Blend	(0.3,0.3) / (0.07,0.3)	99.89 / 98.95	0.61 / -0.47	(0.3,0.3) / (0.07,0.1)	99.89 / 94.14	88.00 / -34.32	(0.3,0.3) / (0.05,0.2)	99.89 / 94.12	95.04 / -94.76
SIG	(20,20) / (10,20)	96.52 / 96.51	0.46 / -0.22	(18,18) / (10,20)	94.67 / 96.51	83.28 / -19.08	(20,20) / (10,20)	96.52 / 96.51	75.40 / -64.40
Compress	(30,30) / (60,40)	94.92 / 94.45	0.24 / -0.09	(50,50) / (80,40)	93.57 / 93.19	69.88 / -17.40	(30,30) / (80,40)	94.92 / 93.19	69.36 / -69.22
WaNet	(2.0,2.0) / (0.6,2.0)	99.83 / 99.86	0.58 / -0.13	(1.8,1.8) / (0.4,2.0)	99.66 / 72.78	85.96 / - 9.92	-	-	-

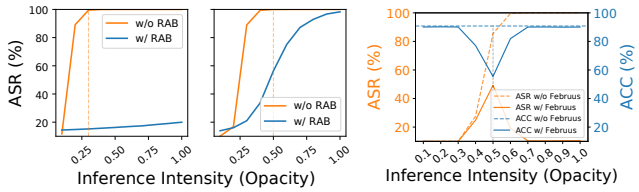


(a) BadNets (Square) (b) BadNets (c) Blended (d) SIG (e) Compress (f) WaNet

Figure 14: Results of input detection defense against different attacks. The upper and lower rows represent Scale-Up and STRIP, respectively.

the highest level specified. We then evaluate datasets with different intensities (comprising half-clean and half-poisoned samples) and collected their ASR and AUC for the defense methods.

The results are shown in Figure 14. As the intensity of the inference triggers used in the evaluation datasets decreases, the performance of the defenses exhibits a more rapid upward trend compared to the attacks. For example, a backdoored



(a) RAB (b) Februus

Figure 15: (a) Results of RAB against BadNets at two training intensities (left: opacity=0.3, right: opacity=0.5) with varying inference-phase trigger intensities. (b) ASR and Acc on sanitized inputs at varied intensities of Februus, which achieves unsatisfactory results around opacity 0.5.

Table 7: Results of model detection defenses against BadNets with different intensities. TI and AI denote the trigger intensity and anomaly index, respectively.

TI	ASR (%)	Neural Cleanse			FeatureRE	
		reASR ↑ (%)	AI ↑	L1 Norm ↓	reASR ↑ (%)	Mixed Value ↓
0.2	9.75	100.0(±1.0)	2.1(± 1.3)	593.4(± 1.9)	97.22(±2.33)	-0.89(±0.05)
0.3	11.75	100.0(±1.0)	1.8(± 0.9)	592.6(± 7.4)	95.83(±3.61)	-0.89(±0.02)
0.4	67.66	100.0(±1.0)	25.1(± 5.7)	405.4(± 63.5)	98.83(±1.17)	-0.93(±0.05)
0.5	95.07	99.99(±1.0)	70.1(±59.3)	260.9(± 86.9)	92.83(±1.05)	-0.80(±0.04)
0.6	98.33	100.0(±1.0)	50.1(±18.5)	212.4(± 58.7)	92.50(±3.27)	-0.70(±0.16)
0.7	99.24	100.0(±1.0)	39.6(±21.5)	240.8(± 80.8)	95.28(±4.72)	-0.53(±0.31)
0.8	99.21	99.99(±1.0)	41.8(±24.4)	279.4(± 86.0)	94.83(±5.17)	-0.71(±0.09)
0.9	99.38	100.0(±1.0)	39.8(±12.1)	259.8(±137.3)	94.11(±5.89)	-0.66(±0.14)
1.0	99.65	100.0(±1.0)	29.8(±16.5)	341.0(±116.8)	89.77(±0.89)	-0.57(±0.06)

model formed with a Square training trigger intensity of 1.0 can still be activated with a high ASR of 91.62% by using a poisoned dataset with an inference trigger intensity of 0.7. Meanwhile, the AUC of STRIP and Scale-Up can be reduced from 0.99 and 0.96 to 0.80 and 0.62, respectively. This indicates that attackers can employ triggers with relatively lower intensity during the inference phase than in the training phase, thereby bypassing input sample detection while maintaining the activatability of the backdoor attack.

Input Preprocessing Defenses. Input preprocessing defenses try to purify poisoned input samples during the inference stage, rendering them incapable of activating the backdoor. We test the Februus defense [13] with a model trained with the flower trigger at opacity 1.0. As illustrated in Figure 15b, Februus cannot consistently achieve satisfactory results, with about 50% of ASR and Acc on sanitized inputs at opacity 0.5.

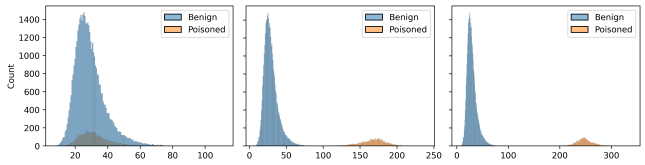
Model Detection Defenses. Model detection defenses assess whether a suspicious model is infected through reverse engineering or a meta-classifier. We evaluate backdoored models with varying training trigger intensities using two typical trigger-reversing defenses: Neural Cleanse [84] and FeatureRE [88]. In this context, the defender assumes access to the model and a small dataset containing benign samples of each label. Note that the Mixed Value of FeatureRE is a metric from its source code that incorporates ASR, the similarity between poisoned and benign samples, the cosine similarity of their activations, the standard deviation of poisoned sample activations, and the size of the reversed mask.

As shown in Table 7, reverse engineering is effective on models trained with weaker triggers, indicating that backdoors formed with weaker training triggers are more easily detected. Both Neural Cleanse and FeatureRE defenses can successfully obtain a reversed trigger even in cases where the original trigger fails to execute the attack, *i.e.*, when the trigger intensities are lower than 0.4. However, backdoors injected with higher intensity training triggers exhibit greater backdoor exclusivity, as they can only be activated within a narrower range of intensities, making them more challenging to be reverse engineerd [64]. For example, the reverse engineering results of Neural Cleanse show a downward trend as the trigger intensity increases. In the worst cases, FeatureRE consistently reports a backdoored model trained with a trigger intensity of 1.0 as benign with a mixed value of -0.57 (above the threshold of -0.75). We attribute this to the regularizations (*e.g.*, the ℓ_1 norm of the trigger mask and the similarity between the poisoned and original images)), which is integral to the optimization process of most trigger reverse-engineering methods.

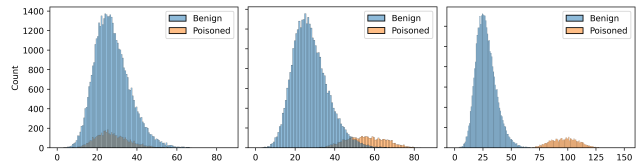
These findings suggest that attackers may circumvent such defenses by simply employing training triggers with higher intensity, thereby prompting defenders to develop adaptive defenses.

Robust Training Defenses. Robust training defenses aim to train robust models on untrusted data and obtain a robust bound R. They ensure that as long as triggers employed in test instances remain within an L_p-ball of radius R, the output of robust models is guaranteed to be consistent with benign models. We evaluate a state-of-the-art defense of this type, namely RAB [90], which first adds noise sampled from a smoothing distribution to the original training dataset to create a large number of "smoothed" training datasets. It then trains models on these datasets and aggregates their final outputs as the final "smoothed" prediction.

Due to its extremely high overhead for training a large number of models, we only evaluate RAB on CIFAR-10 with ResNet-18. We train 1,000 models on each poisoned training dataset and evaluate them on poisoned inference datasets with varying trigger intensities. As shown in Figure 15a, RAB successfully strengthens backdoored models with lower training intensity (*i.e.*, opacity 0.3) and reduces the ASR by 80% across all inference intensities. However, it fails on higher training intensities since the ASR of high-intensity inference triggers remains higher than 98%. Moreover, we notice that RAB inevitably sacrifices the model Acc by above 10%.



(a) Training intensity (opacity)=0.4.



(b) Training intensity (opacity)=1.0.

Figure 16: Neuron activation distribution of two training intensities and three inference intensities (from left to right: opacity=0.1, 0.5, 1.0, respectively; blue: benign, orange: poisoned).

5 Discussion

5.1 Explanations of Our Observations

In this section, we explain our observations regarding training-inference trigger generalization and overfitting through neuron activations and UMAP visualizations.

Neuron Activations. We train a backdoored model with a single training intensity and then identify the compromised neurons employing NONE [87]. Subsequently, we input poisoned samples with varying inference intensities into the model to collect the activation values of identified compromised neurons and then compare their distribution with that of benign samples.

As illustrated in Figure 16, trigger intensity has a monotonic relationship with neuron activation values. Specifically, as the inference intensity increases, the activation distribution of compromised neurons between poisoned and benign samples becomes more separable as the trigger intensity increases. This explains the high ASR in the generalization phenomenon. In contrast, as the training intensity increases, the distribution becomes less separable. This explains the low ASR in the overfitting phenomenon. We find that when replacing the ReLU function with Sigmoid, the above explanations still hold.

UMAP Visualizations. We use UMAP [59] to visualize the activations of the fully connected layer inputs of a backdoored model trained with trigger opacity=0.5. We first construct an embedding space based on the eigenvectors of the benign dataset. Subsequently, we extract the activation values from two poisoned datasets attacked by BadNets with two inference trigger intensities opacity=0.5 and opacity=1.0. Finally, we embed these poisoned eigenvectors into the previously constructed embedding space.

Table 8: Anomaly Index (AI) and ℓ_1 Norm (L1) of the reversed triggers for the original and adaptive version of Neural Cleanse (NC) against BadNets with different intensities; TI denotes the trigger intensity.

TI		0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
ASR (%)		9.75	11.75	67.66	95.07	98.33	99.24	99.21	99.38	99.65
NC	AI↑	2.8	3.2	16.4	26.3	43.7	80.0	22.4	25.6	21.7
	L1↓	592.2	587.6	392.6	165.8	172.1	175.1	193.5	334.1	274.6
Adap-NC	AI↑	5.0	4.7	3.5	9.9	0.5	3.9	6.3	4.7	4.7
	L1↓	489.9	465.5	305.3	87.6	144.2	167.4	133.1	327.7	262.9

As shown in Figure 17, when the training intensity is fixed with opacity= 1.0, poisoned inference samples with higher trigger intensity (*i.e.*, opacity= 1.0) exhibit a more concentrated distribution than opacity= 0.5 within the embedding space. This suggests that activation values with higher inference intensities may reside in a subspace (characterized by a high ASR) within the broader space of low-intensity activation values, explaining the better generalizability of lower-intensity triggers. Conversely, the lower inference intensity causes the activation values to form a similar distribution to that of benign samples, *i.e.*, the trigger is not sufficient to activate the backdoor. This explains the low ASR for the overfitting phenomenon of higher training intensities.

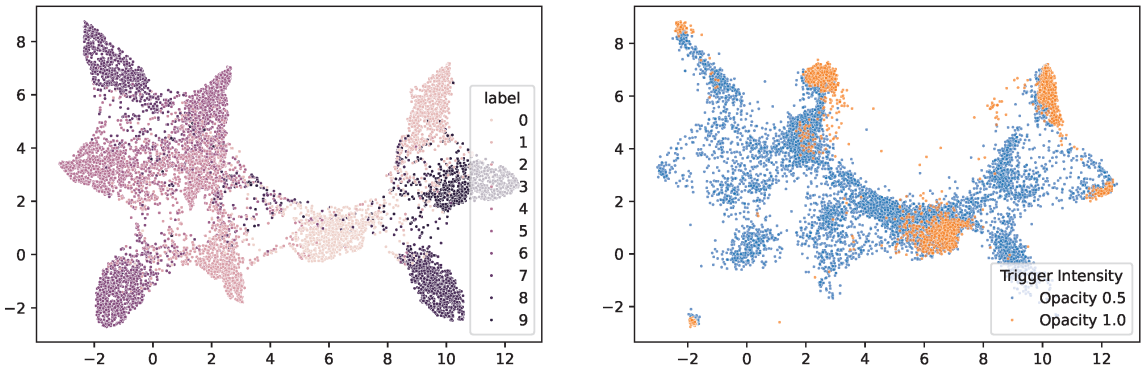
5.2 Adaptive Defenses

In this section, we discuss two possible adaptive defenses, namely Adaptive Neural Cleanse and Multi-Stage Defenses.

Adaptive Neural Cleanse. We implement an adaptive version of Neural Cleanse (Adap-NC), which computes the losses for both the reverse-engineered trigger and its lower-opacity version. As shown in Table 8, the reversing effectiveness of Adap-NC has improved as expected (the ℓ_1 norm of the reversed pattern is smaller), particularly for lower-intensity triggers. However, this improvement comes at the cost of a corresponding drop in the Anomaly Index, as it reverses smaller patterns for all targets, making the overall enhancement modest. Moreover, the necessity of knowing the attacker's choice of intensity renders it impractical.

Multi-Stage Defense. As mentioned in Section 4.4, attackers can exploit the intensity mismatch to bypass defenses, potentially leading to severe consequences in real-world applications. Thus, a single-stage defense mechanism may not be sufficient to protect against such adaptive attacks, as attackers can easily tune the intensity of triggers targeting a single defense at particular stages.

For the design of new defenses, the defenders shall consider integrating mechanisms at different stages to mitigate the impact of intensity mismatch or varying intensity triggers. Defenders can also incorporate multiple existing defenses at different stages; however, this may introduce challenges such as performance degradation due to interactions between de-



(a) Activation visualization of benign images.

(b) Activation visualization of poisoned images.

Figure 17: UMAP visualization for benign and varying intensity samples.

fenses. For instance, after applying an incomplete data cleaning defense, the model might learn a weaker backdoor, potentially making it harder to detect by subsequent defenses. More discussion on this topic is left for our future work.

6 Related Work

In this section, we introduce some existing works on backdoor attacks and defenses, as well as the backdoor studies related to trigger mismatch.

Backdoor Attacks. Backdoor attack methods can be roughly divided into poisoning-based, weight-oriented, and structure-modified attacks [51].

Data poisoning basically carries out backdoor attacks by manipulating the training data. BadNets [28] pioneered the concept of backdoor attacks and exposed the threats in DNN training, i.e., exactly matched trigger between the training and inference phase can cause the victim model to exhibit incorrect behavior. The following methods mainly focus on generating triggers with a higher attack success rate and stealthiness, consisting of visible trigger [28, 32], invisible trigger [2, 4, 8, 14, 15, 19, 21, 22, 40, 46, 48, 56, 60, 66, 94, 102, 106], style change trigger [10, 54], frequency domain trigger [33, 85], clean label attacks [3, 68, 81, 105], and others according to the different types of triggers.

Weight-oriented attacks modify model parameters directly instead of training the whole model on poisoned datasets [20, 24, 67, 104]. Structure-modified attacks literally inject backdoors by modifying the model structure, *e.g.*, inserting a backdoored module [47, 78] or replacing a narrow subnet [62]. Since we mainly focus on the impact of trigger intensity mismatch between the training and inference phases, and weight-oriented attacks and structure-modified attacks do not involve trigger intensity during the training phase. Our research does not cover these two types of attack methods.

Backdoor Defenses. To ensure the security of the model, researchers put forward various methods of backdoor defense. We roughly divide them into data cleaning backdoor defense, model detection backdoor defense, and input detection backdoor defense.

Data cleaning defenses focus on identifying abnormal training samples or apply normalization to them [6, 11, 34, 77, 79, 80, 101]. Model detection defenses utilize gradient descent [7, 54, 70, 75, 84, 93, 95] or generative adversarial network (GAN) [26, 63, 107] to obtain potential trigger patterns and then judge the backdoor by the pattern properties, *e.g.*, size and attack success rate. Note that there are some implementations in black-box settings in which defenders can reverse trigger patterns even when they only have access to model predictions (logits or labels) [1, 16, 30]. Input detection backdoor defense aims at filtering malicious inputs after deploying the model, and thus preventing the backdoor from being triggered [18, 23, 39, 41].

There are also some methods try to eliminate backdoor through retraining [43, 45, 57, 100], backdoor-related neurons pruning [53, 91], or knowledge distillation [25, 36, 50, 98], which are beyond this research and left for our future work.

Mismatched Triggers. Some prior works also studied the generalizability of backdoor triggers. [63] attacked a model trained on the CIFAR-10 dataset with a 3×3 trigger. By repeating the reverse engineering process with random seeds, they found that the reversed triggers formed a continuous set in the pixel space of all possible 3×3 triggers, indicating that the injected backdoor exhibits certain generalizability beyond the original trigger pattern. [74, 75] also showed that, given access to a backdoored model, one can reliably generate multiple alternative triggers without accessing the training dataset or the original trigger, suggesting that an injected backdoor trigger pattern usually generalize to other shapes or colors. Additionally, other trigger-reversing studies [16, 30, 38] have reported similar phenomena. [64] defined perturbed triggers while capable of activating backdoor as fuzzy triggers. They also proposed a trigger upper bound that measures how much perturbation a trigger can endure while retaining its ability to

activate the backdoor, which implies triggers' generalization.

The above work showed that triggers can vary between training and inference phases, while there are no explicit restrictions on their difference other than ℓ_n norms. Our works distinguish from existing studies with a semantic restriction, i.e. focusing on the impact of trigger intensity on backdoor attacks, and thoroughly investigated the impact of intensity on the effectiveness and generalizability of backdoor attacks.

7 Conclusion and Outlook

In this paper, we have systematically investigated the trigger intensity mismatch training and inference. Based on our new workflow TITIM (Training-Inference Trigger Intensity Manipulation), we reveal that consistent trigger configurations between the training and inference phases may indeed not be optimal for backdoor attacks; Instead, they may be leveraged to improve attacks, even when defenses are present. Extensive experiments demonstrate that our findings hold across various attack methods, trigger configurations, model architectures, datasets, etc. We have also explained the reasons behind our findings and discussed their implications for adapting backdoor defenses.

Future work should explore optimization-based backdoors [15, 55, 89], which generally exploit unpredictable sample-wise triggers, beyond the dynamic triggers (*e.g.*, WaNet, and Compress) we have considered. Although we have shown that using the interpolation between the original image and its poisoned version as the intensity is practical, a "universal intensity metric" to incorporate different attacks is still worth exploring. Additionally, our findings regarding trigger mismatch should be further evaluated in more learning paradigms, such as federated learning and reinforcement learning, which assume distributed and interactive training process, respectively.

Acknowledgments

This research is supported by the National Key Research and Development Program of China (2023YFE0209800), the National Natural Science Foundation of China (T2341003, 62376210, 62406240, 62161160337, 62132011, U24B20185, U21B2018, U20A20177, U244120060, 62206217), the Shaanxi Province Key Industry Innovation Program (2023-ZDLGY-38).

Ethics Considerations

Our paper proposes a new adaptive backdoor attack strategy under train-inference intensity mismatched triggers, that can bypass several defenses. It can be used to adversarially manipulate the behavior of a model even in the presence of a specific defense mechanism.

We admit that the proposed attack strategy can be used for malicious purposes, however, it is unlikely to cause severe harm to individuals or society, since selecting the optimal trigger for each setting would require exponentially more resources than training. Moreover, the phenomenon of train-inference intensity mismatch can help form a foundation for developing more robust defenses against backdoor attacks, which can be beneficial to the machine learning community in the long term.

Open Science

The source code and datasets are available on GitHub (https://github.com/cv12ha0/TITIM) and Zenodo (https://zenodo.org/records/14729436).

References

[1] W. Aiken, H. Kim, S. S. Woo, and J. Ryoo. Neural network laundering: Removing black-box backdoor watermarks from deep neural networks. *Comput. Secur.*, 2021.

[2] E. Bagdasaryan and V. Shmatikov. Blind backdoors in deep learning models. *USENIX Security*, 2021.

[3] M. Barni, K. Kallas, and B. Tondi. A new backdoor attack in CNNS by training set corruption without label poisoning. *IEEE ICIP*, 2019.

[4] H. Cai, P. Zhang, H. Dong, Y. Xiao, S. Koffas, and Y. Li. Toward stealthy backdoor attacks against speech recognition via elements of sound. *IEEE TIFS*, 2024.

[5] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. *IEEE S&P*, 2017.

[6] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. M. Molloy, and B. Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *AAAI Workshops*, 2019.

[7] H. Chen, C. Fu, J. Zhao, and F. Koushanfar. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. *IJCAI*, 2019.

[8] X. Chen, C. Liu, B. Li, K. Lu, and D. Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv*, 2017.

[9] X. Chen, A. Salem, D. Chen, M. Backes, S. Ma, Q. Shen, Z. Wu, and Y. Zhang. Badnl: Backdoor attacks against NLP models with semantic-preserving improvements. *ACSAC*, 2021.

[10] S. Cheng, Y. Liu, S. Ma, and X. Zhang. Deep feature space trojan attack of neural networks by controlled detoxification. *AAAI*, 2021.

[11] E. Chou, F. Tramèr, and G. Pellegrino. Sentinet: Detecting localized universal attacks against deep learning systems. *IEEE S&P Workshops*, 2020.

[12] L. Deng. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag.*, 2012.

[13] B. G. Doan, E. Abbasnejad, and D. C. Ranasinghe. Februus: Input purification defense against trojan attacks on deep neural network systems. *ACSAC*, 2020.

[14] K. D. Doan, Y. Lao, and P. Li. Backdoor attack with imperceptible input and latent modification. *NeurIPS*, 2021.

[15] K. D. Doan, Y. Lao, W. Zhao, and P. Li. LIRA: learnable, imperceptible and robust backdoor attacks. *ICCV*, 2021.

[16] Y. Dong, X. Yang, Z. Deng, T. Pang, Z. Xiao, H. Su, and J. Zhu. Black-box detection of backdoor attacks with limited information and data. *ICCV*, 2021.

[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[18] M. Du, R. Jia, and D. Song. Robust anomaly detection and backdoor attack detection via differential privacy. *ICLR*, 2020.

[19] Q. Duan, Z. Hua, Q. Liao, Y. Zhang, and L. Y. Zhang. Conditional backdoor attack via JPEG compression. *AAAI*, 2024.

[20] J. Dumford and W. J. Scheirer. Backdooring convolutional neural networks via targeted weight perturbations. *IEEE IJCB*, 2020.

[21] Y. Gao, Y. Li, X. Gong, Z. Li, S.-T. Xia, and Q. Wang. Backdoor attack with sparse and invisible trigger. *IEEE TIFS*, 2024.

[22] Y. Gao, Y. Li, L. Zhu, D. Wu, Y. Jiang, and S.-T. Xia. Not all samples are born equal: Towards effective clean-label backdoor attacks. *Pattern Recognition*, 2023.

[23] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal. STRIP: a defence against trojan attacks on deep neural networks. *ACSAC*, 2019.

[24] S. Garg, A. Kumar, V. Goel, and Y. Liang. Can adversarial weight perturbations inject neural backdoors. *ACM CIKM*, 2020.

[25] X. Gong, Y. Chen, W. Yang, Q. Wang, Y. Gu, H. Huang, and C. Shen. Redeem myself: Purifying backdoors in deep learning models using self attention distillation. *IEEE S&P*, 2023.

[26] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. *NeurIPS*, 2014.

[27] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015.

[28] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 2019.

[29] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 2019.

[30] J. Guo, A. Li, and C. Liu. AEVA: black-box backdoor detection using adversarial extreme value analysis. *ICLR*, 2022.

[31] J. Guo, Y. Li, X. Chen, H. Guo, L. Sun, and C. Liu. SCALE-UP: an efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. *ICLR*, 2023.

[32] W. Guo, B. Tondi, and M. Barni. A master key backdoor for universal impersonation attack against dnn-based face verification. *Pattern Recognit. Lett.*, 2021.

[33] H. A. A. K. Hammoud and B. Ghanem. Check your other door! creating backdoor attacks in the frequency domain. *BMVC*, 2022.

[34] J. Hayase, W. Kong, R. Somani, and S. Oh. SPECTRE: defending against backdoor attacks using robust statistics. *arXiv*, 2021.

[35] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. *ECCV*, 2016.

[36] G. E. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv*, 2015.

[37] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *arXiv*, 2017.

[38] X. Hu, X. Lin, M. Cogswell, Y. Yao, S. Jha, and C. Chen. Trigger hunting with a topological prior for trojan detection. *ICLR*, 2022.

[39] M. Javaheripi, M. Samragh, G. Fields, T. Javidi, and F. Koushanfar. Cleann: Accelerated trojan shield for embedded neural networks. *IEEE/ACM ICCAD*, 2020.

[40] W. Jiang, H. Li, G. Xu, and T. Zhang. Color backdoor: A robust poisoning attack in color space. *CVPR*, 2023.

[41] K. Jin, T. Zhang, C. Shen, Y. Chen, M. Fan, C. Lin, and T. Liu. A unified framework for analyzing and detecting malicious examples of DNN models. *arXiv*, 2020.

[42] K. Karra, C. Ashcraft, and N. Fendley. The trojai software framework: An opensource tool for embedding trojans into deep learning models. *arXiv*, 2020.

[43] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *arXiv*, 2016.

[44] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[45] B. Li, Y. Cai, J. Cai, Y. Li, H. Qiu, R. Wang, and T. Zhang. Purifying quantization-conditioned backdoors via layer-wise activation correction with distribution approximation. *ICML*, 2024.

[46] S. Li, M. Xue, B. Z. H. Zhao, H. Zhu, and X. Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Trans. Dependable Secur. Comput.*, 2021.

[47] Y. Li, J. Hua, H. Wang, C. Chen, and Y. Liu. Deeppayload: Black-box backdoor attack on deep learning models through neural payload injection. *IEEE/ACM ICSE*, 2021.

[48] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu. Invisible backdoor attack with sample-specific triggers. *ICCV*, 2021.

[49] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma. Anti-backdoor learning: Training clean models on poisoned data. *NeurIPS*, 2021.

[50] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *ICLR*, 2021.

[51] Y. Li, B. Wu, Y. Jiang, Z. Li, and S. Xia. Backdoor learning: A survey. *arXiv*, 2020.

[52] Y. Li, T. Zhai, Y. Jiang, Z. Li, and S. Xia. Backdoor attack in the physical world. *arXiv*, 2021.

[53] K. Liu, B. Dolan-Gavitt, and S. Garg. Fine-pruning: Defending against backdooring attacks on deep neural

networks. *RAID*, 2018.

[54] Y. Liu, W. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang. ABS: scanning neural networks for back-doors by artificial brain stimulation. *ACM CCS*, 2019.

[55] Y. Liu, S. Ma, Y. Aafer, W. Lee, J. Zhai, W. Wang, and X. Zhang. Trojaning attack on neural networks. *NDSS*, 2018.

[56] Y. Liu, X. Ma, J. Bailey, and F. Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. *ECCV*, 2020.

[57] Y. Liu, Y. Xie, and A. Srivastava. Neural trojans. *IEEE ICCD*, 2017.

[58] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. *IEEE ICCV*, 2015.

[59] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv*, 2018.

[60] T. A. Nguyen and A. T. Tran. Wanet - imperceptible warping-based backdoor attack. *ICLR*, 2021.

[61] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. *NeurIPS*, 2017.

[62] X. Qi, J. Zhu, C. Xie, and Y. Yang. Subnet replacement: Deployment-stage backdoor attack against deep neural networks in gray-box setting. *arXiv*, 2021.

[63] X. Qiao, Y. Yang, and H. Li. Defending neural backdoors via generative distribution modeling. *NeurIPS*, 2019.

[64] H. Qiu, J. Sun, M. Zhang, X. Pan, and M. Yang. BELT: old-school backdoor attacks can evade the state-of-the-art defense with backdoor exclusivity lifting. *arXiv*, 2023.

[65] H. Qiu, Y. Zeng, S. Guo, T. Zhang, M. Qiu, and B. Thuraisingham. Deepsweep: An evaluation framework for mitigating DNN backdoor attacks using data augmentation. *ACM Asia CCS*, 2021.

[66] E. Quiring and K. Rieck. Backdooring and poisoning neural networks with image-scaling attacks. *IEEE S&P Workshops*, 2020.

[67] A. S. Rakin, Z. He, and D. Fan. TBT: targeted neural network attack with bit trojan. *CVPR*, 2020.

[68] A. Saha, A. Subramanya, and H. Pirsiavash. Hidden trigger backdoor attacks. *AAAI*, 2020.

[69] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *CVPR*, 2018.

[70] G. Shen, Y. Liu, G. Tao, S. An, Q. Xu, S. Cheng, S. Ma, and X. Zhang. Backdoor scanning for deep neural networks through k-arm optimization. *ICML*, 2021.

[71] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[72] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 2012.

[73] M. Subedar, N. A. Ahuja, R. Krishnan, I. J. Ndiour, and O. Tickoo. Deep probabilistic models to detect data poisoning attacks. *arXiv*, 2019.

[74] M. Sun, S. Agarwal, and J. Z. Kolter. Poisoned classifiers are not only backdoored, they are fundamentally broken. *arXiv*, 2020.

[75] M. Sun and Z. Kolter. Single image backdoor inversion via robust smoothed classifiers. *CVPR*, 2023.

[76] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *ICML*, 2019.

[77] D. Tang, X. Wang, H. Tang, and K. Zhang. Demon in the variant: Statistical analysis of dnns for robust backdoor contamination detection. *USENIX Security*, 2021.

[78] R. Tang, M. Du, N. Liu, F. Yang, and X. Hu. An embarrassingly simple approach for trojan attack in deep neural networks. *ACM KDD*, 2020.

[79] R. Tang, J. Yuan, Y. Li, Z. Liu, R. Chen, and X. Hu. Setting the trap: Capturing and defeating backdoor threats in plms through honeypots. *NeurIPS*, 2023.

[80] B. Tran, J. Li, and A. Madry. Spectral signatures in backdoor attacks. *NeurIPS*, 2018.

[81] A. Turner, D. Tsipras, and A. Madry. Label-consistent backdoor attacks. *arXiv*, 2019.

[82] S. Udeshi, S. Peng, G. Woo, L. Loh, L. Rawshan, and S. Chattopadhyay. Model agnostic defence against backdoor attacks in machine learning. *IEEE Trans. Reliab.*, 2022.

[83] M. Villarreal-Vasquez and B. K. Bhargava. Confoc: Content-focus protection against trojan attacks on neural networks. *arXiv*, 2020.

[84] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. *IEEE S&P*, 2019.

[85] T. Wang, Y. Yao, F. Xu, S. An, H. Tong, and T. Wang. An invisible black-box backdoor attack through frequency domain. *ECCV*, 2022.

[86] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 2004.

[87] Z. Wang, H. Ding, J. Zhai, and S. Ma. Training with more confidence: Mitigating injected and natural backdoors during training. *NeurIPS*, 2022.

[88] Z. Wang, K. Mei, H. Ding, J. Zhai, and S. Ma. Rethinking the reverse-engineering of trojan triggers. *NeurIPS*, 2022.

[89] Z. Wang, J. Zhai, and S. Ma. Bppattack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. *CVPR*, 2022.

[90] M. Weber, X. Xu, B. Karlas, C. Zhang, and B. Li. RAB: provable robustness against backdoor attacks. *IEEE S&P*, 2023.

[91] D. Wu and Y. Wang. Adversarial neuron pruning purifies backdoored deep models. *NeurIPS*, 2021.

[92] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang. Cvt: Introducing convolutions to vision transformers. *ICCV*, 2021.

[93] Z. Xiang, D. J. Miller, and G. Kesidis. Detection of backdoors in trained classifiers without access to the training set. *IEEE Trans. Neural Netw. Learn. Syst.*, 2022.

[94] Q. Xiao, Y. Chen, C. Shen, Y. Chen, and K. Li. Seeing is not believing: Camouflage attacks on image scaling algorithms. *USENIX Security*, 2019.

[95] X. Xu, K. Huang, Y. Li, Z. Qin, and K. Ren. Towards reliable and efficient backdoor trigger inversion via decoupling benign features. *ICLR*, 2024.

[96] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li. Detecting AI trojans using meta neural analysis. *IEEE S&P*, 2021.

[97] M. Xue, C. He, Y. Wu, S. Sun, Y. Zhang, J. Wang, and W. Liu. PTB: robust physical backdoor attacks against deep neural networks in real world. *Comput. Secur.*, 2022.

[98] K. Yoshida and T. Fujino. Disabling backdoor and identifying poison data by using knowledge distillation in backdoor attacks on deep neural networks. *ACM Workshop on Artificial Intelligence and Security*, 2020.

[99] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. H. Tay, J. Feng, and S. Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *ICCV*, 2021.

[100] Y. Zeng, S. Chen, W. Park, Z. Mao, M. Jin, and R. Jia. Adversarial unlearning of backdoors via implicit hypergradient. *ICLR*, 2022.

[101] Y. Zeng, W. Park, Z. M. Mao, and R. Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. *ICCV*, 2021.

[102] J. Zhang, D. Chen, Q. Huang, J. Liao, W. Zhang, H. Feng, G. Hua, and N. Yu. Poison ink: Robust and invisible backdoor attack. *IEEE TIP*, 2022.

[103] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. *CVPR*, 2018.

[104] Z. Zhang, L. Lyu, W. Wang, L. Sun, and X. Sun. How to inject backdoors with better consistency: Logit anchoring on clean data. *ICLR*, 2022.

[105] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, and Y. Jiang. Clean-label backdoor attacks on video recognition models. *CVPR*, 2020.

[106] H. Zhong, C. Liao, A. C. Squicciarini, S. Zhu, and D. J. Miller. Backdoor embedding in convolutional neural network models via invisible perturbation. *ACM CODASPY*, 2020.

[107] L. Zhu, R. Ning, C. Wang, C. Xin, and H. Wu. Gangsweep: Sweep out neural backdoors by GAN. *ACM MM*, 2020.

A Case Study: BadNets on MNIST

Our initial series of experiments employ the BadNets [29] approach on MNIST [12], utilizing the trigger opacity as an intuitive metric to quantify the intensity of the trigger.

Setup. We adopt the configuration employed by [29] to establish our network architecture and training parameters. The backdoor trigger is set to a 3×3 pixel patch located at the bottom right corner, as shown in Figure 18. The Architecture of the baseline network for this task is shown in Table 3.

Results. We conduct two sets of experiments on MNIST, where the opacity of the trigger is set to 0.5 during both the training and inference phases, respectively. Then we examine the impact on the attack success rate when the trigger's opacity is adjusted during the other stage. Here, we measure the trigger intensity with its opacity.

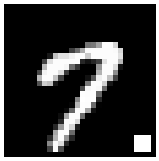


Figure 18: The backdoor trigger for MNIST.

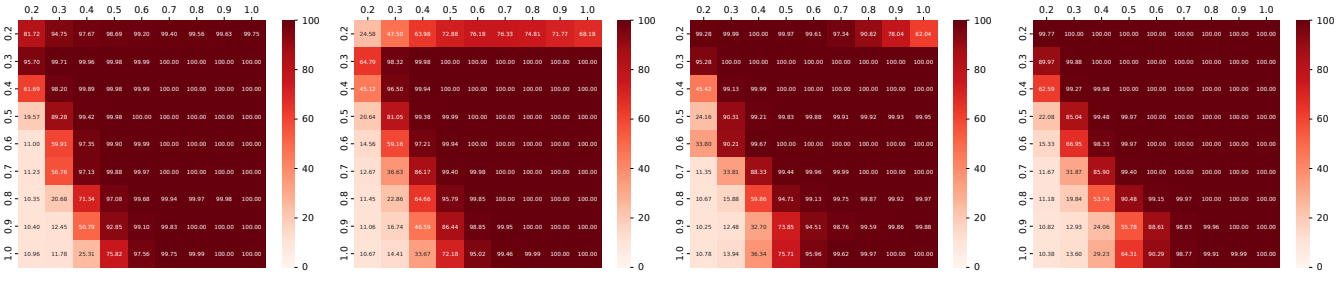
We train three models on a poisoned dataset with the trigger opacity set to 0.25, 0.5, and 0.75 and then evaluate its performance across 20 distinct poisoned datasets, each characterized by varying levels of trigger opacity. The results are shown in Figure 12 (left). The result is in line with our experience: the backdoor is successfully activated when the trigger intensity of the test reaches that of the training set, and the attack success rate increases as the trigger intensity is augmented. In Figure 12 (right), we train 20 models attacked by different intensities of the backdoor and evaluate them on three poisoned test sets. The attack success rate firstly goes up as the trigger is injected successfully but then drops significantly. In similar repeated experiments, the ASR dropping consistently occurs after a certain point, which means injecting a more potent backdoor does not always yield a better attack result.

Additional Analysis of ASR Dropping. To gain further insights into the ASR dropping phenomenon, we conduct cross-testing on each pair of trigger configurations for the training/inference stages mentioned above. Figure 5 illustrates the attack success rate across each pair of configurations in two phases, wherein each set employs an identical white square as the trigger pattern and varies only in the trigger opacity. The ASR always appears to drop when the attack intensity of the training set exceeds that of the test set, this observation implies that models may be overfitting to the trigger pattern with a higher intensity.

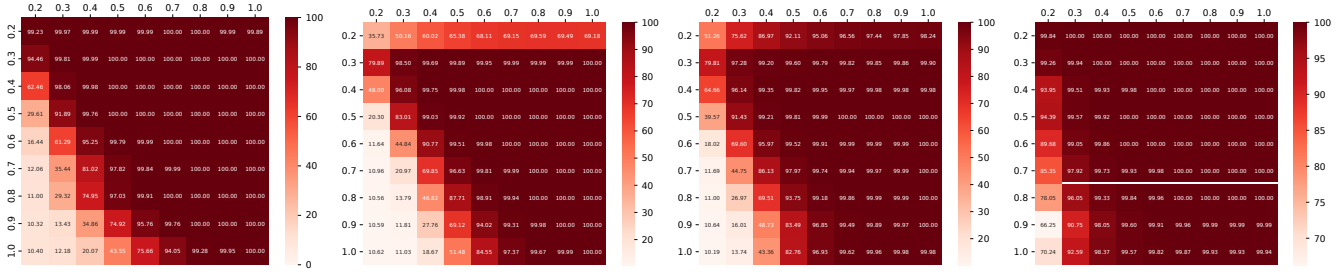
B Results on More Model Architectures

In this section, we additionally verify the generalization of our work to various architectures on 4 CNNs and 3 ViTs: for CNN, we tested VGG-16 [71], EfficientNet B0 [76], MobileNet V2 [69], and SENet-18 [37]; and for ViT, we tested SimpleViT [17], CvT [92], and T2T ViT [99] We adopt the BadNets with square triggers, as noted in Table 1, and set the poisoning rate to 10% for the following experiments.

As shown in Figure 19, the ASR distribution across varying training- and inference-phase trigger intensity settings is mostly consistent, except for SimpleViT, which exhibits a consistently higher ASR than other models. While the overall pattern aligns with our previous findings, this is reasonable

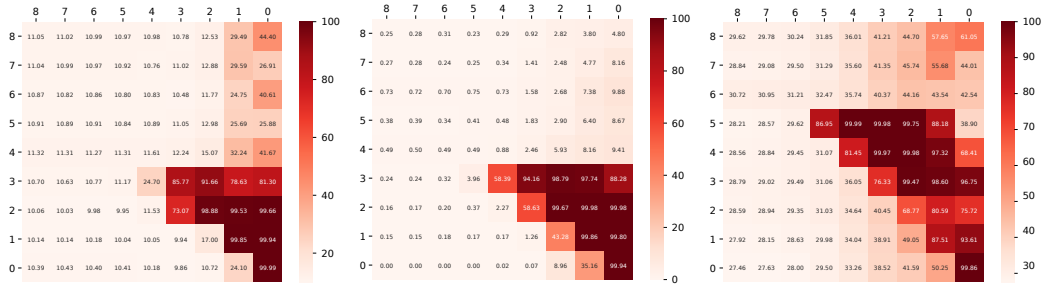


(a) ResNet-18 (b) VGG-16 (c) EfficientNet B0 (d) MobileNet V2



(e) SENet-18 (f) CvT (g) T2T ViT (h) SimpleViT

Figure 19: Results of BadNets with Square patch on different model architectures.



(a) CIFAR-10 (b) GTSRB (c) CelebA

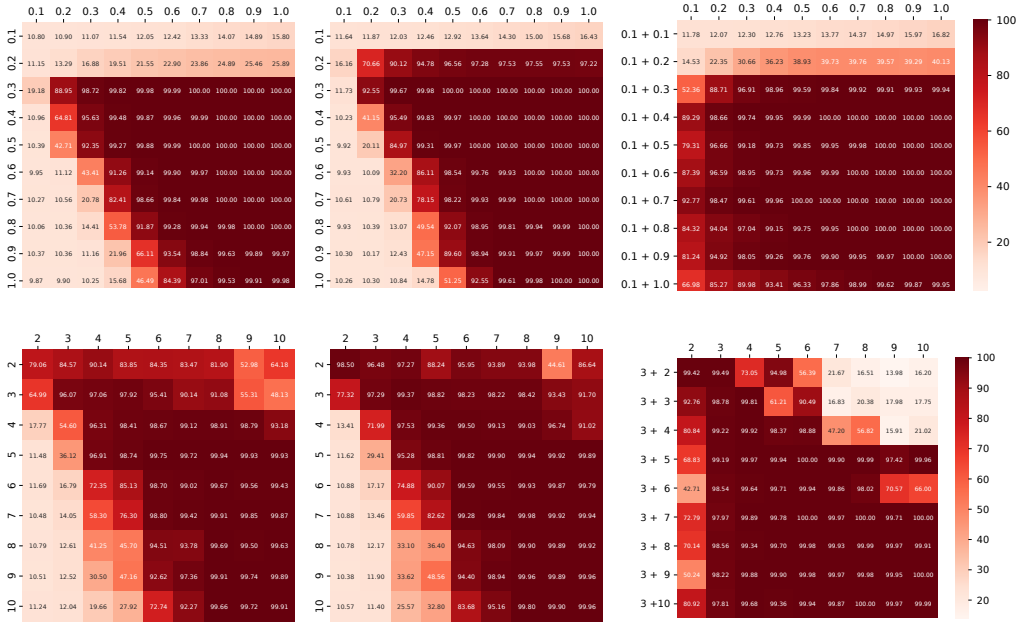
Figure 20: Results of BppAttack. The x- and y-axis indicate inference- and training-phase trigger intensity, respectively.

for its significantly larger size, confirming that our observed phenomena are model architecture agnostic. Moreover, the similar ASR distribution indicates that it is feasible for attackers to obtain desired training or inference intensities on surrogate models, enabling attacks in black-box scenarios.

C Detailed Results of Intensity Mixture

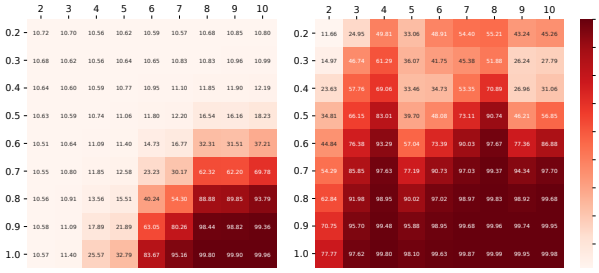
The detailed results of the intensity mixture are shown in Figure 21 (upper row). Triggers with intensity mixtures consistently outperform those utilizing a single intensity, except for a trigger intensity of 0.2 at a poisoning rate of 0.1, where the mixed triggers are too weak to execute the attack effectively. As shown in Figure 21 (lower row), We further confirmed that mixing other intensities can yield similar outcomes by mixing triggers of different sizes. Although the localized nature of BadNets triggers restricts the effect of low-intensity mixing (*e.g.*, mixing size 3 with 2 to 4) on larger triggers, the mixture of size 3 with higher intensities continues to demonstrate promising results.

Additionally, attackers can simultaneously mix varying intensities. For instance, we equally mix triggers with three configurations, resulting in a total poisoning rate of 10%: opacity=1.0&size=10, opacity=0.1&size=10, and opacity=1.0&size=3. Subsequently, we collect the backdoored model's ASR on triggers with varying opacities and sizes and then compare wit ith the ASR of another victim model trained with single intensity (opacity=1.0&size=10, with a poisoning rate at 10%, too). As shown in Figure 22, this mixing strategy enhances the generalizability across varying trigger intensities. This approach is similar to trigger-augmentation, which is utilized to enhance the robustness of backdoors in



(a) non-mix 5% poisoning rate (b) non-mix 10% poisoning rate (c) mix 10% poisoning rate

Figure 21: The detailed ASR of the BadNets with mixed training trigger intensities (upper row: opacity mixing, lower row: size mixing).



(a) w/o mixing (b) w/ mixing

Figure 22: Results of simultaneously mixing triggers of varying opacities and sizes.

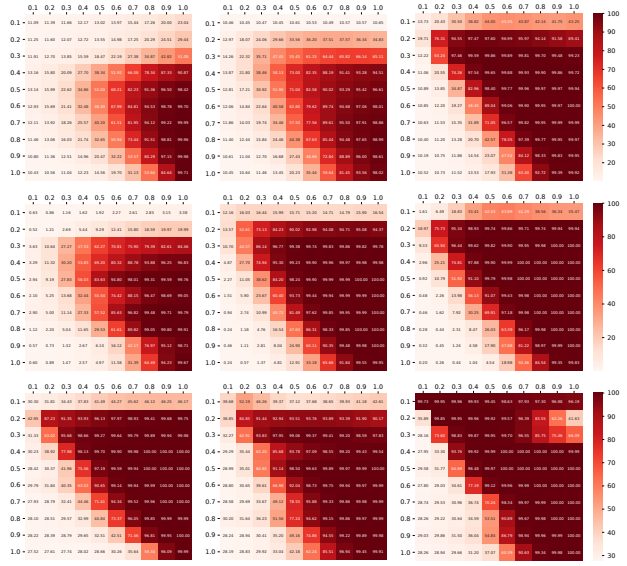
the physical world [52], *etc.*.

D Additional Overall Attack Results

Here we demonstrate the general applicability of our findings to two more attacks: BppAttack [89] and Styled [54].

In BppAttack, although the ASR is lower at low intensities (*i.e.*, large color depth) as we omit the contrastive learning that modifies the training process, the overall pattern of ASR in Figure 20 is consistent with our previous findings.

In Styled, we define an artificial intensity by the interpolation between the original images and their poisoned versions, as this attack lacks tunable parameters other than the image



(a) Gotham (b) Kelvin (c) Lomo

Figure 23: Results of Styled (the dataset for each row being CIFAR-10, GTSRB, and CelebA from top to bottom, respectively).

filter. As shown in Figure 23, the ASR pattern of this artificial intensity aligns well with our previous findings, demonstrating a practical approach to adjusting the intensity of triggers in backdoor attacks without requiring tunable parameters.