Cert-SSB: Toward Certified Sample-Specific Backdoor Defense

Ting Qiao, Yingjia Wang, Xing Liu, Sixing Wu, Jianbing Li, and Yiming Li

Fig. 1: An overview of existing randomized smoothing-based certified backdoor defenses and our Cert-SSB. The existing methods apply fixed noise to smooth classifiers for all inputs, ignoring sample diversity. This often leads to suboptimal certification performance. In contrast, Cert-SSB optimizes the noise, enabling the smoothing strategy to adapt to different inputs (as shown in the right figure), thereby achieving more robust certified backdoor defenses.

Abstract—Deep neural networks (DNNs) are vulnerable to backdoor attacks, where an attacker manipulates a small portion of the training data to implant hidden backdoors into the model. The compromised model behaves normally on clean samples but misclassifies backdoored samples into the attacker-specified target class, posing a significant threat to real-world DNN applications. Currently, several empirical defense methods have been proposed to mitigate backdoor attacks, but they are often bypassed by more advanced backdoor techniques. In contrast, certified defenses based on randomized smoothing have shown promise by adding random noise to training and testing samples to counteract backdoor attacks. In this paper, we reveal that existing randomized smoothing defenses implicitly assume that all samples are equidistant from the decision boundary. However, it may not hold in practice, leading to suboptimal certification performance. To address this issue, we propose a sample-specific certified backdoor defense method, termed Cert-SSB. Cert-SSB first employs stochastic gradient ascent to optimize the noise magnitude for each sample, ensuring a sample-specific noise level that is then applied to multiple poisoned training sets to retrain several smoothed models. After that, Cert-SSB aggregates the predictions of multiple smoothed models to generate the final robust prediction. In particular, in this case, existing certification methods become inapplicable since the optimized noise varies across different samples. To conquer this challenge, we introduce a storage-update-based certification method, which dynamically adjusts each sample's certification region to improve certification performance. We conduct extensive experiments on multiple benchmark datasets, demonstrating the effectiveness of our proposed method. Our code is available at https://github.com/NcepuQiaoTing/Cert-SSB.

Index Terms—Certified Backdoor Defense, Backdoor Defense, Randomized Smoothing, Trustworthy ML, AI Security

I. Introduction

RECENTLY, deep neural networks (DNNs) have been widely and successfully adopted in various domains, including mission-critical applications, such as face recognition [1], [2], [3]. However, training high-performance models typically requires large amounts of data and computational resources, which can be costly. Consequently, researchers

Ting Qiao, Yingjia Wang, Sixing Wu and Jianbing Li are with School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China (e-mail: qiaoting@ncepu.edu.cn, wyj@ncepu.edu.cn, wusx@ncepu.edu.cn, lijb87@ncepu.edu.cn).

Xing Liu is with Research Institute, China Unicom, Beijing 100048, China, and also with National Engineering Research Center of Next Generation Internet Broadband Service Application, Beijing 100037, China (e-mail: liux737@chinaunicom.cn).

Yiming Li is with College of Computing and Data Science, Nanyang Technology University, Singapore 639798 (e-mail: liyiming.tech@gmail.com).

Corresponding Author(s): Jianbing Li and Yiming Li.

often rely on third-party resources, such as publicly available datasets, cloud computing platforms, and pre-trained models, to reduce the training burden. Arguably, this reliance introduces security risks, with backdoor attacks [4], [5], [6], [7] being among the most severe threats. In a backdoor attack, adversaries inject predefined trigger patterns into a subset of the training data, causing the model to misclassify any input containing the trigger according to the attacker's intent. These attacks are both stealthy and highly detrimental, making them a key concern in both academia and industry. An industry report [8] highlights that backdoor attacks rank as the fourth most significant security threat faced by enterprises. Government agencies also recognize the severity of this issue. For instance, the U.S. intelligence community [9] has launched a dedicated funding program to counter backdoor attacks and related threats. To prevent models from becoming compromised due to backdoor attacks, developing effective defense mechanisms has become an urgent priority.

To mitigate backdoor threats, researchers have made significant efforts in both detection [10], [11], [12] and defense [13], [14], [15]. However, advanced backdoor attacks [16], [17], [18] can still easily bypass existing defenses, leading to an ongoing arms race between defenders and attackers. To address this issue, some studies have proposed certified backdoor defense methods, primarily categorized into deterministic certification [19], [20], [21], [22] and probabilistic certification [23], [24]. These methods aim to provide theoretical guarantees, ensuring that the classification results of testing samples remain consistent regardless of whether the model is trained on clean or backdoor data, as long as the

perturbation induced by the trigger remains within an ℓ_p norm ball of radius r. However, deterministic methods face scalability challenges when applied to large-scale neural networks. Consequently, probabilistic certification approaches based on randomized smoothing have emerged as a more practical alternative and have demonstrated robustness on large-scale datasets such as ImageNet [25]. Randomized smoothing was initially developed to certify robustness against adversarial examples. Its principle is to introduce random noise into the input data, ensuring that the classification results remain consistent within a specified region (*e.g.*, an ℓ_p norm neighborhood), thereby achieving robustness. Notably, pioneering studies [23], [24] showed that certified backdoor defenses based on random smoothing, which are robust against bounded backdoor patterns (*i.e.*, constrained pixel-level perturbation), can be achieved by introducing isotropic Gaussian noise into a tuple consisting of a testing instance and the training set to mitigate the impact of attacker-injected triggers, effectively neutralizing backdoor attacks during the training phase.

In this paper, we revisit existing randomized smoothing-based certified backdoor defenses. We find that these methods typically apply a fixed (*i.e.*, identical) magnitude of Gaussian noise to each sample to smooth the base classifier (*i.e.*, the decision boundary), thereby producing the final robust predictions. In other words, this approach (implicitly) assumes that all samples are equidistant from the decision boundary. However, inspired by [26], we recognize that this assumption may not hold in practice and could even degrade defense performance, as it may not be optimal for every sample. For example, as shown in the left part of Figure 1, adding an overly large noise magnitude to samples near the decision boundary can lead to misclassification, whereas increasing the noise magnitude for samples farther from the decision boundary can potentially enhance their certification performance. Based on this observation, we further analyze the intrinsic characteristics of samples, particularly their distances to the decision boundary. We find that these distances vary significantly among samples, and regardless of whether they belong to the training or testing set, their certification radius under a fixed noise magnitude is influenced by their individual properties. Therefore, an ideal strategy should be: applying smaller noise to samples near the decision boundary while assigning larger noise to those farther away, thereby better balancing classification performance and robustness, as illustrated in the right part of Figure 1. This finding raises a key question: *How can we exploit the intrinsic properties of samples to adjust the noise magnitude for each sample to design more effective certified backdoor defenses?*

Fortunately, the answer to the above question is affirmative. Arguably, the most direct approach is to optimize the noise at each sample by maximizing the confidence margin between the top-1 and top-2 predicted classes of the classifier (*i.e.*, the certification radius). However, due to the lack of an analytical expression for the certification radius, direct optimization is challenging. Inspired by [27], we employ a stochastic gradient ascent method to iteratively optimize the noise in order to maximize the certification radius. However, during the optimization process, the continuous adjustment of noise alters

the data distribution, leading to increased variance in gradient estimation and affecting optimization stability. To address this issue, we propose an advanced sample-specific certified backdoor defense method, termed Cert-SSB. In general, Cert-SSB consists of two main stages: training and inference. In the first stage, we train multiple smoothed models using the optimized noise, which is obtained through stochastic gradient ascent to maximize the certification radius. Generally, the certification radius is computed based on the predictions of classifiers trained with fixed noise. Besides, we adopt a reparameterization technique to reduce gradient variance and enhance optimization stability. In the inference stage, we aggregate multiple smoothed classifiers trained in the first stage to generate the final smoothed prediction. However, since the optimized noise results in different noise magnitudes for each sample, existing certification methods, which typically assume a fixed noise level, are no longer directly applicable. To resolve this issue, we propose a storage-update-based certification method, which dynamically adjusts the certification region (*i.e.*, the space covered by the certification radius) for each sample. This ensures that certification regions do not overlap between different samples and that predictions remain consistent within each certified region.

Our main contributions can be summarized as follows:

- We revisit existing randomized smoothing-based certified backdoor defenses and reveal that their use of fixed noise results in suboptimal certification performance for samples, affecting both training and testing samples.
- We propose a sample-specific certified backdoor defense method (*i.e.*, Cert-SSB) to dynamically adjust the smoothing noise magnitude for different samples to optimize certification performance.
- We introduce a storage-update-based certification method to dynamically update each sample's certification region, ensuring non-overlapping certified regions across different samples and improving certification robustness.
- We conduct extensive experiments on benchmark datasets to validate Cert-SSB's effectiveness, demonstrating its superior certification performance over existing methods.

The remainder of this paper is organized as follows. Section II provides a brief overview of related work. Section III revisits the limitations of existing certified backdoor defenses. Following that, in Section IV, we describe the threat model, defense goal, and an overview of our approach, followed by a detailed design of our certified backdoor defense method. Section V reports experimental results and performance analysis. Finally, we conclude this paper at the end.

II. RELATED WORKS

A. Backdoor Attacks

Backdoor attacks [28], [29], [30], [31] have emerged as a new threat in the training process of deep neural networks (DNNs). There are various ways to categorize backdoor attacks. First, they can be classified based on the adversarial objective into all-to-one and all-to-all. In all-to-one attacks, all samples with backdoor triggers are misclassified into a

predefined fixed target label, making the attack relatively simple and straightforward. In contrast, all-to-all attacks involve misclassifying samples with triggers into specific target classes based on their original categories, with a fixed mapping relationship between classes. This type of attack is more complex. Second, backdoor attacks can be categorized based on the threat scenario into three major types: **(1)** poison-only attacks [32], [16], **(2)** training-controlled attacks [33], [34], and **(3)** model-modified attacks [35], [36]. Specifically, poison-only attacks restrict the adversary to modifying the training dataset; training-controlled attacks allow the adversary to fully control the training process, including both the training data and algorithms. In contrast to these approaches, model-modified attacks mainly focus on the deployment phase rather than the training phase, embedding hidden backdoors by directly modifying model weights or introducing malicious DNN modules. In this paper, we mainly focus on poison-only backdoor attacks, which represent the most classical setting and pose the broadest threat scenarios. Recently, there are also a few works exploring how to exploit backdoor attacks for positive purposes [37], [38], [39], [40], [41], [42], which is out of the scope of this paper.

B. Backdoor Defense

In general, existing backdoor defense methods can be categorized into empirical defenses [43], [44], which rely on heuristic approaches to counter specific types of attacks, and certified defenses [22], [24], which provide theoretical guarantees for classifier robustness against adversarial perturbations.

1) Empirical defenses: Existing empirical defense methods can be classified into five main categories: **(1)** the detection of poisoned training samples [45], [46], **(2)** poison suppression [47], [48], **(3)** backdoor removal [49], [50], **(4)** the detection of poisoned testing samples [51], [52], and **(5)** the detection of attacked models [53], [54]. Specifically, the detection of poisoned training samples aims to identify and filter out malicious samples from the training set. Poison suppression prevents the model from learning poisoned samples by modifying the training process, thereby inhibiting the formation of hidden backdoors. Backdoor removal focuses on eliminating hidden backdoors from pre-trained (third-party) models. Detection of poisoned testing samples is designed to identify and block poisoned inputs during the testing phase. Lastly, the detection of attacked models determines whether a given model has been compromised by analyzing certain model properties. However, [55] and [56] revealed that new attack strategies could circumvent these empirical defenses, highlighting the ongoing arms race between attack and defense techniques.

2) Certified defenses: Existing certified defense methods can be categorized into deterministic defenses [19], [20], [21], [22], which provide a guaranteed 'certified' outcome when the input is robust to attacks, and probabilistic defenses [23], [24], which ensure a 'certified' result with a certain probability (*e.g.*, 99.9%), where the randomness is independent of the input sample. However, deterministic defenses often face scalability challenges when applied to large networks. In this work, we focus on probabilistic certified defenses.

Probabilistic certification offers better scalability. Previous methods primarily relied on intrinsic mechanisms [57], [58] or randomized smoothing techniques [23], [24] to achieve robust predictions. For example, Jia *et al.* [57] leveraged the intrinsic ensemble technique within the bagging mechanism, focusing on the number of poisoned samples. Similarly, Jia *et al.* [58] employed the intrinsic majority voting mechanism in the k-nearest neighbors algorithm, focusing on the number of neighbors. However, these methods do not consider the size of the trigger, making them unsuitable for defending against backdoor attacks. On the other hand, Wang *et al.* [23] were the first to apply randomized smoothing to defend against backdoor attacks, introducing random noise to neutralize potential triggers. However, this method lacked comprehensive experimental evaluation and failed to achieve a high robustness bound. Recently, Weber *et al.* [24] proposed the robustness against backdoor attacks (RAB) framework, which provides certified robustness against backdoor attacks and sets a theoretical benchmark for provable defenses in this field. The detailed implementation of RAB will be discussed in Section II-C. However, current methods implicitly assume that all samples are equidistant from the decision boundary, which may not hold in practice. This leads to suboptimal certification and highlights the urgent need for adaptive approaches that account for sample-specific characteristics.

C. Randomized smoothing and RAB

Randomized Smoothing (RS) [59] is a probabilistic defense method that enhances classifier robustness by smoothing predictions. Specifically, given an input $\boldsymbol{x}$, the smoothed classifier $g(\boldsymbol{x}, \sigma)$ selects the most probable class predicted by the base classifier f under isotropic Gaussian noise. Formally:

$$g(\boldsymbol{x}, \sigma) := \arg\max_{y \in \mathcal{Y}} \mathcal{P}_{\boldsymbol{\epsilon}}(f(\boldsymbol{x} + \boldsymbol{\epsilon}) = y), \quad (1)$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 I)$. The noise level σ is a hyperparameter that controls the trade-off between robustness; it does not change with the input $\boldsymbol{x}$. Using the Neyman-Pearson lemma [60], Cohen *et al.* [59] proved that $g(\boldsymbol{x}, \sigma)$ is certifiably robust to adversarial perturbations under the ℓ_2 norm constraint. Define $y_A = \arg\max_y \mathcal{P}_{\boldsymbol{\epsilon}}(f(\boldsymbol{x} + \boldsymbol{\epsilon}) = y)$, and assume that when classifying a perturbed input $\boldsymbol{x} + \boldsymbol{\epsilon}$, the base classifier f assigns the most probable class y_A with probability $P_A = \mathcal{P}_{\boldsymbol{\epsilon}}(f(\boldsymbol{x} + \boldsymbol{\epsilon}) = y_A)$, and the second most probable class y with probability $P_B = \max_{y \neq y_A} \mathcal{P}_{\boldsymbol{\epsilon}}(f(\boldsymbol{x} + \boldsymbol{\epsilon}) = y)$. Then, it is always true that $g(\boldsymbol{x} + \boldsymbol{\Delta}, \sigma) = y_A$ as long as $\|\boldsymbol{\Delta}\|_2 < r$, where the certified robust radius r is given by:

$$r(\boldsymbol{x}, \sigma) := \frac{\sigma}{2} \left(\Phi^{-1}(P_A(\boldsymbol{x}, \sigma)) - \Phi^{-1}(P_B(\boldsymbol{x}, \sigma)) \right), \quad (2)$$

where Φ^{-1} represents the inverse Gaussian cumulative distribution function (CDF).

In general, RS techniques are primarily designed to certify adversarial robustness by adding noise to testing instances. Most recently, a few pioneering research [23], [24] showed that we can achieve certified backdoor defenses that are robust against bounded backdoor patterns by introducing isotropic Gaussian noise to a tuple consisting of a testing instance and

Fig. 2: Distribution of ℓ_2 norm distances between samples and their closest boundary samples.

the training set to neutralize backdoor effects. Among these approaches, the most notable is RAB [24]. In the following, we briefly describe the implementation details of RAB.

Overview of RAB [24]. Given a dataset $\mathcal{D}$ and a testing instance $\boldsymbol{x}$, the base classifier f, learns a probability distribution $\mathcal{P}_\epsilon(\boldsymbol{q}(\boldsymbol{x}, \mathcal{D}) = y)$ over the class labels. It outputs the label that is most likely under this learned distribution $\boldsymbol{q}$: $f(\boldsymbol{x}, \mathcal{D}) = \arg\max_y \mathcal{P}_\epsilon(\boldsymbol{q}(\boldsymbol{x}, \mathcal{D}) = y)$. A smoothed classifier $g(\boldsymbol{x}, \mathcal{D}, \sigma)$ returns whichever class the base classifer $f(\boldsymbol{x}, \mathcal{D})$ is most likely to predict when $\boldsymbol{x}$ is perturbed by smoothing distributions $X = (Z, D)$:

$$g(\boldsymbol{x}, \mathcal{D}, \sigma) = \arg\max_y \mathcal{P}_{\epsilon(Z,D)}(f(\boldsymbol{x} + Z, \mathcal{D} + D) = y), \quad (3)$$

where $Z \sim \mathcal{N}(0, \sigma^2 I)$ is assumed to be independent, and $D \sim \mathcal{N}(0, \sigma^2 I)$ consists of n independent and identically distributed random variables $D^{(i)}$, each added to a training instance in $\mathcal{D}$. Let $\boldsymbol{\delta} = (\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_n)$ denote backdoor patterns applied to n training instances in $\mathcal{D}$, and let $\mathcal{B}_{\boldsymbol{x}}$ denote the backdoor trigger added to the testing instance $\boldsymbol{x}$. Define $y_A = \arg\max_y \mathcal{P}_{\epsilon(Z,D)}(f(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}} + Z, \mathcal{D} + \boldsymbol{\delta} + D) = y)$, assume that when classifying a point $\mathcal{N}(\boldsymbol{x}, \sigma^2 I)$, the base classifier $f(\boldsymbol{x}, \mathcal{D})$ assigns the most probable class y_A with probability $P_A(\boldsymbol{x}, \mathcal{D}, \sigma) = \mathcal{P}_{\epsilon(Z,D)}(f(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}} + Z, \mathcal{D} + \boldsymbol{\delta} + D) = y_A)$, and the "runner–up" class y with probability $P_B(\boldsymbol{x}, \mathcal{D}, \sigma) = \max_{y \neq y_A} \mathcal{P}_{\epsilon(Z,D)}(f(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}} + Z, \mathcal{D} + \boldsymbol{\delta} + D)) = y)$. Then, it is always true that $g(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}}, \mathcal{D}, \sigma) = g(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}}, \mathcal{D} + \boldsymbol{\delta}, \sigma) = y_A$ as long as the backdoor patterns $\sqrt{\sum_{i=1}^n \|\boldsymbol{\Delta}_i\|_2^2} \leq r$, where

$$r = \frac{\sigma}{2}\left(\Phi^{-1}(P_A(\boldsymbol{x}, \mathcal{D}, \sigma)) - \Phi^{-1}(P_B(\boldsymbol{x}, \mathcal{D}, \sigma))\right). \quad (4)$$

By analyzing Eq. (4), we find that increasing the hyperparameter σ enlarges the certified radius r, thereby enhancing the model's robustness. However, excessively increasing the noise magnitude may degrade classification accuracy (*i.e.*, incorrect predictions), which reflects the trade-off between robustness and accuracy. Therefore, a key challenge remains: how to determine the optimal noise level σ for each input.

III. REVISITING CERTIFIED BACKDOOR DEFENSES

Existing randomized smoothing-based certified backdoor defense methods *implicitly* assume that all samples are equidistant from the decision boundary, *i.e.*, they apply a fixed noise magnitude to each sample to smooth the classifier and obtain the final robust prediction. In this section, we analyze the variations in sample-to-decision-boundary distances from an intrinsic sample property perspective and further explore the limitations of using fixed Gaussian noise in existing methods.

A. Preliminaries

The Main Pipeline of (Poisoning-based) Backdoor Attacks. Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ represents the benign dataset consisting of n samples, where $\boldsymbol{x}_i \in \mathcal{X}$ is the i-th image, $y_i \in \mathcal{Y} = \{1, 2, \cdots, K\}$ is its corresponding label, and K denotes the total number of classes. In general, adversaries create a poisoned dataset $\mathcal{D}_p$ to train the target model using either a standard loss function or a customized one specified by the attacker. Specifically, $\mathcal{D}_p$ consists of two main parts: **1)** the modified version of a selected subset (*i.e.*, $\mathcal{D}_s$) of $\mathcal{D}$, and **2)** the remaining benign subset $\mathcal{D}_b$. Formally, $\mathcal{D}_p = \mathcal{D}_m(\mathcal{B}_{\boldsymbol{x}}, \hat{y}) \cup \mathcal{D}_b$, where $\mathcal{D}_m(\mathcal{B}_{\boldsymbol{x}}, \hat{y}) = \{\boldsymbol{x}_i + \boldsymbol{\Delta}_i, \hat{y}\}_{i=1}^d$, $\mathcal{D}_b = \mathcal{D} \backslash \mathcal{D}_s = \{\boldsymbol{x}_i, y_i\}_{i=d+1}^n$, $\mathcal{B}_{\boldsymbol{x}}$ introduces unique trigger patterns $\boldsymbol{\Delta}_i$ into the selected training instances, and $\hat{y} = G_Y(y)$. Here, $\lambda \triangleq \frac{|\mathcal{D}_m|}{|\mathcal{D}|}$ is the *poisoning rate*, and G_Y is adversary-specified poisoned label generator. For example, in Badnets [32], $G_Y(y) = y_t$ for all-to-one attacks, where $y_t \in \mathcal{Y}$ is the target label, and $G_Y(y) = y + 1 \mod K$ for all-to-all attacks. The attack succeeds if the classifier predicts the target label $\hat{y}$ with high probability for a testing example $\boldsymbol{x}$ modified with the backdoor pattern $\mathcal{B}_{\boldsymbol{x}}$: $f(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}}, \mathcal{D}_m(\mathcal{B}_{\boldsymbol{x}}, \hat{y})) = \hat{y}$.

Definition 1 (Boundary Samples and Closet Boundary Samples.). *Consider the logit margin of model $f : \mathcal{X} \to [0, 1]^K$ with respect to the label y, defined as: $\phi_y(\boldsymbol{x}; \boldsymbol{w}) = f_y(\boldsymbol{x}; w) - \max_{y' \neq y} f_{y'}(\boldsymbol{x}; \boldsymbol{w})$. A sample $\boldsymbol{x}$ is classified as y by the model $f(\cdot; \boldsymbol{w})$ if and only if $\phi_y(\boldsymbol{x}; \boldsymbol{w}) \geq 0$. The set of **boundary samples** belonging to class y can be expressed as $\mathcal{T}(y; \boldsymbol{w}) = \{\boldsymbol{x}^* : \phi_y(\boldsymbol{x}^*; \boldsymbol{w}) = 0\}$. Following the prior work [61], the **closest boundary sample** for $\boldsymbol{x}$ is defined as:*

$$\bar{\boldsymbol{x}}^* \triangleq \arg\min_{\bar{\boldsymbol{x}}} \|\boldsymbol{x}^* - \boldsymbol{x}\|_p, \quad s.t. \quad \phi_y(\boldsymbol{x}^*, \boldsymbol{w}) = 0, \quad (5)$$

where $\|\cdot\|_{1 \leq p \leq \infty}$ is the ℓ_p norm.

Generating the Closest Boundary Samples. To compute the closest boundary sample, we leverage the fast adaptive boundary attack (FAB) [62]. Specifically, we modify FAB to implement an iterative algorithm using gradient ascent with $\nabla_{\boldsymbol{x}} \phi_y(\boldsymbol{x}, \boldsymbol{w})$, updating the boundary sample at the $(t + 1)$-th iteration as follows:

$$\boldsymbol{x}_{t+1}^* = \beta_t \cdot \boldsymbol{x}_0 + (1 - \beta_t)\left\{\boldsymbol{x}_t^* + \alpha_t \frac{\nabla_{\boldsymbol{x}} \phi_y(\boldsymbol{x}_t^*; \boldsymbol{w})}{\|\nabla_{\boldsymbol{x}} \phi_y(\boldsymbol{x}_t^*; \boldsymbol{w})\|}\right\}, \quad (6)$$

where α_t is a positive step size, $\boldsymbol{x}_0$ is an initial point satisfying $\phi_y(\boldsymbol{x}_0; \boldsymbol{w}) \leq 0$ and $\beta_t \in [0, 1]$ is a line search parameter, to ensure $\boldsymbol{x}_{t+1}^*$ satisfies $\phi_y(\boldsymbol{x}_{t+1}^*; \boldsymbol{w}) = 0$. In practice, $\boldsymbol{x}_0$ is randomly selected from the validation set, ensuring its label differs from y.

B. Analysis of Sample's Distance to Decision Boundary

We hereby analyze how the distance from a sample to the decision boundary varies across inputs. Specifically, this distance is estimated using the 'closest boundary sample' defined in Definition 1 to avoid the inaccurate estimation using a random boundary one since there are multiple of them.

Setting. We hereby use BadNets [32] attack with a ResNet model [63] on the CIFAR-10 [64] datasets for discussion.

Fig. 3: Effect of different noise levels on the certified radius for MNIST and CIFAR-10 datasets. The first two subfigures show results for testing samples, while the last two show results for training samples.

Specifically, we set the target label y_t as '0' and the poisoning rate as 5%. Following the previous work [24], we use a one-pixel patch located at the lower right corner of the image as the trigger pattern. We randomly select 2,000 poisoned samples and use Eq. (6) to generate their closest boundary samples for the target label y_t. During training, we compute the ℓ_2 norm between each sample and its closest decision boundary sample. Samples with a small distance to the decision boundary are referred to as *easy samples*, while those with a larger distance are referred to as *hard samples*.

Result. As shown in Figure 2, the ℓ_2 distances to the closest boundary samples vary significantly among different samples in the poisoned dataset. Specifically, although most samples have relatively small distances (*e.g.*, $\ell_2 \leq 0.3$), a considerable number of hard samples exhibit larger distances to their closest boundary samples. Therefore, these hard samples require a larger magnitude of noise to effectively defend against backdoor attacks during training. In contrast, for easy samples with smaller distances, only a smaller magnitude of noise is needed to achieve the desired defense effect. For samples that fall between hard samples and easy samples, *a trade-off and adjustment in noise selection are necessary*.

C. Limitations of Fixed Noise in Testing Samples

Setting. We hereby randomly select three testing samples from the MNIST and CIFAR-10 datasets, respectively, and evaluate the certification radius of the RAB model [24] trained with $\sigma = 1.0$. The certification radius is computed following Eq. (4), using different noise levels with σ values ranging from 0 to 1.0 in increments of 0.2. All other experimental settings remain as described in Section III-B.

Result. As shown in Figures 3(a) and 3(b), the three samples from both MNIST and CIFAR-10 datasets exhibit a trend where the certification radius first increases and then decreases as the noise magnitude increases. Notably, although the model was trained with $\sigma = 1.0$, the optimal certification radius does not occur at this noise level. Taking the MNIST dataset as an example, sample 1 reaches its maximum certification radius of approximately 1.5 at $\sigma = 0.8$, sample 2 peaks at about 1.3 when σ approaches 0.9, while sample 3 maintains a relatively stable low value. This result suggests that the optimal certification performance is not necessarily achieved by using the same noise magnitude during testing as in training. Therefore, *the σ value should be optimized for each sample* to achieve the maximum certification radius.

D. Limitations of Fixed Noise in Training Samples

Setting. We randomly select three training samples from the MNIST and CIFAR-10 datasets, respectively, and train multiple models with different noise levels. Specifically, we apply noise with standard deviations of $\sigma \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ during training. During the testing phase, we evaluate each model using the same noise level as in its training phase. That is, for a model trained with $\sigma = 0.5$, its certification radius is also computed using $\sigma = 0.5$. All other experimental settings remain as described in Section III-B.

Result. As shown in Figures 3(c) and 3(d), the certification radii of different samples exhibit distinct trends as the noise level varies. Overall, while some samples achieve larger certification radii at appropriate noise levels, others are more sensitive to noise, showing instability or even misclassification at higher noise values. For example, Sample 1 shows a continuously increasing certification radius as the noise σ increases, indicating that its robustness remains stable even at higher noise levels. Sample 2 exhibits a stable certification radius in the range of $\sigma = 0.6$ to 0.8, without significant changes as the noise level further increases. This suggests that this noise level may be optimal for this sample. In this case, increasing the noise further may not improve the certification radius and could even negatively impact classification accuracy. Therefore, for this sample, a trade-off must be made between accuracy and robustness. In contrast, Sample 3 experiences a gradual decrease in certification radius as the noise level increases and eventually undergoes misclassification at higher noise levels. The "×" markers in the figure indicate misclassified points. This result suggests that *the noise level used for training should be optimized based on the characteristics of individual samples* rather than using a fixed value to achieve better certification performance.

IV. METHODOLOGY

A. Threat Model and the Goal of Certified Defense

1) Threat Model: This work focuses on defending against poison-only backdoor attacks. Adversaries can manipulate the training data but cannot modify other training components, such as the loss function or model architecture. Defenders have full control over the training process but cannot detect whether the data is poisoned, nor do they know the trigger pattern.

2) Goal of Certified Defense: The primary goal is to defend against poison-only backdoor attacks by obtaining a robustness

Fig. 4: The main pipeline of our Cert-SSB consists of two stages. In the first stage, we adopt a stochastic gradient ascent (SGA) strategy to iteratively optimize the noise in order to maximize the certification radius r, thereby solving for the optimal noise (*i.e.*, sample-specific noise). The value of r is computed based on the predictions of a base model trained with fixed noise. This optimized noise is then injected into the poisoned training set and used to train M smoothed models. In the second stage, the M smoothed models trained in the first stage are aggregated to generate the final smoothed prediction. Notably, under this setting, the traditional certification method, which typically assumes a fixed noise level, is no longer applicable. To conquer this challenge, we propose a novel storage-update-based certification method, which ensures that each certification region is non-overlapping and maintains consistent predictions within each region (see Figure 5 for more details).

threshold r through analysis, ensuring that if the total backdoor modification satisfies $\sqrt{\sum_{i=1}^{d} \|\Delta_i\|_2^2} < r$, the classifier's predictions on testing samples containing backdoor triggers remain unaffected by whether the model was trained on poisoned or clean data. In other words, the model's predictions should be consistent, expressed as: $f(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}}, \mathcal{D}_m(\mathcal{B}_{\boldsymbol{x}}, \hat{y})) = f(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}}, \mathcal{D}_m(\emptyset))$, where $\mathcal{D}_m(\emptyset)$ denotes the clean dataset without any embedded backdoor patterns (*i.e.*, $\Delta_i = 0$).

B. Overview of the Proposed Method

As demonstrated in Section III, existing randomized smoothing-based certified backdoor defenses exhibit suboptimal certification performance, regardless of whether fixed noise is applied to training or testing samples. This is because each sample has a different distance to the decision boundary. To address this issue, we propose a sample-specific certified backdoor defense method, dubbed Cert-SSB, in which the noise level is adaptively adjusted for each individual sample.

As shown in Figure 4, our method consists of two main stages: **(1)** Cert-SSB training stage, and **(2)** Cert-SSB inference stage. In the training stage, we apply stochastic gradient ascent to iteratively solve for the optimal noise level $\sigma_{\boldsymbol{x}}^*$ that maximizes the certification radius. The radius is computed based on the prediction probabilities of a base classifier trained with fixed noise. Once obtained, the sample-specific noise $\sigma_{\boldsymbol{x}}^*$ is injected into the poisoned training set to train M smoothed models. In the inference stage, we aggregate predictions from the M smoothed models trained in the first stage to produce the final smoothed output. Intuitively, as long as the predicted probability of the most likely class exceeds that of the second most likely class, the model can be considered certifiably robust. However, under this sample-specific noise setting, traditional certification methods become inapplicable, as they typically assume a uniform noise level across all inputs. To overcome this limitation, we introduce a storage-update-based certification method, which categorizes certification regions

(*i.e.*, regions defined by the certified radius of each input) to ensure that they remain non-overlapping across inputs and that prediction consistency is maintained within each region (see Figure 5). The technical details are as follows.

C. Cert-SSB Training: Train the Model with Optimized Noise

In this stage, we train multiple smoothed models based on optimized noise, where the optimized noise is obtained by applying stochastic gradient ascent to maximize the certification radius. In general, the certification radius is computed based on the predictions of a classifier trained with fixed noise.

1) Optimized Sample-Specific Noise Generation: Given a smoothed classifier with fixed noise σ_0 (which reduces to the base classifier $f(\boldsymbol{x}, \mathcal{D})$ when $\sigma_0 = 0$), our goal is to construct a new smoothed classifier $g(\boldsymbol{x}, \mathcal{D}, \sigma_{\boldsymbol{x}}^*)$ based on an optimized noise $\sigma_{\boldsymbol{x}}^*$ for each sample. The new classifier should ensure that for all samples $\boldsymbol{x}$, the predictions of the two smoothed classifiers (with σ_0 and $\sigma_{\boldsymbol{x}}^*$) remain identical while also maximizing the certification radius at each sample $\boldsymbol{x}$. Formally, let y_A represent the predicted label under the fixed noise level σ_0, it can be defined as:

$$y_A = \arg \max_y \mathcal{P}_{\boldsymbol{\epsilon}(Z,D)}[f^y(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}} + Z, \mathcal{D} + \boldsymbol{\delta} + D)], \quad (7)$$

where $Z \sim \mathcal{N}(0, \sigma_0^2 I)$ is assumed to be independent, and $D \sim \mathcal{N}(0, \sigma_0^2 I)$ consists of n *i.i.d.* random variables $D^{(i)}$, each added to a training instance in $\mathcal{D}$. For each input $\boldsymbol{x}$, the optimized noise $\sigma_{\boldsymbol{x}}^*$ is obtained by solving for σ that maximizes the certification radius $r(\boldsymbol{x}, \sigma)$ in Eq. (4):

$$\sigma_{\boldsymbol{x}}^* = \arg \max_\sigma \frac{\sigma}{2}(\Phi^{-1}(P_A(\boldsymbol{x}, \mathcal{D}, \sigma)) - \Phi^{-1}(P_B(\boldsymbol{x}, \mathcal{D}, \sigma))), \tag{8}$$

where $P_A(\boldsymbol{x}, \mathcal{D}, \sigma) = \mathcal{P}_{\boldsymbol{\epsilon}(Z,D)}[f^{y_A}(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}} + Z, \mathcal{D} + \boldsymbol{\delta} + D)]$, $P_B(\boldsymbol{x}, \mathcal{D}, \sigma) = \max_{y \neq y_A} \mathcal{P}_{\boldsymbol{\epsilon}(Z,D)}[f^y(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}} + Z, \mathcal{D} + \boldsymbol{\delta} + D)]$.

In practice, we solve Eq. (8) using stochastic gradient ascent, where the probabilities of predicting class y_A and y are estimated via Monte Carlo approximation. Specifically, we introduce noise multiple times, record the output count for

these two classes, and approximate the probability distribution using their relative frequencies. Formally, the gradient of the objective at the t-th iteration is approximated as follows:

$$\nabla_{\sigma^t}\{\tfrac{\sigma^t}{2}\cdot[\Phi^{-1}(\tfrac{1}{J}\textstyle\sum_{j=1}^{J}f^{y_A}(\boldsymbol{x}+\mathcal{B}_{\boldsymbol{x}}+Z_i,\mathcal{D}+\boldsymbol{\delta}+D_i))$$
$$-\Phi^{-1}(\max_{y\neq y_A}\tfrac{1}{J}\textstyle\sum_{j=1}^{J}f^{y}(\boldsymbol{x}+\mathcal{B}_{\boldsymbol{x}}+Z_i,\mathcal{D}+\boldsymbol{\delta}+D_i))]\}, \tag{9}$$

where $Z_1,\ldots,Z_J \sim \mathcal{N}(0,(\sigma^t)^2 I)$ as well as $D_1,\ldots,D_J \sim \mathcal{N}(0,(\sigma^t)^2 I)$ are independently sampled at each iteration.

However, since the probabilities depend on the optimization variable σ, and σ parameterizes the smoothed distribution $\mathcal{N}(0,\sigma^2 I)$ [65], any change in σ affects the underlying distribution, which can result in high variance in the gradient estimation method. To address this problem, we adopt the reparameterization technique proposed by Kingma *et al.* [66] and Rezende *et al.* [67], which allows for a lower-variance gradient estimation of the objective in Eq. (9). Specifically, we reparameterize the noise as $Z = \sigma\hat{Z}$ and $D = \sigma\hat{D}$, where $\hat{Z}$ and $\hat{D}$ are sampled from a standard normal distribution, *i.e.*, $\hat{Z},\hat{D} \sim \mathcal{N}(0,I)$. This transformation allows us to reformulate the objective in Eq. (8) as follows:

$$\sigma_{\boldsymbol{x}}^* = \arg\max_{\sigma}\frac{\sigma}{2}(\Phi^{-1}(\hat{P}_A(\boldsymbol{x},\mathcal{D},\sigma)) - \Phi^{-1}(\hat{P}_B(\boldsymbol{x},\mathcal{D},\sigma))), \tag{10}$$

where $\hat{P}_A(\boldsymbol{x},\mathcal{D},\sigma) = \mathcal{P}_{\epsilon(\hat{Z},\hat{D})}[f^{y_A}(\boldsymbol{x}+\mathcal{B}_{\boldsymbol{x}}+\sigma\hat{Z},\mathcal{D}+\boldsymbol{\delta}+\sigma\hat{D})]$, $\hat{P}_B(\boldsymbol{x},\mathcal{D},\sigma) = \max_{y\neq y_A}\mathcal{P}_{\epsilon(\hat{Z},\hat{D})}[f^{y}(\boldsymbol{x}+\mathcal{B}_{\boldsymbol{x}}+\sigma\hat{Z},\mathcal{D}+\boldsymbol{\delta}+\sigma\hat{D})]$. Note that under this reparameterization, the distributions $\hat{Z}$ and $\hat{D}$ are no longer dependent on the optimization variable σ. As a result, Eq. (10) typically yields lower-variance gradient estimates compared to the original formulation in Eq. (8).

2) Robust Training Process: Once the optimized noise $\sigma_{\boldsymbol{x}}^*$ is obtained, we incorporate it into the training process to enhance robustness. Specifically, we first sample M sets of noise vectors $b_1,\cdots,b_M$ from the distribution $D \sim \prod_{i=1}^{n}\mathcal{N}(0,I)$, where each set contains $n = |\mathcal{D}|$ *i.i.d.* vectors corresponding to the size of the training dataset. For each sampled noise set b_m, we construct a perturbed (poisoned) training dataset $\mathcal{D}_p + \sigma_{\boldsymbol{x}}^* b_m$ by introducing the sample-specific noise $\sigma_{\boldsymbol{x}}^*$. Next, we train M smoothed models on these perturbed datasets, denoted as $g_1(\boldsymbol{x},\mathcal{D},\sigma_{\boldsymbol{x}}^*),\ldots,g_M(\boldsymbol{x},\mathcal{D},\sigma_{\boldsymbol{x}}^*)$. To maintain consistency between the noise distributions during training and inference, we further sample a noise vector $\mu_M \sim \mathcal{N}(0,\sigma_0^2 I_h)$ for each model g_m, using the hash value of its trained model file as the random seed. This noise vector is stored alongside the model parameters and added to the input during inference. By introducing noise in both the training and testing phases, we ensure that the ensemble of smoothed models $\{g_1,\ldots,g_M\}$ avoids performance degradation when classifying clean inputs. See Algorithm 1 in our Appendix A for training details.

D. Cert-SSB Inference: Storage-update-based Certification

In this stage, we aggregate the ensemble of smoothed classifiers trained in the first stage to form the final prediction. However, under the use of sample-specific noise $\sigma_{\boldsymbol{x}}^*$, traditional certification methods are no longer applicable. To address this, we propose a novel storage-update-based certification method. By introducing a 'storage' mechanism, this method dynamically adjusts certification regions to ensure they are non-overlapping across inputs while preserving prediction consistency for each individual sample.

Formally, given a trained model $g(\boldsymbol{x},\mathcal{D},\sigma_{\boldsymbol{x}}^*)$ and a testing input $\boldsymbol{x}$, the prediction is computed via majority voting with the optimized noise. Specifically, we estimate the class probability as $\mathcal{P}_{\epsilon}(g(\boldsymbol{x},\mathcal{D},\sigma_{\boldsymbol{x}}^*) = y) = \frac{1}{M}\sum_{m=1}^{M}\mathbb{I}\{g_m(\boldsymbol{x}+\mu_m,\mathcal{D}_p+\sigma_{\boldsymbol{x}}^* b_m) = y\}$, where μ_m is the deterministic noise vector sampled during training, and $\mathbb{I}\{\cdot\}$ is the indicator function. The following theorem states the robustness guarantee:

Theorem 1 (Certified Robustness of Cert-SSB). *Let $\mathcal{B}_{\boldsymbol{x}} \in R^d$ and let $\boldsymbol{\delta} := (\boldsymbol{\Delta}_1,\boldsymbol{\Delta}_2,...,\boldsymbol{\Delta}_n)$ for backdoor patterns $\boldsymbol{\Delta}_i \in R^d$, and let $\mathcal{D}$ be a training set, and let smoothing noise $\hat{Z} \sim \mathcal{N}(0,I)$, $\hat{D} \sim \mathcal{N}(0,I)$. Let $y_A \in \mathcal{Y}$, such as $y_A = g(\boldsymbol{x}+\mathcal{B}_{\boldsymbol{x}},\mathcal{D}+\boldsymbol{\delta})$ with class probabilities satisfying $\mathcal{P}_{\epsilon(\hat{Z},\hat{D})}(f(\boldsymbol{x}+\mathcal{B}_{\boldsymbol{x}}+\sigma_{\boldsymbol{x}}^*\hat{Z},\mathcal{D}+\boldsymbol{\delta}+\sigma_{\boldsymbol{x}}^*\hat{D}) = y_A) \geq P_A \geq P_B \geq \max_{y\neq y_A}\mathcal{P}_{\epsilon(\hat{Z},\hat{D})}(f(\boldsymbol{x}+\mathcal{B}_{\boldsymbol{x}}+\sigma_{\boldsymbol{x}}^*\hat{Z},\mathcal{D}+\boldsymbol{\delta}+\sigma_{\boldsymbol{x}}^*\hat{D}) = y)$. Then, we have $g(\boldsymbol{x}+\mathcal{B}_{\boldsymbol{x}},\mathcal{D}) = g(\boldsymbol{x}+\mathcal{B}_{\boldsymbol{x}},\mathcal{D}+\boldsymbol{\delta}) = y_A$ for all backdoor patterns $\boldsymbol{\Delta}$ satisfying $\sqrt{\sum_{i=1}^{n}\|\boldsymbol{\Delta}_i\|_2^2} \leq r(g;\sigma_{\boldsymbol{x}_i}^*)$, where the certified robust radius r is given by*

$$r(g;\sigma_{\boldsymbol{x}_i}^*) = \frac{\sqrt{\sum_{i=1}^{n}(\sigma_{\boldsymbol{x}_i}^*)^2}}{2}\left(\Phi^{-1}(P_A(\sigma_{\boldsymbol{x}_i}^*)) - \Phi^{-1}(P_B(\sigma_{\boldsymbol{x}_i}^*))\right). \tag{11}$$

Compared to RAB [24], our method achieves a better trade-off between robustness and accuracy by replacing the fixed smoothing noise σ with optimized sample-specific noise $\sigma_{\boldsymbol{x}_i}^*$. This advantage is further supported by experimental results presented later. The formal proof is provided in Appendix B.

Notably, traditional certification methods become inapplicable in our setting due to two core limitations: **(1)** they assume non-overlapping certification regions for all inputs, and **(2)** they require a globally fixed robustness parameter (*e.g.*, noise level σ). To overcome these limitations, we propose a storage-update-based certification strategy. Before introducing the proposed method, we first provide a brief definition of the concepts of 'overlapping' and 'non-overlapping' certification regions. Based on this definition, we introduce Definition 3, which classifies the possible types of certification region overlapping. Then, following this classification, we describe the storage-update mechanism in detail (see Proposition 1).

Definition 2 (Overlapping and Non-overlapping of Certification Regions). *Let g be a sample-specific smoothed classifier, and let $r(\sigma_{\boldsymbol{x}_1}^*)$ denote the certification radius of g at input $\boldsymbol{x}_1$. For any other input $\boldsymbol{x}_2$, if $\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2 \leq r(\sigma_{\boldsymbol{x}_1}^*)$, the certification regions of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are said to overlapping; otherwise, they are said to be non-overlapping.*

Definition 3 (Classification Criteria of Certification Regions). *Let the triplet storage set $\mathcal{S} = \{(\boldsymbol{x}_i,\mathcal{Y}_i,\mathcal{R}_i)\}_{i=1}^{n}$ store all previously predicted inputs $\boldsymbol{x}_i$, their corresponding predictions $\mathcal{Y}_i$, and their associated certification regions $\mathcal{R}_i$. Here, $\mathcal{R}_i$ denotes the certification region centered at $\boldsymbol{x}_i$, characterized by the certification radius r_i. The certification regions $\mathcal{R}_i$ for different inputs are classified as follows (see Figure 5):*

- *Case 1: Non-overlapping Certification Regions. All certification regions are non-overlapping, i.e., $\forall i \neq j, \mathcal{R}_i \cap$*

Fig. 5: Storage-update-based certification (see proposition 1). Given two inputs x_1 and x_2 with certified regions $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively, there are three cases based on Definition 3.

$\mathcal{R}_j = \emptyset$, and the corresponding predictions are different, i.e., $\mathcal{Y}_i \neq \mathcal{Y}_j$.

- **Case 2: Overlapping Certification Regions with Consistent Predictions.** The certification region $\mathcal{R}_{n+1}$ of a new input x_{n+1} overlaps with an existing certification region $\mathcal{R}_i$, and their predictions are consistent, i.e., $\exists i$ such that $\mathcal{R}_i \cap \mathcal{R}_{n+1} \neq \emptyset$ and $\mathcal{Y}_{n+1} = \mathcal{Y}_i$.

- **Case 3: Overlapping Certification Regions with Inconsistent Predictions.** The certification region $\mathcal{R}_{n+1}$ of a new input x_{n+1} overlaps with an existing region $\mathcal{R}_i$, but their predictions differ, i.e., $\mathcal{Y}_{n+1} \neq \mathcal{Y}_i$. This case can be further divided into two subcases:
  - *The new input lies* inside *the existing certification region, i.e., $x_{n+1} \in \mathcal{R}_i$ and $\mathcal{R}_{n+1} \cap \mathcal{R}_i \neq \emptyset$.*
  - *The new input lies* outside *the existing certification region, i.e., $x_{n+1} \notin \mathcal{R}_i$ but $\mathcal{R}_{n+1} \cap \mathcal{R}_i \neq \emptyset$.*

Based on the classification in Definition 3, we propose a storage-update-based certification method, that enforces non-overlapping certification regions while maintaining prediction consistency (i.e., $\forall i \neq j, \mathcal{R}_i \cap \mathcal{R}_j = \emptyset, \mathcal{Y}_i \neq \mathcal{Y}_j$). In this way, the certification regions of inputs with different predicted labels do not overlapping within the storage set $\mathcal{S}$. This is a key property of a sound certification process.

Now, we introduce Proposition 1, which formalizes the proposed storage-update-based certification method.

Proposition 1 (Storage-update-based Certification). *Based on Definition 3, the storage-update-based certification method handles new inputs according to the following cases:*

- **Case 1:** *If $\forall i \neq j$, $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$ and $\mathcal{Y}_i \neq \mathcal{Y}_j$, then all existing triplets $(x_i, \mathcal{Y}_i, \mathcal{R}_i)$ and $(x_j, \mathcal{Y}_j, \mathcal{R}_j)$ in storage remain unchanged.*
- **Case 2:** *If there exists some i such that $\mathcal{R}_{n+1} \cap \mathcal{R}_i \neq \emptyset$ and $\mathcal{Y}_{n+1} = \mathcal{Y}_i$, then the new certification triplet $(x_{n+1}, \mathcal{Y}_{n+1}, \mathcal{R}_{n+1})$ can be directly added to the storage.*
- **Case 3:** *If $\mathcal{R}_{n+1} \cap \mathcal{R}_i \neq \emptyset$ and $\mathcal{Y}_{n+1} \neq \mathcal{Y}_i$, the method*

proceeds as follows[1] *(see Figure 5):*

- *If $x_{n+1} \in \mathcal{R}_i$: The new certification region is updated to the largest subset $\tilde{\mathcal{R}}_{n+1}$ such that $\tilde{\mathcal{R}}_{n+1} \subseteq \mathcal{R}_{n+1}$ and $\tilde{\mathcal{R}}_{n+1} \subseteq \mathcal{R}_i$. Then, $\mathcal{R}_{n+1}$ is replaced by $\tilde{\mathcal{R}}_{n+1}$, and the label $\mathcal{Y}_{n+1}$ is updated to $\mathcal{Y}_i$ to ensure prediction consistency.*
- *If $x_{n+1} \notin \mathcal{R}_i$: The new certification region is updated to the largest subset $\mathcal{R}'_{n+1}$ such that $\mathcal{R}'_{n+1} \subseteq \mathcal{R}_{n+1}$ and $\mathcal{R}'_{n+1} \cap \mathcal{R}_i = \emptyset$. Then, original $\mathcal{R}_{n+1}$ is replaced by $\mathcal{R}'_{n+1}$.*

After applying the appropriate case, the final triplet $(x_{n+1}, \mathcal{Y}_{n+1}, \mathcal{R}_{n+1})$ (or its updated form) is added to the storage set $\mathcal{S}$ for use in future certification.

In practice, in our experiments, we did not observe any cases where inputs with different predictions have overlapping certified regions. That is, for each input, the certified region stored in $\mathcal{S}$ is essentially determined by the certification radius computed using Eq. (11) for the sample-specific smoothed classifier $g(x, \mathcal{D}, \sigma_x^*)$. This can be attributed to two main reasons: **1)** Due to the high dimensionality of image datasets, the ℓ_2-norm distance between samples is significantly larger than the certification radius provided by randomized smoothing; **2)** The optimized noise σ_x^* tends to have a moderate value ($\sigma_x^* \leq 1.0$), resulting in relatively small certification regions. For example, the certification region corresponds to an ℓ_2-ball with a radius of approximately $4\sigma_x^*$, which is much smaller than the distances between samples in high-dimensional datasets (*e.g.*, ImageNet). Nonetheless, in certain rare or specifically constructed scenarios, it remains theoretically possible for inputs with different predictions to exhibit overlapping certified regions, particularly in low-dimensional input spaces or when abnormally large values of σ_x^* lead to significantly enlarged certification radii. For such cases, our storage-update-based certification strategy becomes indispensable: by appropriately adjusting the certified regions and associated predictions, it effectively resolves potential conflicts, as described in Proposition 1.

V. EXPERIMENTS

A. Main Settings

1) Datasets and Models: We conduct experiments on MNIST [68], CIFAR-10 [64], and ImageNette [69], using a simple CNN model [32], a lightweight ResNet-like model [59], and standard ResNet-18 model [70], respectively.

2) Training Settings: We adopt a sample-specific smoothing approach during training. In this stage, we set the number of sampled Gaussian noise vectors (*i.e.*, augmented datasets) to $M = 1,000$ for MNIST and CIFAR-10, and $M = 200$ for ImageNette, resulting in ensembles of 1,000 and 200 models, respectively. Following previous works [59], [27], the added noise follows a Gaussian distribution with mean $\mu = 0$ and standard deviation σ, set as follows: for MNIST and CIFAR-10, $\sigma \in \{0.12, 0.25, 0.5, 1.0\}$; for ImageNette, $\sigma \in \{0.25, 0.5, 1.0\}$. Additionally, we set the number of stochastic gradient ascent iterations to $T = 100$, the number

---

[1]This process is straightforward when the certification regions are ℓ_2-balls.

TABLE I: Certified performance (*i.e.*, ERA and AER) is reported using the best results across different noise levels at each radius under the all-to-one setting. We evaluate three types of attacks (one-pixel, four-pixel, and blending) on the MNIST, CIFAR-10, and ImageNette datasets. In particular, we mark the best certification results in boldface.

Dataset↓	Attack Setting↓, Metric→	Method↓	AER	Radius (ERA↑)							
				0	0.25	0.5	0.75	1.0	1.25	1.5	1.75
MNIST	One-pixel	RAB	1.48	**100**	99.91	99.76	99.43	99.05	97.73	55.79	0
		Cert-SSB	**1.65**	99.95	**99.91**	**99.81**	**99.62**	**99.34**	**98.82**	**86.53**	**42.98**
	Four-pixel	RAB	1.49	99.95	99.86	99.72	99.39	99.01	97.78	56.12	0
		Cert-SSB	**1.69**	**99.95**	**99.86**	**99.72**	**99.57**	**99.20**	**98.63**	**81.94**	**42.98**
	Blending	RAB	1.46	**100**	99.86	99.67	99.39	99.05	97.35	42.03	0
		Cert-SSB	**1.70**	99.95	**99.86**	**99.76**	**99.72**	**99.20**	**98.72**	**72.15**	**42.84**
CIFAR-10	One-pixel	RAB	0.55	**87.80**	69.70	56.70	38.30	16.55	2.60	0	0
		Cert-SSB	**0.62**	86.55	**71.90**	**60.75**	**46.30**	**26.10**	**11.50**	**1.45**	0
	Four-pixel	RAB	0.56	**88.70**	69.50	55.70	36.60	14.15	**2.25**	**0.05**	0
		Cert-SSB	**0.65**	86.40	**70.30**	**59.50**	**43.55**	**20.90**	1.60	0	0
	Blending	RAB	0.56	**88.00**	69.80	56.25	36.95	15.00	**2.35**	0	0
		Cert-SSB	**0.64**	86.15	**73.40**	**61.55**	**46.55**	**27.25**	0.05	0	0
ImageNette	One-pixel	RAB	0.49	94.62	74.18	52.60	35.42	14.60	0	0	0
		Cert-SSB	**0.64**	**95.20**	**86.36**	**72.50**	**45.08**	**32.10**	**17.36**	**5.08**	0
	Four-pixel	RAB	0.48	94.80	73.94	52.26	33.36	13.26	0	0	0
		Cert-SSB	**0.67**	**94.90**	**86.82**	**77.00**	**55.22**	**34.52**	**20.22**	**5.76**	0
	Blending	RAB	0.47	94.78	74.32	51.44	33.02	12.62	0	0	0
		Cert-SSB	**0.64**	**94.94**	**83.46**	**58.66**	**46.30**	**34.52**	**20.22**	**5.76**	0

TABLE II: Certified performance (*i.e.*, CRA and ACR) is reported using the best results across different noise levels at each radius under the all-to-one setting. We evaluate three types of attacks (one-pixel, four-pixel, and blending) on the MNIST, CIFAR-10, and ImageNette datasets. In particular, we mark the best certification results in boldface.

Dataset↓	Attack Setting↓, Metric→	Method↓	ACR	Radius (CRA↑)							
				0	0.25	0.5	0.75	1.0	1.25	1.5	1.75
MNIST	One-pixel	RAB	0.69	**46.37**	46.24	46.10	45.01	45.91	45.49	42.51	0
		Cert-SSB	**0.84**	46.29	**46.24**	**46.24**	**46.10**	**45.96**	**45.91**	**45.20**	**42.88**
	Four-pixel	RAB	0.68	**46.34**	46.24	46.10	**46.72**	45.91	45.63	41.23	0
		Cert-SSB	**0.87**	46.29	**46.24**	**46.24**	46.10	**46.01**	**45.91**	**45.67**	**43.88**
	Blending	RAB	0.69	**46.34**	46.24	46.10	45.01	45.91	45.49	42.46	0
		Cert-SSB	**0.87**	**46.34**	**46.29**	**46.24**	**46.19**	**45.96**	**45.91**	**45.63**	**44.30**
CIFAR-10	One-pixel	RAB	0.32	48.30	39.40	30.40	20.05	**8.35**	0.55	0	0
		Cert-SSB	**0.33**	**52.65**	**41.60**	**34.65**	**21.30**	3.50	0	0	0
	Four-pixel	RAB	0.33	48.90	41.00	32.05	21.35	9.65	**0.65**	0	0
		Cert-SSB	**0.35**	**56.55**	**44.00**	**35.90**	**26.30**	**10.30**	0	0	0
	Blending	RAB	**0.32**	48.40	40.70	31.55	20.75	**8.90**	**0.65**	0	0
		Cert-SSB	**0.32**	**58.55**	**42.05**	**35.30**	**24.70**	0.95	0	0	0
ImageNette	One-pixel	RAB	0.27	48.40	39.78	30.30	20.30	8.22	0	0	0
		Cert-SSB	**0.36**	**48.70**	**43.96**	**38.06**	**26.66**	**16.58**	**9.48**	**3.42**	0
	Four-pixel	RAB	0.26	48.68	40.10	29.00	18.32	7.10	0	0	0
		Cert-SSB	**0.49**	**49.00**	**42.48**	**36.76**	**26.26**	**19.08**	**11.44**	**4.00**	0
	Blending	RAB	0.27	48.72	40.14	29.62	19.02	7.16	0	0	0
		Cert-SSB	**0.48**	**49.10**	**42.86**	**38.96**	**34.48**	**27.62**	**19.06**	**6.74**	0

of Monte Carlo samples to $J = 1$, and the learning rate to $\alpha = 10^{-4}$. We initialize σ_0 using the optimal sample-specific noise level σ_x^* obtained during training.

3) Attack Settings: We evaluate the certified performance of Cert-SSB against three representative backdoor attacks: one-pixel pattern, four-pixel pattern, and random but fixed noise patterns blended across the entire image [71]. The perturbation magnitude of the attack is controlled by the ℓ_2-norm of the backdoor patterns, with $\|\Delta\|_2 = 0.1$. Following prior work [24], we inject 10% poisoned samples into the MNIST dataset and 5% into the CIFAR-10 and ImageNette datasets. The goal of these attacks is to induce the model to misclassify inputs as '0' in MNIST, 'airplane' in CIFAR-10, and 'tench' in ImageNette. In addition to the all-to-one attack, we also consider an all-to-all attack objective [32], where the compromised model alters its predictions based on the original labels. We hereby primarily focus on the perturbation magnitude and the number of injected backdoor samples without considering specific backdoor patterns.

4) Evaluation Metrics: Following previous works [59], [24], we evaluate the effectiveness of our method using empirical robust accuracy (ERA), certified robust accuracy (CRA), average empirical radius (AER), and average certified radius (ACR). Specifically, ERA is defined as the proportion of clean testing samples that are correctly classified by the Cert-SSB model, serving as an upper bound for the corresponding CRA. CRA is defined as the proportion of backdoor-triggered testing samples that are consistently classified correctly within the certified radius r. This implies that the trained Cert-SSB model not only ensures that its predictions are consistent with those of a model trained on a clean dataset but also guarantees that these predictions are equal to the ground truth labels.

TABLE III: Certified performance (*i.e.*, ERA and AER) is reported using the best results across different noise levels at each radius under the all-to-all setting. We evaluate three types of attacks (one-pixel, four-pixel, and blending) on the MNIST, CIFAR-10, and ImageNette datasets. In particular, we mark the best certification results in boldface.

Dataset↓	Attack Setting↓, Metric→	Method↓	AER	Radius (ERA↑)							
				0	0.25	0.5	0.75	1.0	1.25	1.5	1.75
MNIST	One-pixel	RAB	1.46	99.95	99.81	99.62	99.48	98.77	95.93	61.94	0
		Cert-SSB	**1.67**	**99.95**	**99.86**	**99.72**	**99.53**	**99.11**	**97.87**	**92.11**	**11.11**
	Four-pixel	RAB	1.44	**99.95**	**99.86**	99.62	87.61	98.72	95.41	46.24	0
		Cert-SSB	**1.66**	99.91	99.81	**99.76**	**99.57**	**99.11**	**98.35**	**92.77**	**5.21**
	Blending	RAB	1.46	99.91	99.86	99.67	99.34	98.72	95.56	60.57	0
		Cert-SSB	**1.66**	**99.95**	**99.91**	**99.77**	**99.72**	**99.05**	**97.97**	**92.25**	**16.17**
CIFAR-10	One-pixel	RAB	0.54	86.50	69.70	55.90	36.05	14.11	2.85	0.05	0
		Cert-SSB	**0.62**	**86.55**	**74.25**	**61.50**	**42.35**	**21.25**	**5.55**	**1.80**	**0.5**
	Four-pixel	RAB	0.55	**87.70**	69.70	56.80	38.15	17.05	3.10	0.05	0
		Cert-SSB	**0.73**	85.55	**74.75**	**68.35**	**58.15**	**39.95**	**10.65**	**1.20**	**0.05**
	Blending	RAB	0.50	87.40	49.50	24.10	2.65	0	0	0	0
		Cert-SSB	**0.67**	**86.90**	**68.10**	**50.10**	**22.35**	**22.35**	**0.45**	0	0
ImageNette	One-pixel	RAB	0.49	94.56	73.36	52.86	35.04	14.24	0	0	0
		Cert-SSB	**0.74**	**94.62**	**81.06**	**61.46**	**50.84**	**38.84**	**25.08**	**10.28**	0
	Four-pixel	RAB	0.48	**94.44**	73.66	51.48	33.24	13.46	0	0	0
		Cert-SSB	**0.65**	94.00	**77.78**	**59.64**	**44.80**	**28.36**	**14.40**	**5.68**	0
	Blending	RAB	0.48	94.66	74.28	**51.56**	33.60	13.28	0	0	0
		Cert-SSB	**0.70**	93.32	**78.16**	42.48	**52.34**	**38.26**	**17.28**	**1.40**	0

TABLE IV: Certified performance (*i.e.*, CRA and ACR) is reported using the best results across different noise levels at each radius under the all-to-all setting. We evaluate three types of attacks (one-pixel, four-pixel, and blending) on the MNIST, CIFAR-10, and ImageNette datasets. In particular, we mark the best certification results in boldface.

Dataset↓	Attack Setting↓, Metric→	Method↓	ACR	Radius (CRA↑)							
				0	0.25	0.5	0.75	1.0	1.25	1.5	1.75
MNIST	One-pixel	RAB	0.01	0.19	0.14	0.10	0.10	0.10	0	0	0
		Cert-SSB	**0.77**	**46.34**	**46.24**	**46.15**	**45.96**	**45.91**	**45.34**	**43.36**	**6.71**
	Four-pixel	RAB	0	0.10	0.10	0	0	0	0	0	0
		Cert-SSB	**0.76**	**46.29**	**46.24**	**46.05**	**45.91**	**45.86**	**45.01**	**42.36**	**4.49**
	Blending	RAB	0.01	0.52	0.52	0.52	0.426	0.28	0.14	0.14	0
		Cert-SSB	**0.77**	**46.29**	**46.24**	**46.10**	**45.96**	**45.82**	**45.34**	**42.93**	**5.39**
CIFAR-10	One-pixel	RAB	0.04	12.6	0	0	0	0	0	0	0
		Cert-SSB	**0.24**	**51.55**	**38.65**	**26.30**	**10.20**	**1.75**	**0.10**	**0**	**0**
	Four-pixel	RAB	0.04	10.90	6.80	3.60	1.20	0.10	0	0	0
		Cert-SSB	**0.30**	**48.65**	**42.60**	**35.05**	**19.50**	**0.10**	0	0	0
	Blending	RAB	0.04	11.80	6.90	3.60	1.20	0	0	0	0
		Cert-SSB	**0.29**	**47.70**	**36.90**	**29.50**	**21.30**	**9.65**	0	0	0
ImageNette	One-pixel	RAB	0.01	7.76	3.88	1.68	0.04	0	0	0	0
		Cert-SSB	**0.44**	**52.72**	**46.86**	**38.24**	**30.64**	**22.54**	**13.64**	**4.26**	0
	Four-pixel	RAB	0.01	6.52	2.84	1.04	0.48	0	0	0	0
		Cert-SSB	**0.41**	**56.88**	**49.42**	**39.58**	**28.40**	**20.78**	**11.86**	**3.28**	0
	Blending	RAB	0.01	6.64	3.20	1.36	0.32	0	0	0	0
		Cert-SSB	**0.36**	**50.58**	**42.02**	**32.24**	**24.02**	**16.58**	**8.92**	**2.70**	0

AER represents the average radius over clean test samples, while ACR denotes the average radius over all backdoor-triggered test samples. In general, higher values of ERA, CRA, AER, and ACR indicate better certification performance. In particular, we present certification curves (see Figure 6–7) to provide a more intuitive comparison of certified performance (*i.e.*, ERA and CRA) under different noise levels.

B. Main Results under the All-to-One Setting

As shown in Tables I-II, our Cert-SSB achieves the best performance under the all-to-one setting across three datasets and three attack types (one-pixel, four-pixel, and blending). For instance, on the MNIST dataset, at a radius of 1.5, ERA exceeds 72% (an improvement of approximately 30%), while CRA surpasses 45% (an increase of around 3%). Even on the more challenging ImageNette dataset, at a radius of 0.75, ERA exceeds 45% (an improvement of nearly 15%), and CRA is above 26% (an increase of 10%). In both cases, AER and ACR also improve by approximately 0.2. These experimental results validate the effectiveness of our certification method.

As shown in Figure 6, our method achieves significantly higher ERA and CRA across different noise levels (*e.g.*, 0.25, 0.5, and 1.0) compared to traditional approaches on the ImageNette dataset, further validating its superior certification performance. Notably, in these scenarios, the trade-off between model accuracy and robustness becomes more pronounced: stronger noise tends to degrade certification performance at smaller radii while improving performance at larger ones. The certification curves for the CIFAR-10 and MNIST datasets are provided in our Appendix C.

Fig. 6: Certified performance (*i.e.*, ERA and CRA) under different certification radii on the ImagNette dataset in the all-to-one setting with various noise levels (0.25, 0.5, and 1.0). The first column corresponds to the one-pixel attack, the second to the four-pixel attack, and the third to the blending attack.

Fig. 7: Certified performance (*i.e.*, ERA and CRA) under different certification radii on the ImagNette dataset in the all-to-all setting with various noise levels (0.25, 0.5, and 1.0). The first column corresponds to the one-pixel attack, the second to the four-pixel attack, and the third to the blending attack.

Fig. 8: Effect of stochastic gradient ascent iterations T.

Fig. 9: Effect of Optimized-Noise Model Count M_o.

C. Main Results under the All-to-All Setting

As shown in Tables III-IV, our Cert-SSB method also achieves the best performance under the all-to-all setting across three datasets and three attack types (one-pixel, four-pixel, and blending). For example, on the MNIST dataset, at a radius of 1.5, ERA exceeds 92% (an improvement of approximately 30%), while CRA surpasses 42% (an increase of about 40%). Even on the more challenging ImageNette dataset, at a radius of 0.75, ERA exceeds 40% (an improvement of nearly 15%), and CRA is above 20% (an increase of 20%). In both cases, AER improves by approximately 0.2, while ACR increases by 0.7 on MNIST and 0.4 on ImageNette. These results validate the effectiveness of our method.

As shown in Figure 7, both ERA and CRA under different noise levels (*e.g.*, 0.25, 0.5, 1.0) are also significantly higher than those of traditional methods on the ImageNette dataset, with even more pronounced improvements compared to the all-to-one setting. This experimental result further demonstrates the superior certification performance of our Cert-SSB method. The trade-off between model accuracy and robustness is consistent with that in the all-to-one setting. The certification curves for CIFAR-10 and MNIST are in our Appendix C.

D. Discussions

In this section, we discuss the effectiveness of Cert-SSB with different key hyperparameters. For simplicity, we take the one-pixel attack on the ImageNette dataset under the all-to-one setting as an example for discussion.

Effect of Stochastic Gradient Ascent Iterations T. As shown in Figure 8, both the empirical robust accuracy and

certified robust accuracy consistently increase as T increases, particularly at larger certification radii. The underlying reason is that a larger T allows for a more optimized smoothing parameter σ_x^* for each input x, thereby expanding the certified radius and leaving room for further improvements in strong defense methods. However, excessively increasing T also leads to higher computational costs. Therefore, defenders must choose an appropriate T based on specific requirements.

Effect of Optimized-Noise Model Count M_o. Considering that the final prediction is obtained through an ensemble of multiple models, we further investigate the impact of the number of optimized-noise models on certification performance (*i.e.*, ERA and CRA) under different certification radii. Specifically, we trained 50 models with fixed noise σ_0 and 150 models with optimized noise σ_x^*, forming an ensemble of 200 models (*i.e.*, $M_f = 50$, $M_o = 150$). This setup is compared against two baselines: one where all models are trained with fixed noise (*i.e.*, $M_f = 200$, $M_o = 0$), and another where all models are trained with optimized noise (*i.e.*, $M_f = 0$, $M_o = 200$). We evaluate the certification performance of these three settings under various noise levels (*i.e.*, $\sigma = 0.25, 0.5, 1.0$) and across different certification radii. As shown in Figure 9, the ensemble trained entirely with optimized noise achieves significantly higher ERA and CRA at all certification radii, compared to those incorporating a portion of fixed-noise models. These results indicate that increasing the number of optimized-noise models helps improve the robustness of the ensemble, while introducing fixed-noise models may limit the overall performance.

(a) clean image   (b) clean image   (c) clean image   (d) clean image

(e) $\sigma_{\boldsymbol{x}}^* = 0.311$ (f) $\sigma_{\boldsymbol{x}}^* = 0.248$ (g) $\sigma_{\boldsymbol{x}}^* = 0.236$ (h) $\sigma_{\boldsymbol{x}}^* = 0.299$

Fig. 10: Examples of clean and perturbed images using optimized noise $\sigma_{\boldsymbol{x}}^*$ (initialized from $\sigma_0 = 0.25$).

Visualization of Optimized Noise $\sigma_{\boldsymbol{x}}^*$. We hereby randomly select two input images from two categories, respectively, and perform noise optimization for each input image $\boldsymbol{x}$, starting from a fixed initial noise level of $\sigma_0 = 0.25$, to obtain the optimal noise $\sigma_{\boldsymbol{x}}^*$ that maximizes the certified radius. As shown in Figure 10, the optimized noise values vary significantly across different inputs, with some being larger and others smaller. Notably, even within the same category, there exist considerable differences among the optimized results. These findings further demonstrate the necessity of adaptively optimizing the noise for each individual input.

E. The Analysis of Computational Complexity

In this section, we analyze the computational complexity of Cert-SSB under an experimental setup running Ubuntu 22.04, equipped with an Intel Xeon Silver 4214 CPU, a Tesla V100-PCIE-32GB GPU, and CUDA 12.0. We particularly focus on the computational costs of the noise optimization and storage-update-based certification processes.

The Complexity of Noise Optimization. Let N, λ, T denote the number of samples in the training set, the poisoning rate, and the number of stochastic gradient ascent (SGA) iterations used for noise optimization, respectively. Since Cert-SSB first poisons a small subset of selected samples and then performs T rounds of SGA optimization on each poisoned sample to obtain the optimized noise, its computational complexity is $\mathcal{O}(N \cdot \lambda \cdot T)$. Furthermore, Cert-SSB supports parallel processing, as the optimization process for each sample is independent. In practice, Cert-SSB requires only one round of SGA iteration to obtain sufficiently optimized noise. For instance, on the CIFAR-10 dataset, computing optimized noise for a single sample takes approximately 2 seconds. Therefore, the additional computational overhead of our method in the noise optimization phase is acceptable.

The Complexity of Storage-update-based Certification. In this stage, the defender adopts a storage-update-based method to dynamically update the certification process for n samples, ensuring that the certification regions of different samples do not overlap. The computational complexity of this step is $\mathcal{O}(n)$, and it also supports parallel processing. For example, in batch mode, executing the storage-based update certification method on 2000 testing samples takes only about 2 seconds.

Arguably, this computational overhead is negligible compared to the improvement in certification performance.

VI. CONCLUSION

In this paper, we revisited existing randomized smoothing-based certified backdoor defense methods and revealed that using fixed noise for all samples led to suboptimal certification performance. To address this issue, we proposed a sample-specific certified backdoor defense method (*i.e.*, Cert-SSB), which employed stochastic gradient ascent to iteratively optimize sample-specific noise in order to maximize the certification radius. The optimized noise was then injected into the poisoned training set to retrain multiple smoothed models, whose predictions are aggregated to obtain the final robust prediction. Since existing certification methods typically assumed a fixed noise level and thus did not apply to our setting, we further introduced a storage-update-based certification approach to improve certification accuracy and reliability. Extensive experiments on multiple benchmark datasets demonstrated that Cert-SSB significantly outperformed existing methods in terms of certification performance. We hope this work inspires future exploration of how sample-specific noise relates to model decision boundaries for better personalized certification.

REFERENCES

[1] X. Yang, X. Jia, D. Gong, D.-M. Yan, Z. Li, and W. Liu, "Larnet: Lie algebra residual network for face recognition," in *ICML*, 2021.

[2] Z. Deng, X. Peng, Z. Li, and Y. Qiao, "Mutual component convolutional neural networks for heterogeneous face recognition," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 3102–3114, 2019.

[3] M. Luo, H. Wu, H. Huang, W. He, and R. He, "Memory-modulated transformer network for heterogeneous face recognition," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2095–2109, 2022.

[4] M. Fan, Z. Si, X. Xie, Y. Liu, and T. Liu, "Text backdoor detection using an interpretable rnn abstract model," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4117–4132, 2021.

[5] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 1, pp. 5–22, 2022.

[6] 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 Transactions on Information Forensics and Security*, vol. 19, pp. 5852–5866, 2024.

[7] Z. Zhang, X. Yuan, L. Zhu, J. Song, and L. Nie, "Badcm: Invisible backdoor attack against cross-modal learning," *IEEE Transactions on Image Processing*, vol. 33, pp. 2558–2571, 2024.

[8] R. S. S. Kumar, M. Nyström, J. Lambert, A. Marshall, M. Goertzel, A. Comissoneru, M. Swann, and S. Xia, "Adversarial machine learning-industry perspectives," in *SPW*. IEEE, 2020, pp. 69–75.

[9] [Online]. Available:https://news.uchicago.edu/story/computer-scientists-design-way-close-backdoors-ai-based-security-systems.

[10] Z. Xiang, Z. Xiong, and B. Li, "Umd: Unsupervised model detection for x2x backdoor attacks," in *ICML*, 2023.

[11] Z. Xiang, D. J. Miller, and G. Kesidis, "Revealing backdoors, post-training, in dnn classifiers via novel inference on optimized perturbations inducing group misclassification," in *ICASSP*, 2020.

[12] Z. Xiang, Z. Xiong, and B. Li, "Cbd: A certified backdoor detector based on local dominant probability," in *NeurIPS*, 2023.

[13] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "Strip: A defence against trojan attacks on deep neural networks," in *ACSAC*, 2019.

[14] H. Qiu, H. Ma, Z. Zhang, A. Abuadbba, W. Kang, A. Fu, and Y. Gao, "Towards a critical evaluation of robustness for deep learning backdoor countermeasures," *IEEE Transactions on Information Forensics and Security*, 2023.

[15] 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," in *ICLR*, 2023.

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

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

[18] Q. Duan, Z. Hua, Q. Liao, Y. Zhang, and L. Y. Zhang, "Conditional backdoor attack via jpeg compression," in *AAAI*, 2024.

[19] A. Levine and S. Feizi, "Deep partition aggregation: Provable defense against general poisoning attacks," in *ICLR*, 2021.

[20] W. Wang, A. J. Levine, and S. Feizi, "Improved certified defenses against data poisoning with (deterministic) finite aggregation," in *ICML*, 2022.

[21] K. Rezaei, K. Banihashem, A. Chegini, and S. Feizi, "Run-off election: Improved provable defense against data poisoning attacks," in *ICML*, 2023.

[22] Y. Zhang, A. Albarghouthi, and L. D'Antoni, "Pecan: A deterministic certified defense against backdoor attacks," *arXiv preprint arXiv:2301.11824*, 2023.

[23] B. Wang, X. Cao, N. Z. Gong *et al.*, "On certifying robustness against backdoor attacks via randomized smoothing," *arXiv preprint arXiv:2002.11750*, 2020.

[24] M. Weber, X. Xu, B. Karlaš, C. Zhang, and B. Li, "Rab: Provable robustness against backdoor attacks," in *IEEE S&P*, 2023.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[26] 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*, vol. 139, p. 109512, 2023.

[27] R. Zhai, C. Dan, D. He, H. Zhang, B. Gong, P. Ravikumar, C.-J. Hsieh, and L. Wang, "Macer: Attack-free and scalable robust training via maximizing certified radius," in *ICLR*, 2020.

[28] S. Li, T. Dong, B. Z. H. Zhao, M. Xue, S. Du, and H. Zhu, "Backdoors against natural language processing: A review," *IEEE Security & Privacy*, vol. 20, no. 5, pp. 50–59, 2022.

[29] Y. Ding, Z. Wang, Z. Qin, E. Zhou, G. Zhu, Z. Qin, and K.-K. R. Choo, "Backdoor attack on deep learning-based medical image encryption and decryption network," *IEEE Transactions on Information Forensics and Security*, 2023.

[30] K. Gao, J. Bai, B. Wu, M. Ya, and S.-T. Xia, "Imperceptible and robust backdoor attack in 3d point cloud," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1267–1282, 2023.

[31] Y. Gao, Y. Li, X. Gong, Z. Li, S.-T. Xia, and Q. Wang, "Backdoor attack with sparse and invisible trigger," *IEEE Transactions on Information Forensics and Security*, 2024.

[32] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47230–47244, 2019.

[33] 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 Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2088–2105, 2020.

[34] Y. Li, H. Zhong, X. Ma, Y. Jiang, and S.-T. Xia, "Few-shot backdoor attacks on visual object tracking," in *ICLR*, 2022.

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

[36] J. Bai, K. Gao, D. Gong, S.-T. Xia, Z. Li, and W. Liu, "Hardly perceptible trojan attack against neural networks with bit flips," in *ECCV*, 2022.

[37] Y. Li, M. Zhu, X. Yang, Y. Jiang, T. Wei, and S.-T. Xia, "Black-box dataset ownership verification via backdoor watermarking," *IEEE Transactions on Information Forensics and Security*, 2023.

[38] J. Guo, Y. Li, L. Wang, S.-T. Xia, H. Huang, C. Liu, and B. Li, "Domain watermark: Effective and harmless dataset copyright protection is closed at hand," in *NeurIPS*, 2023.

[39] C. Wei, Y. Wang, K. Gao, S. Shao, Y. Li, Z. Wang, and Z. Qin, "Pointncbw: Towards dataset ownership verification for point clouds via negative clean-label backdoor watermark," *IEEE Transactions on Information Forensics and Security*, 2024.

[40] B. Li, Y. Wei, Y. Fu, Z. Wang, Y. Li, J. Zhang, R. Wang, and T. Zhang, "Towards reliable verification of unauthorized data usage in personalized text-to-image diffusion models," in *IEEE S&P*, 2025.

[41] Y. Li, L. Zhu, X. Jia, Y. Bai, Y. Jiang, S.-T. Xia, X. Cao, and K. Ren, "Move: Effective and harmless ownership verification via embedded external features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[42] S. Shao, Y. Li, H. Yao, Y. He, Z. Qin, and K. Ren, "Explanation as a watermark: Towards harmless and multi-bit model ownership verification via watermarking feature attribution," in *NDSS*, 2025.

[43] B. Yi, T. Huang, S. Chen, T. Li, Z. Liu, C. Zhixuan, and Y. Li, "Probe before you talk: Towards black-box defense against backdoor unalignment for large language models," in *ICLR*, 2025.

[44] Y. Chen, S. Shao, E. Huang, Y. Li, P.-Y. Chen, Z. Qin, and K. Ren, "Refine: Inversion-free backdoor defense via model reprogramming," in *ICLR*, 2025.

[45] 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," in *IEEE S&P*, 2019.

[46] J. Hayase, W. Kong, R. Somani, and S. Oh, "Spectre: Defending against backdoor attacks using robust statistics," in *ICML*, 2021.

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

[48] 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," in *NeurIPS*, 2023.

[49] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *RAID*, 2018.

[50] B. Li, Y. Cai, H. Li, F. Xue, Z. Li, and Y. Li, "Nearest is not dearest: Towards practical defense against quantization-conditioned backdoor attacks," in *CVPR*, 2024.

[51] E. Chou, F. Tramer, and G. Pellegrino, "Sentinet: Detecting localized universal attacks against deep learning systems," in *IEEE S&P Workshops*. IEEE, 2020, pp. 48–54.

[52] L. Hou, R. Feng, Z. Hua, W. Luo, L. Y. Zhang, and Y. Li, "Ibd-psc: Input-level backdoor detection via parameter-oriented scaling consistency," in *ICML*, 2024.

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

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

[55] P. W. Koh, J. Steinhardt, and P. Liang, "Stronger data poisoning attacks break data sanitization defenses," *Machine Learning*, pp. 1–47, 2022.

[56] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the tails: Yes, you really can backdoor federated learning," in *NeurIPS*, 2020.

[57] J. Jia, X. Cao, and N. Z. Gong, "Intrinsic certified robustness of bagging against data poisoning attacks," in *AAAI*, 2021.

[58] J. Jia, Y. Liu, X. Cao, and N. Z. Gong, "Certified robustness of nearest neighbors against data poisoning and backdoor attacks," in *AAAI*, 2022.

[59] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *ICML*, 2019.

[60] J. Neyman and E. S. Pearson, "Ix. on the problem of the most efficient tests of statistical hypotheses," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 231, no. 694-706, pp. 289–337, 1933.

[61] J. Guo, Y. Li, R. Chen, Y. Wu, C. Liu, and H. Huang, "Zeromark: Towards dataset ownership verification without disclosing watermarks," in *NeurIPS*, 2024.

[62] F. Croce and M. Hein, "Minimally distorted adversarial examples with a fast adaptive boundary attack," in *ICML*, 2020.

[63] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[64] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," *Technical report, University of Toronto*, 2009.

[65] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, pp. 229–256, 1992.

[66] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014.

[67] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *ICML*, 2014.

[68] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[69] J. Howard, "Imagenette," https://github.com/fastai/imagenette/, 2020, [Online].

[70] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[71] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

APPENDIX

A. The Detailed Algorithm for Cert-SSB

Algorithm 1 details the training procedure of Cert-SSB.

Algorithm 1 Cert-SSB Training: Train the Model with Optimized Noise

---

1: **Input**: Stochastic gradient ascent iterations T, poisoned training dataset $\mathcal{D}_p = \{(\boldsymbol{x}_i + \boldsymbol{\Delta}_i, \tilde{y}_i)\}_{i=1}^n$, initial noise scale σ_0, number of models M, learning rate α
2: **Output**: Model collection $\{(g_1, \mu_1), \ldots, (g_M, \mu_M)\}$
3: **for** $m = 1, \ldots, M$ **do**
4:    **Step 1: Optimize Sample-Specific Noise $\sigma_{\boldsymbol{x}}^*$**
5:    Initialize $\sigma_{\boldsymbol{x}}^0 = \sigma_0$
6:    **for** $t = 0, \ldots, T-1$ **do**
7:       Sample $\hat{Z}_1, ..., \hat{Z}_J (\hat{D}_1, ..., \hat{D}_J) \sim \mathcal{N}(0, I)$
8:       Compute class probabilities:
      $\varphi(\sigma_{\boldsymbol{x}}^j) = \frac{1}{J}\sum_{j=1}^J f((\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}} + \sigma_x^t \hat{Z}_j, \mathcal{D} + \boldsymbol{\delta} + \sigma_x^t \hat{D}_j))$
9:       Define $F_A(\sigma_x^t) = \max_y \varphi_y$, $y_A = \arg\max_y \varphi_y$, and $F_B(\sigma_{\boldsymbol{x}}^t) = \max_{y \neq y_A} \varphi_y$
10:      Compute certified radius:
      $r(\sigma_{\boldsymbol{x}}^t) = \frac{\sigma_{\boldsymbol{x}}^t}{2}(\Phi^{-1}(F_A) - \Phi^{-1}(F_B))$
11:      Update $\sigma_{\boldsymbol{x}}^{t+1} = \sigma_x^t + \alpha \nabla_{\sigma_{\boldsymbol{x}}^t} r(\sigma_{\boldsymbol{x}}^t)$
12:    **end for**
13:    Set $\sigma_{\boldsymbol{x}_i}^* = \sigma_{\boldsymbol{x}}^T$ for all t
14:    **Step 2: Robust Training Process**
15:    Sample noise vectors $b_{m_1}, ..., b_{m_n} \sim \prod_{i=1}^n \mathcal{N}(0, I)$
16:    Construct augmented dataset: $\mathcal{D}_m = \{(\boldsymbol{x}_i + \boldsymbol{\Delta}_i + \sigma_{\boldsymbol{x}_i}^* b_{m,i})_{i=1}^n\})$
17:    Train model $g_m(\boldsymbol{x}, \mathcal{D}, \sigma_{\boldsymbol{x}}^*) = \text{train\_model}(\mathcal{D}_m)$
18:    Sample $\mu_m \sim \mathcal{N}(0, \sigma_0^2 I_d)$ deterministically using random seed based on $hash(g_m(\boldsymbol{x}, \mathcal{D}, \sigma_{\boldsymbol{x}}^*))$
19: **end for**

B. Proof of Theorem 1

Here we provide the proof for Theorem 1. As the proof is based on statistical hypothesis testing, we begin by defining the type-I and type-II error probabilities. Formally, we denote the type-I error probability under the null hypothesis H_0 as $\alpha(\phi) = \alpha(\phi; H_0)$ and the type-II error probability under the alternative hypothesis H_1 as $\beta(\phi) = \beta(\phi; H_1)$. To facilitate the proof of Theorem 1, we first state and apply Lemma 1, which establishes a key robustness condition based on hypothesis testing. This result ensures that the classifier's decision remains stable under specified probability constraints, even in the presence of perturbations.

Lemma 1 ([24]). *Let g be the sample-specific smoothed classifier defined as $g(\boldsymbol{x}, \mathcal{D}, \sigma) = \arg\max_y \mathcal{P}_{\boldsymbol{\epsilon}(Z,D)}\big(f(\boldsymbol{x} + Z, \mathcal{D} + D) = y\big)$, where the smoothing distribution is given by $X := (Z, D)$, with Z taking values in $\mathbb{R}^d$ and D being a collection of n independent $\mathbb{R}^d$-valued random variables: $D = (D^{(1)}, \cdots, D^{(n)}) = (\sigma_{\boldsymbol{x}_1}^* \boldsymbol{\epsilon}^1, \cdots, \sigma_{\boldsymbol{x}_n}^* \boldsymbol{\epsilon}^n)$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$. Let $\mathcal{B}_{\boldsymbol{x}} \in R^d$ and let $\boldsymbol{\delta} = (\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2, ..., \boldsymbol{\Delta}_n)$ for backdoor*

patterns $\boldsymbol{\Delta}_i \in R^d$. *Let $y_A \in \mathcal{Y}$ and let $P_A, P_B \in [0,1]$ such that $y_A = g(\boldsymbol{x}, \mathcal{D}, \sigma)$ and*

$$\mathcal{P}_{\boldsymbol{\epsilon}}(g(\boldsymbol{x}, \mathcal{D}, \sigma) = y_A) \geq P_A \geq P_B \geq \max_{y \neq y_A} \mathcal{P}_{\boldsymbol{\epsilon}}(g(\boldsymbol{x}, \mathcal{D}, \sigma) = y), \tag{1}$$

If the optimal type II errors, for testing the null $X \sim H_0$ against the alternative $X + (\mathcal{B}_{\boldsymbol{x}}, \boldsymbol{\delta}) \sim H_1$, satisfy

$$\beta^*(1 - P_A; H_1) + \beta^*(P_B; H_1) > 1, \tag{2}$$

then it is guaranteed that $y_A = g(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}}, \mathcal{D} + \boldsymbol{\delta}, \sigma)$.

Building upon Lemma 1, we derive Theorem 1, which formally guarantees robustness by providing an explicit certified radius within which the classifier's prediction remains unchanged. The key idea is to ensure that the likelihood ratio test satisfies the probability bounds established earlier.

Theorem 1 (Certified Robustness of Cert-SSB). *Let $\mathcal{B}_{\boldsymbol{x}} \in R^d$ and let $\boldsymbol{\delta} := (\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2, ..., \boldsymbol{\Delta}_n)$ for backdoor patterns $\boldsymbol{\Delta}_i \in R^d$, and let $\mathcal{D}$ be a training set, and let smoothing noise $\hat{Z} \sim \mathcal{N}(0, I)$, $\hat{D} \sim \mathcal{N}(0, I)$. Let $y_A \in \mathcal{Y}$, such as $y_A = g(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}}, \mathcal{D} + \boldsymbol{\delta})$ with class probabilities satisfying $\mathcal{P}_{\boldsymbol{\epsilon}(\hat{Z},\hat{D})}(f(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}} + \sigma_{\boldsymbol{x}}^* \hat{Z}, \mathcal{D} + \boldsymbol{\delta} + \sigma_{\boldsymbol{x}}^* \hat{D}) = y_A) \geq P_A \geq P_B \geq \max_{y \neq y_A} \mathcal{P}_{\boldsymbol{\epsilon}(\hat{Z},\hat{D})}(f(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}} + \sigma_{\boldsymbol{x}}^* \hat{Z}, \mathcal{D} + \boldsymbol{\delta} + \sigma_{\boldsymbol{x}}^* \hat{D}) = y)$. Then, we have $g(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}}, \mathcal{D}) = g(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}}, \mathcal{D} + \boldsymbol{\delta}) = y_A$ for all backdoor patterns $\boldsymbol{\Delta}$ satisfying $\sqrt{\sum_{i=1}^n \|\boldsymbol{\Delta}_i\|_2^2} \leq r(g; \sigma_{\boldsymbol{x}_i}^*)$, where the certified robust radius r is given by*

$$r(g; \sigma_{\boldsymbol{x}_i}^*) = \frac{\sqrt{\sum_{i=1}^n (\sigma_{\boldsymbol{x}_i}^*)^2}}{2} \left(\Phi^{-1}(P_A(\sigma_{\boldsymbol{x}_i}^*)) - \Phi^{-1}(P_B(\sigma_{\boldsymbol{x}_i}^*))\right). \tag{3}$$

Proof. We prove this theorem by directly applying Lemma 1. Consider the smoothing noise jointly distributed as $X = (Z, D)$ and define the perturbed and unperturbed input distributions as follows: $\tilde{X} = (\mathcal{B}_{\boldsymbol{x}}, \boldsymbol{\delta}) + X$, $\tilde{X}' := (\mathcal{B}_{\boldsymbol{x}}, 0) + X$. Correspondingly, the probability of the smoothed classifier can be expressed as: $\mathcal{P}_{\boldsymbol{\epsilon}}(\tilde{g}(\boldsymbol{x}, \mathcal{D}) = y) = \mathcal{P}_{\boldsymbol{\epsilon}}(g(\boldsymbol{x} + \mathcal{B}_{\boldsymbol{x}}, \mathcal{D} + \boldsymbol{\delta}) = y)$. By assumption, the classifier satisfies:

$$\mathcal{P}_{\boldsymbol{\epsilon}}(\tilde{g}(\boldsymbol{x}, \mathcal{D}) = y_A) \geq P_A, \quad \max_{y \neq y_A} \mathcal{P}_{\boldsymbol{\epsilon}}(\tilde{g}(\boldsymbol{x}, \mathcal{D}) = y) \leq P_B. \tag{4}$$

Applying Lemma 1, it follows that if $\beta(\phi_a) + \beta(\phi_b) > 1$, then the classifier output remains unchanged under perturbations, ensuring: $\tilde{g}(\boldsymbol{x}, \mathcal{D}) = \tilde{g}(\boldsymbol{x}, \mathcal{D} - \boldsymbol{\delta}) = y_A$. To verify this condition, we analyze the likelihood ratio between $\tilde{X}$ and $\tilde{X}'$ at $z = (\boldsymbol{x}, b)$, given by $\Lambda(z) = \exp\{\sum_{i=1}^n (-\frac{\|\boldsymbol{\Delta}_i\|^2}{2(\sigma_{\boldsymbol{x}_i}^*)^2} + \frac{b_i^T \boldsymbol{\Delta}_i}{(\sigma_{\boldsymbol{x}_i}^*)^2})\}$. Since Gaussian distributions assign probability density rather than discrete probabilities, any likelihood ratio test takes the form:

$$\phi_t(z) = \begin{cases} 1 & \Lambda(z) \geq t, \\ 0 & \Lambda(z) < t. \end{cases} \tag{5}$$

To compute the error probabilities, the threshold for $P \in [0, 1]$ is given by: $t_P := \exp(\Phi^{-1}(P)\sqrt{\sum_{i=1}^n \frac{\|\boldsymbol{\Delta}_i\|_2^2}{(\sigma_{\boldsymbol{x}_i}^*)^2}} - \sum_{i=1}^n \frac{\|\boldsymbol{\Delta}_i\|_2^2}{2(\sigma_{\boldsymbol{x}_i}^*)^2})$ and note that $\alpha(\phi(t_P)) = 1 - P$ since

$$\alpha(\phi(t_P)) = 1 - \Phi\left(\frac{\log(t_P) + \frac{1}{2}\sum_{i=1}^n \frac{\|\boldsymbol{\Delta}_i\|_2^2}{(\sigma_{\boldsymbol{x}_i}^*)^2}}{\sqrt{\sum_{i=1}^n \frac{\|\boldsymbol{\Delta}_i\|_2^2}{(\sigma_{\boldsymbol{x}_i}^*)^2}}}\right), \text{ where } \Phi \text{ is}$$

Fig. 1: Certified performance (*i.e.*, ERA, CRA) under different certification radii on the MNIST dataset in the all-to-one setting with various noise levels (0.12, 0.25, 0.5, and 1.0). The first column corresponds to the one-pixel attack, the second to the four-pixel attack, and the third to the blending attack.

Fig. 2: Certified performance (*i.e.*, ERA, CRA) under different certification radii on the MNIST dataset in the all-to-all setting with various noise levels (0.12, 0.25, 0.5, and 1.0). The first column corresponds to the one-pixel attack, the second to the four-pixel attack, and the third to the blending attack.

Fig. 3: Certified performance (*i.e.*, ERA, CRA) under different certification radii on the CIFAR-10 dataset in the all-to-one setting with various noise levels (0.12, 0.25, 0.5, and 1.0). The first column corresponds to the one-pixel attack, the second to the four-pixel attack, and the third to the blending attack.

Fig. 4: Certified performance (*i.e.*, ERA, CRA) under different certification radii on the CIFAR-10 dataset in the all-to-all setting with various noise levels (0.12, 0.25, 0.5, and 1.0). The first column corresponds to the one-pixel attack, the second to the four-pixel attack, and the third to the blending attack.

the CDF of the standard normal distribution. For the test $\phi_a = \phi_{t_a}$ with $t_a \equiv t_{P_A}$, the type I error probability satisfies: $\alpha(\phi_a) = 1 - P_A$. Similarly, for $\phi_b = \phi_{t_b}$ with $t_b \equiv t_{1-P_B}$, we have: $\alpha(\phi_a) = P_B$. Evaluating the type II error probabilities, we obtain: $\beta(\phi_a) = \Phi(\Phi^{-1}(P_A) - \sqrt{\sum_{i=1}^{n} \frac{\|\Delta_i\|_2^2}{(\sigma_{x_i}^*)^2}})$, $\beta(\phi_b) = \Phi(\Phi^{-1}(1 - P_B) - \sqrt{\sum_{i=1}^{n} \frac{\|\Delta_i\|_2^2}{(\sigma_{x_i}^*)^2}})$. Substituting these into condition $\beta(\phi_a) + \beta(\phi_b) > 1$, we conclude that the inequality holds if and only if: $\sqrt{\sum_{i=1}^{n} \|\Delta_i\|_2^2} < \frac{\sqrt{\sum_{i=1}^{n}(\sigma_{x_i}^*)^2}}{2}(\Phi^{-1}(P_A) - \Phi^{-1}(P_B))$. Rearranging, the certified robust radius is obtained as: $r(g; \sigma_{x_i}^*) = \frac{\sqrt{\sum_{i=1}^{n}(\sigma_{x_i}^*)^2}}{2}(\Phi^{-1}(P_A) - \Phi^{-1}(P_B))$. Thus, the classifier g remains robust against backdoor patterns, ensuring: $\tilde{g}(\boldsymbol{x}, \mathcal{D}) = \tilde{g}(\boldsymbol{x}, \mathcal{D} - \boldsymbol{\delta}) = y_A$.   □

C. Additional Experimental Results

We hereby present additional experimental results, with all experimental settings consistent with those described in

Section V-B. Figures 1-2 illustrate the certification curves for the MNIST dataset under the all-to-one and all-to-all settings, respectively. Figures 3-4 show the certification curves for the CIFAR-10 dataset under the all-to-one and all-to-all settings. The results are consistent with the conclusions in Sections V-B and V-C, demonstrating that our method maintains strong certification performance (*i.e.*, empirical robust accuracy (ERA) and certified robust accuracy (CRA)) across different datasets. In particular, in the all-to-all setting, our method achieves a significant improvement in certified robust accuracy, further validating its effectiveness and generalization capability.