2023 IEEE Symposium on Security and Privacy (SP) | 978-1-6654-9336-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/SP46215.2023.10179451

RAB: Provable Robustness Against Backdoor Attacks

Maurice Weber[†] * Xiaojun Xu[‡] * Bojan Karlaš[†] Ce Zhang[†] Bo Li[‡]
[†] ETH Zurich, Switzerland {maurice.weber, karlasb, ce.zhang}@inf.ethz.ch
[‡] University of Illinois at Urbana-Champaign, USA {xiaojun3, lbo}@illinois.edu

Abstract*—Recent studies have shown that deep neural networks (DNNs) are vulnerable to adversarial attacks, including evasion and backdoor (poisoning) attacks. On the defense side, there have been intensive efforts on improving both empirical and provable robustness against evasion attacks; however, the provable robustness against backdoor attacks still remains largely unexplored. In this paper, we focus on certifying the machine learning model robustness against general threat models, especially backdoor attacks. We first provide a unified framework via randomized smoothing techniques and show how it can be instantiated to certify the robustness against both evasion and backdoor attacks. We then propose the *first* robust training process, RAB, to smooth the trained model and certify its robustness against backdoor attacks. We theoretically prove the robustness bound for machine learning models trained with RAB and prove that our robustness bound is tight. In addition, we theoretically show that it is possible to train the robust smoothed models efficiently for simple models such as K-nearest neighbor classifiers, and we propose an exact smooth-training algorithm that eliminates the need to sample from a noise distribution for such models. Empirically, we conduct comprehensive experiments for different machine learning (ML) models such as DNNs, support vector machines, and K-NN models on MNIST, CIFAR-10, and ImageNette datasets and provide the first benchmark for certified robustness against backdoor attacks. In addition, we evaluate K-NN models on a spambase tabular dataset to demonstrate the advantages of the proposed exact algorithm. Both the theoretic analysis and the comprehensive evaluation on diverse ML models and datasets shed light on further robust learning strategies against general training time attacks.

1. Introduction

Building machine learning algorithms that are robust to adversarial attacks has been an emerging topic over the last decade. There are mainly two different types of adversarial attacks: (1) *evasion attacks*, in which the attackers manipulate the test examples against a trained machine learning (ML) model, and (2) *data poisoning attacks*, in which the attackers are allowed to perturb the training set. Both types of attacks have attracted intensive interests from academia as well as industry [14], [53], [57], [61].

In response, several empirical solutions have been proposed as defenses against evasion attacks [5], [31], [55], [59]. For instance, adversarial training has been proposed to retrain the ML models with generated adversarial examples [33]; quantization has been applied to either inputs or neural network weights to defend against potential adversarial instances [55]. However, recent studies have shown that these defenses are not resilient against intelligent adversaries responding dynamically to the deployed defenses [1], [5].

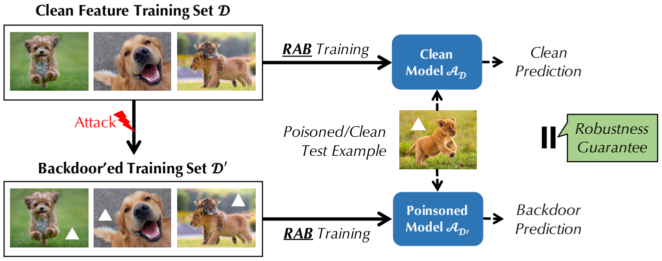
Clean Feature Training Set $\mathcal{D}$
RAB Training
Clean Model $\mathcal{A}_{\mathcal{D}}$
Clean Prediction
Attack
Poisoned/Clean Test Example
Robustness Guarantee
Backdoor'ed Training Set $\mathcal{D}'$
RAB Training
Poisoned Model $\mathcal{A}_{\mathcal{D}'}$
Backdoor Prediction

Figure 1: In this paper, we define a robust training process RAB against backdoor attacks. Given a poisoned dataset $\mathcal{D}'$ — produced by adding backdoor patterns Δ to some instances in the dataset $\mathcal{D}$ with clean features — this robust training process guarantees that, for all test examples x, $\mathcal{A}_{\mathcal{D}'}(x) = \mathcal{A}_{\mathcal{D}}(x)$, with high probability when the magnitude of the backdoor pattern Δ is within the certification radius.

As a result, one recent, exciting line of research aims to develop *certifiably robust* algorithms against *evasion attacks*, including both deterministic and probabilistic certification approaches [28]. In particular, among these certified robustness approaches, only randomized smoothing and its variations are able to provide certified robustness against evasion attacks on large-scale datasets such as ImageNet [10], [25], [58]. Intuitively, the randomized smoothing-based approaches are able to certify the robustness of a smoothed classifier, by outputting a consistent prediction for an adversarial input as long as the perturbation is within a certain radius. The smoothed classifier is obtained by taking the expectation over the possible outputs given a set of randomized inputs which are generated by adding noise drawn from a certain distribution.

Despite these recent developments on certified robustness against *evasion attacks*, only empirical studies have been conducted to defend against *backdoor attacks* [13], [16], [27], [50], and the question of how to improve and certify the robustness of given machine learning models against backdoor attacks remains largely unanswered. To the best of our knowledge, there is no certifiably robust strategy to deal with backdoor attacks yet. Naturally, we ask: *Can we develop certifiably robust ML models against backdoor attacks?*

It is clear that extending existing certification methods against evasion attacks to certifying training-time attacks is challenging given these two significantly different threat models. For instance, even certifying a label flipping

DOI 10.1109/SP46215.2023.00037

training-time attack is non-trivial as illustrated in a concurrent work [40], which proposes to certify against a label flipping attack by setting a limit to how many labels in the training set may be flipped such that it does not affect the final prediction leveraging randomized smoothing. As backdoor attacks involve both label flipping and instance pattern manipulations, providing certifications can be even more challenging.

In particular, to carry out a backdoor attack, an attacker adds small backdoor patterns to a subset of training instances such that the trained model is biased toward test images with the same patterns [8], [15]. Such attacks can be applied to various real-world scenarios such as online face recognition systems [8], [27]. In this paper, we present the first certification process, referred to as RAB, which offers provable robustness for ML models against backdoor attacks. As shown in Figure 1, our **certification goal** is to guarantee *that a test instance, which may contain backdoor patterns, will be classified the same, independent of whether the models were trained on data with or without backdoors, as long as the embedded backdoor patterns are within an L_p-ball of radius R.* We formally define the corresponding threat model and our certification goal in Section 3.

Our approach to achieving this is mainly inspired by randomized smoothing, a technique to certify robustness against evasion attacks [10], but goes significantly beyond it due to the different settings (e.g. evasion and backdoor attacks). Our **first step/contribution** is to develop a general theoretical framework to generalize randomized smoothing to a much larger family of functions and smoothing distributions. This allows us to support cases in which a classifier is a function that takes as input a test instance *and* a training set. With our framework, we can *(1) provide robustness certificates against both evasion and dataset poisoning attacks; (2) certify any classifier which takes as input a tuple of test instance and training dataset* and *(3) prove that the derived robustness bound is tight*. Given this general framework, we can enable a basic version of the proposed RAB framework. At a high level, as shown in Figure 2, given training set $\mathcal{D}$, RAB generates N additional "smoothed" training sets $\mathcal{D}+\epsilon_i$ by adding noise ϵ_i ($i \in \{1, \ldots, N\}$) drawn from a certain smoothing distribution and, for each of these N training sets, a corresponding classifier is trained resulting in an ensemble of N different classifiers. These models are then aggregated to generate a "smoothed classifier" for which we prove that its output will be consistent regardless of whether there are backdoors added during training, as long as the backdoor patterns satisfy certain conditions.

However, this basic version is not enough to provide satisfactory certified robustness against backdoor attacks. When we instantiate our theoretical framework with a practical training pipeline to provide certified robustness against backdoor attacks, we need to further develop nontrivial techniques to improve two aspects: (1) Certification Radius and (2) Certification Efficiency. Our **second step/contribution** are two non-trivial technical optimizations. (1) To improve the *certification radius*, we certify DNN classifiers with a data augmentation the scheme enabled by hash functions and, in the meantime, explore different design decisions such as the smoothness of the training process. This provides additional guidance for improving the certified robustness against backdoor attacks and we hope that it can inspire other researches in the future. (2) To improve the *certification efficiency*, we observed that for certain families of classifiers, namely K-nearest neighbor classifiers, we can develop an efficient algorithm to compute the smoothing result *exactly, eliminating the need to resort to Monte Carlo algorithms as for generic classifiers*.

Our **third contribution** is an extensive benchmark, evaluating our framework RAB on multiple machine learning models and provide the first collection of certified robustness bounds on a diverse range of datasets, namely MNIST, CIFAR-10, ImageNette, as well as spambase tabular data. We hope that these experiments and benchmarks can provide future directions for improving the robustness of ML models against backdoors.

Being the first result on certified robustness against backdoor attacks, we believe that these results can be further improved by future research endeavours inspired by this work. We make the code and evaluation protocol publicly available with the hope to facilitate future research by the community.

Summary of Technical Contributions. Our technical contributions are as follows:

- We propose a unified framework to certify the model robustness against both evasion and backdoor attacks and prove that our robustness bound is tight.
- We provide the first certifiable robustness bound for general machine learning models against backdoor attacks considering *different* smoothing noise distributions.
- We propose an *exact* efficient smoothing algorithm for K-NN models without needing to sample random noise during training.
- We conduct extensive reproducible large-scale experiments and provide a benchmark for certified robustness against three representative backdoor attacks for multiple types of models (e.g., DNNs, support vector machines, and K-NN) on diverse datasets. We also provide a series of ablation studies to further analyze the factors that affect model robustness against backdoors.

Outline. The remainder of this paper is organized as follows. Section 2 provides background on backdoor attacks and related verifiable robustness techniques, followed by the threat model and method overview in Section 3. Section 4 presents the proposed general theoretical framework for certifying robustness against evasion and poisoning attacks, the tightness of the derived robustness bound, and sheds light on a connection between statistical hypothesis testing and certifiable robustness. Section 5 explains in detail the proposed approach RAB for certifying robustness against backdoor attacks under the general framework with Gaussian distributions. Section 6 analyzes the robustness properties of DNNs and K-NN classifiers and presents algorithms to certify robustness for such models (mainly with binary classifiers). Experimental results are presented in section 7.

Finally, Section 8 puts our results in context with existing work, Section 9 discusses the limitations of our work, and Section 10 concludes.

2. Background

In this section, we provide an overview of different backdoor attacks and briefly review the randomized smoothing technique for certifying robustness against evasion attacks.

2.1. Backdoor attacks

A backdoor attack aims to inject certain "backdoor" patterns into the training set and associate such patterns with a specific adversarial target (label). As a result, during testing time, any test instance with such a pattern will be misclassified as the preselected adversarial target [8], [16]. ML models with injected backdoors are called *backdoored models* and they are typically able to achieve performance similar to clean models on benign data, making it challenging to detect whether the model has been backdoored.

There are several ways to categorize backdoor attacks. First, based on the *adversarial target design*, the attacks can be characterized either as *single target attacks* or *all-to-all attacks*. In a *single target attack*, the backdoor pattern will cause the poisoned classifier to always return a designed target label. An *all-to-all* attack leverages the backdoor pattern to permute the classifier results.

The second categorization is based on *different types of backdoor patterns*. There are *region-based* and *blending* backdoor attacks. In the *region-based* attack, a specific region of the training instance is manipulated in a subtle way that will not cause human notification [16], [61]. In particular, it has been shown that such backdoor patterns can be as small as only one or four pixels [48]. On the other hand, Chen et al. [8] shows that by blending the whole instance with a certain pattern such as a fixed random noise pattern, it is possible to generate effective backdoors to poison the ML models.

In this work, we focus on certifying the robustness against general backdoor attacks, where the attacker is able to add any specific or uncontrollable random backdoor patterns for arbitrary adversarial targets.

2.2. Randomized smoothing

To defend against *evasion attacks*, different approaches have been studied: some provide empirical approaches such as adversarial training [4], [30], and some provide theoretical guarantees against L_p bounded adversarial perturbations. In particular, Cohen et al. [10] have proposed *randomized smoothing* to certify the robustness of ML models against the L_2 norm bounded evasion attacks.

On a high level, the randomized smoothing technique [10] provides a way to certify the robustness of a *smoothed* classifier against adversarial examples during test time. First, a given classifier is smoothed by adding Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbb{1}_d)$ around each test instance. Then, the classification gap between a lower bound of the confidence on the top-1 class p_A and an upper bound of the confidence on the top-2 class p_B are obtained. The smoothed classifier will be guaranteed to provide consistent predictions within the perturbation radius, which is a function of the standard deviation σ of the smoothing noise, and the gap between the class probabilities p_A and p_B, for each test instance.

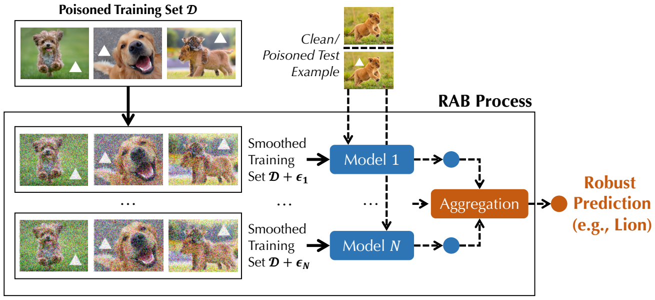

Figure 2: An illustration of the RAB robust training process. Given a poisoned training set $\mathcal{D}+\Delta$ and a training process $\mathcal{A}$ vulnerable to backdoor attacks, RAB generates N smoothed training sets $\{\mathcal{D}_i\}_{i\in[N]}$ and trains N different classifiers $\mathcal{A}_i$.

However, all these approaches focus on the robustness against *evasion attacks* only. In contrast, in this work, we aim to provide a function smoothing framework to certify the robustness against both evasion and poisoning attacks. In particular, the current randomized smoothing strategy focuses on adding noise to induce smoothness on the level of *test instance*, while our unified framework generalizes this to smoothing on the level of *classifiers*. Putting this generalization into practice in the context of certifying robustness against backdoor attacks naturally bears additional challenges which we describe and address in detail. In addition, we provide theoretical robustness guarantees for different machine learning models, smoothing noise distributions, as well as the tightness of the robustness bounds.

3. Threat Model and Method Overview

Here we first define the threat model including concise definitions of a backdoor attack, and then introduce the method overview, where we define our robustness guarantee.

3.1. Notation

We write random variables as uppercase letters X and use the notation $\mathbb{P}_X$ to denote the probability measure induced by X and write f_X to denote the probability density function. Realizations of random variables are written in lowercase letters. For discrete random variables, we use lowercase letters to denote their probability mass function, e.g. $p(y)$ for distribution over labels. Feature vectors are taken to be d-dimensional real vectors $x \in \mathbb{R}^d$ and the set of labels y for a C-multiclass classification problem is given by $\mathcal{C} = \{1, \ldots, C\}$. A training set $\mathcal{D}$ consists of n (feature, label)-pairs $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. For a dataset $\mathcal{D}$ and a collection of n feature vectors $d = \{d_1, \ldots, d_n\}$, we write $\mathcal{D} + d$ to denote the set $\{(x_1 + d_1, y_1), \ldots, (x_n + d_n, y_n)\}$. We view a classifier as a deterministic function that takes as input a tuple with a test instance x and training set $\mathcal{D}$ and returns a class label

$y \in \mathcal{C}$. Formally, given a dataset $\mathcal{D}$ and a test instance x, a classifier h learns a conditional probability distribution $p(y|\,x,\,\mathcal{D})$ over class labels and outputs the label which is deemed most likely under the learned distribution p:

$$h(x,\,\mathcal{D}) = \arg\max_{y} p(y|\,x,\,\mathcal{D}). \qquad (1)$$

We omit the dependence on model parameters throughout this paper and tacitly assume that the model is optimized based on training dataset $\mathcal{D}$ via some optimization schemes such as stochastic gradient descent.

3.2. Threat Model and the Goal of Defense

3.2.1. Threat Model. An adversary carries out a backdoor attack against a classifier h and a clean dataset $\mathcal{D} = \{(x_i, y_i)\}$. The attacker has in mind a target backdoor pattern Ω_x and a target class $\tilde{y}$ and the adversarial goal is to alter the dataset such that, given a clean test example x, adding the backdoor pattern to x (i.e., $x + \Omega_x$) will *alter* the classifier output $\tilde{y}$ with high probability. In general, the attack can replace r training instances $(x_i,\,y_i)$ by backdoored instances $(x_i + \Omega_x,\,\tilde{y}_i)$. We remark that the attacker could embed distinct patterns to each instance and our result naturally extends to this case. Thus, summarizing the backdoor patterns as the collection $\Delta(\Omega_x) := \{\delta_1,\,\ldots,\,\delta_r,\,0,\ldots,0\}$, we formalize a backdoor attack as the transformation $(\mathcal{D}, \Omega_x, \tilde{y}) \to \mathcal{D}_{BD}(\Omega_x, \tilde{y})$ with

$$\mathcal{D}_{BD}(\Omega_x, \tilde{y}) = \{(x_i + \delta_i, \tilde{y}_i)\}_{i=1}^{r} \cup \{(x_i, y_i)\}_{i=r+1}^{n} \qquad (2)$$

We often write $\mathcal{D}_{BD}(\Omega_x)$ instead of $\mathcal{D}_{BD}(\Omega_x, \tilde{y})$ when our focus is on the backdoor pattern Ω_x instead of the target class $\tilde{y}$. The backdoor attack succeeds on test example x whenever

$$h(x + \Omega_x, \mathcal{D}_{BD}(\Omega_x)) = \tilde{y} \qquad (3)$$

3.2.2. Goal of Defense. One natural goal to defend against the above backdoor attack is to ensure that the prediction of $h(x + \Omega_x, \mathcal{D}_{BD}(\Omega_x))$ is *independent* of the backdoor patterns $\Delta(\Omega_x)$ which are present in the dataset, i.e.,

$$h(x + \Omega_x, \mathcal{D}_{BD}(\Omega_x)) = h(x + \Omega_x, \mathcal{D}_{BD}(\emptyset)) \qquad (4)$$

where $\mathcal{D}_{BD}(\emptyset)$ is the dataset without any embedded backdoor patterns ($\delta_i = 0$). When this is true, the attacker obtained *no additional information* by knowing the pattern Ω_x embedded in the training set. That is to say, given a test instance which may contain a backdoor pattern, its prediction stays the same, independent of whether the models were trained with or without backdoors. We assume that the defender has full control of the training process. See Section 9 for more discussions on the assumptions and limitations of RAB.

3.3. Method Overview

3.3.1. Certified Robustness against Backdoor Attacks. We aim to obtain robustness bound R such that, whenever the sum of the magnitude of backdoors is below R, the prediction of the backdoored classifier is the same as when the classifier is trained on benign data. Formally, if $\mathcal{D}_{BD}(\Omega_x)$ denotes the backdoored training set, and $\mathcal{D}$ the training set containing clean features, we say that a classifier is *provably robust* whenever $\sqrt{\sum_{i=1}^{r} \|\delta_i\|_2^2} < R$ implies that $h(x + \Omega_x,\,\mathcal{D}_{BD}(\Omega_x)) = h(x + \Omega_x,\,\mathcal{D}_{BD}(\emptyset))$.

Our approach to obtaining the aforementioned robustness guarantee is based on randomized smoothing, which leads to the robust RAB training pipeline, as is illustrated in Figure 2. Given a clean dataset $\mathcal{D}$ and a backdoored dataset $\mathcal{D}_{BD}(\Omega_x)$, the goal of the defender is to make sure that the prediction on test instances embedded with the pattern Ω_x is the same as for models trained with $\mathcal{D}_{BD}(\emptyset)$.

Different from randomized smoothing-based certification against evasion attacks, here it is not enough to only smooth the test instances. Instead, in RAB, we will first add noise vectors, sampled from a smoothing distribution, to the given training instances, to obtain a collection of "smoothed" training sets. We subsequently train a model on each training set and aggregate their final outputs together as the final "smoothed" prediction. After this process, we show that it is possible to leverage the Neyman Pearson lemma to derive a robustness condition for this smoothed RAB training process. Additionally, the connection with the Neyman Pearson lemma also allows us to prove that the robustness bound is tight. Note that the RAB framework requires the training instances to be "smoothed" by a set of independent noises drawn from a certain distribution.

Additional Challenges. We remark that, within this RAB training and certification process, there are several additional challenges. First, after adding noise to the training data, the clean accuracy of the trained classifier typically drops due to the distribution shift in the training data. To mitigate this problem, we add a deterministic value, based on the hash of the trained model, to test instances (Section 6), which minimizes the distribution shift and leads to improved accuracy scores. Second, considering different smoothing distributions for the training data, we provide rigorous analysis and a robustness bound for Gaussian distributions (Section 5). Third, we note that the proposed training process requires sampling a large number of randomly perturbed training sets. As this is computationally expensive, we propose an efficient PTIME algorithm for K-NN classifiers (Section 6).

Outline. In the following, we illustrate the RAB pipeline in three steps. In Section 4, we introduce the theoretical foundations for a unified framework for certifying robustness against both evasion and backdoor attacks. In Section 5, we introduce how to apply our unified framework to defend against backdoor attacks. In Section 6, we present RAB pipeline for two types of models — DNNs and K-NN.

4. Unified Framework for Certified Robustness

In this section, we propose a unified theoretical framework for certified robustness against evasion and poisoning attacks for classification models. Our framework is based on the intuition that randomizing the prediction or training process will "smoothen" the final prediction and therefore reduce the vulnerability to adversarial attacks. This principle has been successfully applied to certifying robustness against evasion attacks for classification models [10]. We first formally define the notion of a smoothed classifier where we extend upon previous work by randomizing *both* the test instance and the training set. We then introduce basic terminology of hypothesis testing, from where we leverage the Neyman Pearson lemma to derive a generic robustness condition in Theorem 1. Finally, we show that this robustness condition is tight.

4.1. Preliminaries

4.1.1. Smoothed Classifiers. On a high level, a smoothed classifier g is derived from a base classifier h by introducing additive noise to the input consisting of test and training instances. In a nutshell, the intuition behind randomized smoothing classifiers is that noise reduces the occurrence of regions with high curvature in the decision boundaries, resulting in reduced vulnerability to adversarial attacks. Recall that a classifier h, here serving as a base classifier, is defined as $h(x,\, \mathcal{D}) = \arg\max_y p(y|\, x,\, \mathcal{D})$ where p is learned from a dataset $\mathcal{D}$ and defines a conditional probability distribution over labels y. The final prediction is given by the most likely class under this learned distribution. A smoothed classifier is defined by

$$q(y|\, x,\, \mathcal{D}) = \mathbb{P}_{X,D}\left(h(x+X,\, \mathcal{D}+D) = y\right) \tag{5}$$

where we have introduced random variables $X \sim \mathbb{P}_X$ and $D \sim \mathbb{P}_D$ which act as smoothing distributions and are assumed to be independent. We emphasize that D is a collection of n independent and identically distributed random variables $D^{(i)}$, each of which is added to a training instance in $\mathcal{D}$. The final, smoothed classifier then assigns the most likely class to an instance x under this new, "smoothed" model q, so that

$$g(x,\, \mathcal{D}) = \arg\max_y q(y|\, x,\, \mathcal{D}). \tag{6}$$

Within this formulation of a smoothed classifier, we can also model randomized smoothing for defending against evasion attacks by setting the training set noise to be zero, i.e. $D \equiv 0$. We emphasize at this point that the smoothed classifier g implicitly depends on the choice of noise distributions $\mathbb{P}_X$ and $\mathbb{P}_D$. In section 5 we instantiate this classifier with Gaussian noise and with uniform noise and show how this leads to different robustness bounds.

4.1.2. Statistical Hypothesis Testing. Hypothesis testing is a statistical problem that is concerned with the question of whether or not some hypothesis that has been formulated is correct. A decision procedure for such a problem is called a statistical hypothesis test. Formally, the decision is based on the value of a realization x for a random variable X whose distribution is known to be either $\mathbb{P}_0$ (the null hypothesis) or $\mathbb{P}_1$ (the alternative hypothesis). Given a sample $x \in \mathcal{X}$, a randomized test ϕ can be modeled as a function $\phi\colon\, \mathcal{X} \to [0,\, 1]$ which rejects the null hypothesis with probability $\phi(x)$ and accepts it with probability $1-\phi(x)$. The two central quantities of interest are the probabilities of making a type I error, denoted by $\alpha(\phi;\, \mathbb{P}_0)$ and the probability of making a type II error, denoted by $\beta(\phi;\, \mathbb{P}_1)$. The former corresponds to the situation where the test ϕ decides for the alternative when the null is true, while the latter occurs when the alternative is true but the test decides for the null. Formally, α and β are defined as

$$\alpha(\phi;\, \mathbb{P}_0) = \mathbb{E}_0(\phi(X)), \quad \beta(\phi;\, \mathbb{P}_1) = \mathbb{E}_1(1-\phi(X)) \tag{7}$$

where $\mathbb{E}_0(\cdot)$ ($\mathbb{E}_1(\cdot)$) denotes the expected value with respect to $\mathbb{P}_0$ ($\mathbb{P}_1$). The problem is to select the test ϕ which minimizes the probability of making a type II error, subject to the constraint that the probability of making a type-I error is below a given threshold α_0. The Neyman Pearson lemma [35] states that a likelihood ratio test ϕ_{NP} is optimal, i.e. that $\alpha(\phi_{NP};\, \mathbb{P}_0) = \alpha_0$ and $\beta(\phi_{NP};\, \mathbb{P}_1) = \beta^*(\alpha_0;\, \mathbb{P}_0,\, \mathbb{P}_1)$ where

$$\beta^*(\alpha_0;\, \mathbb{P}_0,\, \mathbb{P}_1) = \inf_{\phi\colon\, \alpha(\phi;\, \mathbb{P}_0)\leq\alpha_0} \beta(\phi;\, \mathbb{P}_1). \tag{8}$$

In Theorem 1, we will see that we can leverage this formalism to get a robustness guarantee for smoothed classifiers. Additionally, stemming from the optimality of the likelihood ratio test, we show in Theorem 2 that this condition is tight.

4.2. A General Condition for Provable Robustness

In this section, we derive a tight robustness condition by drawing a connection between statistical hypothesis testing and the robustness of classification models subject to adversarial attacks. We allow adversaries to conduct an attack on either *(i) the test instance* x, *(ii) the training set* $\mathcal{D}$ or *(iii) a combined attack on test and training set*. The resulting robustness condition is of a general nature and is expressed in terms of the optimal type II errors for likelihood ratio tests. We remark that this theorem is a more general version of the result presented in [10], by extending it to general smoothing distributions and smoothing on the training set. In Section 5 we will show how this result can be used to obtain robustness bound in terms of L_p-norm bounded backdoor attacks. We show that smoothing on the training set makes it possible for certifying the robustness against backdoors, and the general smoothing distribution allows us to explore the robustness bound certified by different smoothing distributions.

Theorem 1. *Let q be the smoothed classifier as in* (5) *with smoothing distribution $Z := (X,\, D)$ with X taking values in $\mathbb{R}^d$ and D being a collection of n independent $\mathbb{R}^d$-valued random variables, $D = (D^{(1)},\, \dots,\, D^{(n)})$. Let $\Omega_x \in \mathbb{R}^d$ and let $\Delta := (\delta_1,\, \dots,\, \delta_n)$ for backdoor patterns $\delta_i \in \mathbb{R}^d$. Let $y_A \in \mathcal{C}$ and let $p_A,\, p_B \in [0,\, 1]$ such that $y_A = g(x,\, \mathcal{D})$ and*

$$q(y_A|\, x,\, \mathcal{D}) \geq p_A > p_B \geq \max_{y\neq y_A} q(y|\, x,\, \mathcal{D}). \tag{9}$$

If the optimal type II errors, for testing the null $Z \sim \mathbb{P}_0$ against the alternative $Z + (\Omega_x,\, \Delta) \sim \mathbb{P}_1$, satisfy

$$\beta^*(1-p_A;\, \mathbb{P}_0,\, \mathbb{P}_1) + \beta^*(p_B;\, \mathbb{P}_0,\, \mathbb{P}_1) > 1, \tag{10}$$

then it is guaranteed that $y_A = \arg\max_y q(y|\, x+\Omega_x,\, \mathcal{D}+\Delta)$.

The following is a short sketch of the proof for this theorem. We refer the reader to Appendix A.1 for details.

Proof (Sketch). We first explicitly construct the likelihood ratio tests ϕ_A and ϕ_B for testing the null hypothesis Z against the alternative $Z + (\Omega_x, \Delta)$ with type I errors $\alpha(\phi_A; \mathbb{P}_0) = 1 - p_A$ and $\alpha(\phi_B; \mathbb{P}_0) = p_B$ respectively. An argument similar to the Neyman-Pearson Lemma [35] shows that the class probability for y_A given by q on the perturbed input is lower bounded by $\beta(\phi_A; \mathbb{P}_1) = \beta^*(1-p_A; \mathbb{P}_0, \mathbb{P}_1)$. A similar reasoning leads to the fact that an upper bound on the prediction score for $y \neq y_A$ on the perturbed input is given by $1 - \beta(\phi_B; \mathbb{P}_1) = 1 - \beta^*(p_B; \mathbb{P}_0, \mathbb{P}_1)$. Combining this leads to condition (10). □

We now make some observations about Theorem 1 to get intuition on the robustness condition (10):

- Different smoothing distributions lead to robustness bounds in terms of different norms. For example, Gaussian noise yields robustness bound in L_2 norm while Uniform noise leads to other L_p norms.
- The robustness condition (10) does not make any assumption on the underlying classifier other than on the class probabilities predicted by its smoothed version.
- The random variable $Z + (\Omega_x, \Delta)$ models a general adversarial attack including evasion and backdoor attacks.
- If no attack is present, i.e., if $(\Omega_x, \Delta) = (0, 0)$, then we get the trivial condition $p_A > p_B$.
- As p_A increases, the optimal type II error increases for given backdoor (Ω_x, Δ). Thus, in the simplified setup where $p_A + p_B = 1$ and the robustness condition reads $\beta^*(1 - p_A; \mathbb{P}_0, \mathbb{P}_1) > 1/2$, the distribution shift caused by (Ω_x, Δ) can increase. Thus, as the smoothed classifier becomes more confident, the robust region becomes larger.

While the generality of Theorem 1 allows us to model a multitude of threat models, it bears the challenge of how one should instantiate this theorem such that it is applicable to defend against a specific adversarial attack. In addition to the flexibility with regard to the underlying threat model, we are also provided with flexibility with regard to the smoothing distributions, resulting in different robustness guarantees. This again begs the question, of which smoothing distribution results in useful robustness bounds. In Section 5, we will show how this theorem can be applied to obtain the robustness guarantee against backdoor attacks described in Section 3.

Next, we show that our robustness condition is tight in the following sense: If (9) is all that is known about the smoothed classifier g, then there is no perturbation (Ω_x, Δ) that violates (10). On the other hand, if (10) is violated, then we can always construct a smoothed classifier g^* such that it satisfies the class probabilities (9) but is not robust against this perturbation.

Theorem 2. *Suppose that* $1 \geq p_A + p_B \geq 1 - (C-2) \cdot p_B$. *If the adversarial perturbations* (Ω_x, Δ) *violate (10), then there exists a base classifier* h^* *such that the smoothed classifer* g^* *is consistent with the class probabilities (9) and for which* $g^*(x + \Omega_x, \mathcal{D} + \Delta) \neq y_A$.

5. Provable Robustness Against Backdoors

It is not straightforward to use the result from Theorem 1 to get a robustness certificate against backdoor attacks in terms of L_p-norm bounded backdoor patterns. In this section, we aim to answer the question: *how can we instantiate this result to obtain robustness guarantees against backdoor attacks?* In particular, we show that by leveraging Theorem 1, we obtain the robustness guarantee defined in Section 3. To that end, we derive robustness bounds for smoothing with isotropic Gaussian noise and we also illustrate how to derive certification bounds using other smoothing distributions. Since isotropic Gaussian noise leads to a better radius, we will use this distribution in our experiments as a demonstration.

5.1. Method Outline

5.1.1. Intuition. Suppose that we are given a base classifier that has been trained on a *backdoored* dataset that contains r training samples which are infected with backdoor patterns $\Delta(\Omega_x)$. Our goal is to derive a condition on the backdoor patterns $\Delta(\Omega_x)$ such that the prediction for $x + \Omega_x$ with a classifier trained on the backdoored dataset $\mathcal{D}_{BD}(\Delta(\Omega_x))$ is the same as the prediction (on the same input) that a smoothed classifier would have made, had it been trained on a dataset without the backdoor triggers, $\mathcal{D}_{BD}(\emptyset)$. In other words, we obtain the guarantee that *an attacker can not achieve their goal of systematically leading the test instance with the backdoor pattern to the adversarial target*, meaning they will always obtain the same prediction as long as the added pattern δ satisfies certain conditions (bounded magnitude).

5.1.2. Gaussian Smoothing. We obtain this certificate by instantiating Theorem 1 in the following way. Suppose an attacker injects backdoor patterns $\Delta(\Omega_x) = \{\delta_1, \ldots, \delta_r\} \subset \mathbb{R}^d$ to $r \leq n$ training instances of the training set $\mathcal{D}$, yielding the backdoored training set $\mathcal{D}_{BD}(\Delta(\Omega_x))$. We then train the base classifier on this poisoned dataset, augmented with additional noise on the feature vectors $\mathcal{D}_{BD}(\Delta(\Omega_x)) + D$, where D is the smoothing noise added to the training features. We obtain a prediction of the smoothed classifier g by taking the expectation with respect to the distribution of the smoothing noise D. Suppose that the smoothed classifier obtained in this way predicts a malicious instance $x + \Omega_x$ to be of a certain class with probability at least p_A and the runner-up class with probability at most p_B. Our result tells us that, as long as the introduced patterns satisfy condition (10), we get the guarantee that the malicious test input would have been classified equally as when the classifier had been trained on the dataset with clean features $\mathcal{D}_{BD}(\emptyset)$. In the case where the noise variables are isotropic Gaussians with standard deviation σ, the condition (10) yields a robustness bound in terms of the sum of L_2-norms of the backdoor patterns.

Corollary 1 (Gaussian Smoothing)**.** *Let* $\Delta = (\delta_1, \ldots, \delta_n)$ *and* Ω_x *be* $\mathbb{R}^d$*-valued backdoor patterns and let* $\mathcal{D}$ *be a training set. Suppose that for each* i*, the smoothing noise* *on the training features is $D^{(i)} \overset{iid}{\sim} \mathcal{N}(0,\, \sigma^2 \mathbb{1}_d)$. Let $y_A \in \mathcal{C}$ such that $y_A = g(x + \Omega_x,\, \mathcal{D} + \Delta)$ with class probabilities satisfying*

$$\begin{aligned} q(y_A|\, x + \Omega_x,\, \mathcal{D} + \Delta) &\geq p_A \\ &> p_B \geq \max_{y \neq y_A} q(y|\, x + \Omega_x,\, \mathcal{D} + \Delta). \end{aligned} \tag{11}$$

Then, if the backdoor patterns are bounded by

$$\sqrt{\sum_{i=1}^{n} \|\delta_i\|_2^2} < \frac{\sigma}{2}\left(\Phi^{-1}(p_A) - \Phi^{-1}(p_B)\right), \tag{12}$$

it is guaranteed $y_A = g(x + \Omega_x,\, \mathcal{D}) = g(x + \Omega_x,\, \mathcal{D} + \Delta)$.

This result shows that, whenever the norms of the backdoor patterns are below a certain value, we obtain the guarantee that the classifier makes the same prediction on the test data with backdoors as it does when trained without embedded patterns in the training set. We can further simplify the robustness bound in (12) if we can assume that an attacker poisons at most $r \leq n$ training instances with one single pattern δ. In this case, the bound (12) is given by

$$\|\delta\|_2 < \frac{\sigma}{2\sqrt{r}}\left(\Phi^{-1}(p_A) - \Phi^{-1}(p_B)\right). \tag{13}$$

We thus see that, as we know more about the capabilities of an attacker and the nature of the backdoor patterns, we are able to certify a larger robustness radius, proportional to $1/\sqrt{r}$.

5.2. Other Smoothing Distributions

Given the generality of our framework, it is possible to derive certification bounds using other smoothing distributions. However, different smoothing distributions have vastly different performance and a comparative study among different smoothing distributions is interesting future work. In this paper, we will just illustrate one example of smoothing using a uniform distribution.

Corollary 2 (Uniform Smoothing)**.** *Let $\Delta = (\delta_1,\, \dots,\, \delta_n)$ and Ω_x be $\mathbb{R}^d$ valued backdoor patterns and let $\mathcal{D}$ be a training set. Suppose that for each i, the smoothing noise on the training features is $D^{(i)} \overset{iid}{\sim} \mathcal{U}([a,\, b])$. Let $y_A \in \mathcal{C}$ such that $y_A = g(x + \Omega_x,\, \mathcal{D} + \Delta)$ with class probabilities satisfying*

$$\begin{aligned} q(y_A|\, x + \Omega_x,\, \mathcal{D} + \Delta) &\geq p_A \\ &> p_B \geq \max_{y \neq y_A} q(y|\, x + \Omega_x,\, \mathcal{D} + \Delta). \end{aligned} \tag{14}$$

Then, if the backdoor patterns satisfy

$$1 - \left(\frac{p_A - p_B}{2}\right) < \prod_{i=1}^{n}\left(\prod_{j=1}^{d}\left(1 - \frac{|\delta_{i,j}|}{b-a}\right)_+\right) \tag{15}$$

where $(x)_+ = \max\{x, 0\}$, it is guaranteed that $y_A = g(x + \Omega_x,\, \mathcal{D}) = g(x + \Omega_x,\, \mathcal{D} + \Delta)$.

As in the Gaussian case, the robustness bound in (15) can again be simplified in a similar fashion, if we assume that an attacker poisons at most $r \leq n$ training instances with one single pattern δ.

Discussions. We emphasize that in this paper, we focus on protecting the system against attackers who aim to *trigger* a targeted error with a specific *backdoor pattern*. The system can still be vulnerable to other types of *poisoning attacks*. One such example is the label flipping attack, in which one flips the labels of a subset of examples while keeping the features untouched. Interestingly, one concurrent work explored the possibility of using randomized smoothing to defend against label flipping attack [40]. Developing a single framework to be robust against both backdoor and label flipping attacks is an exciting future direction, and we expect it to require nontrivial extensions of both approaches to achieve non-trivial certified accuracy. Furthermore, while we focus the experiments on Gaussian smoothing and L_2-norm guarantees, it is in principle possible to certify other L_p-norms with different smoothing distributions. For evasion attacks, [29] use exponential smoothing noise with certificates in L_1-norm. Such analysis of different smoothing distributions for different experimental settings goes beyond the scope of this work and is interesting for future research.

6. Instantiating the General Framework with Specific ML Models

In the preceding sections, we presented our approach to certifying robustness against backdoor attacks. Here, we will analyze and provide detailed algorithms for the RAB training pipeline for two types of machine learning models: deep neural networks and K-nearest neighbor classifiers. The success of backdoor poisoning attacks against DNNs has caused a lot of attention recently. Thus, we first aim to evaluate and certify the robustness of DNNs against backdoor attacks. Secondly, given the fact that K-NN models have been widely applied in different applications, either based on raw data or on trained embeddings, it is of great interest to know about the robustness of this type of ML models. Specifically, we are inspired by a recent result [22] and develop an *exact* efficient smoothing algorithm for K-NN models, such that we do not need to draw a large number of random samples from the smoothing distribution for these models. This makes our approach considerably more practical for this type of classifier as it avoids the expensive training of a large number of models, as is required with generic classification algorithms including DNNs.

6.1. Deep Neural Networks

In this section, we consider smoothed models which use DNNs as base classifiers. For a given test input x_{test}, the goal is to calculate the prediction of g on $(x_{test}, \mathcal{D} + \Delta)$ according to Corollary 1 and the corresponding certified bound given in the right hand side of Eq. (12). In the following, we first describe the training process and then the inference algorithm.

Algorithm 1 DNN-RAB for training certifiably robust DNNs.

```
Require: Poisoned training dataset $\mathcal{D} = \{(x_i + \delta_i, \tilde{y}_i)_{i=1}^n\}$, noise scale $\sigma$, model number $N$
for $k = 1, \ldots, N$ do
    Sample $\epsilon_{k,1}, \ldots, \epsilon_{k,n} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2 \mathbb{1}_d)$.
    $\mathcal{D}_k = \{(x_i + \delta_i + \epsilon_{k,i}, \tilde{y}_i)_{i=1}^n\}$.
    $h_k = \texttt{train\_model}(\mathcal{D}_k)$.
    Sample $u_k$ from $\mathcal{N}(0, \sigma^2 \mathbb{1}_d)$ deterministically with random seed based on $hash(h_k)$.
end for
return Model collection $\{(h_1, u_1), \ldots, (h_N, u_N)\}$
```

Algorithm 2 Certified inference with RAB-trained models.

```
Require: Test sample $x$, noise scale $\sigma$, models $\{(h_k, u_k)\}_{k=1}^N$, backdoor magnitude $\|\delta\|_2$, number of poisoned training samples $r$
counts = $|\{k\colon\ h_k(x + u_k, \mathcal{D} + \epsilon_k) = y\}|$ for $y = 1, \ldots, C$
$y_A, y_B$ = top two indices in counts
$n_A, n_B$ = counts[$y_A$], counts[$y_B$]
$p_A, p_B = \texttt{calculate\_bound}(n_A, n_B, N, \alpha)$.
if $p_A > p_B$ then
    $R = \frac{\sigma}{2\sqrt{r}}\left(\Phi^{-1}(p_A) - \Phi^{-1}(p_B)\right)$
    if $R \geq \|\delta\|_2$ then
        return prediction $y_A$, robust radius $R$.
    end if
end if
return ABSTAIN
```

6.1.1. RAB Training for DNNs. First, we draw N samples $d_1, \ldots, d_N$ from the distribution of $D \sim \prod_{i=1}^n \mathcal{N}(0, \sigma^2 \mathbb{1}_d)$. Given the N samples of training noise (each consisting of $|\mathcal{D}| = n$ noise vectors), we train N DNN models on the datasets $\mathcal{D} + d_k$ for $k = 1, \ldots, N$ and obtain classifiers $h_1, \ldots, h_N$. Along with each model h_k, we draw a random noise u_k from $\mathcal{N}(0, \sigma^2 \mathbb{1}_d)$ with a random seed based on the hash of the trained model file. This noise vector is stored along with the model parameters and added to each test input during inference. The reason for this is that, empirically, we observed that inputting test samples without this additional augmentation leads to poor prediction performance since the ensemble of models $\{h_1, \ldots, h_N\}$ has to classify an input that has not been perturbed by Gaussian noise, while it has only "seen" noisy samples, leading to a mismatch between training and test distributions. Algorithm 1 shows the pseudocode describing RAB-training for DNN models.

6.1.2. Inference. To get the prediction of the smoothed classifier on a test sample x_{test} we first compute the empirical majority vote as an unbiased estimate

$$\hat{q}(y|\, x,\, \mathcal{D}) = \frac{|\{k\colon\ h_k(x_{test} + u_k,\, \mathcal{D} + d_k) = y\}|}{N} \tag{16}$$

of the class probabilities and where u_k is the (model-)deterministic noise vector sampled during training in Algorithm 1. Second, for a given error tolerance α, we compute p_A and p_B using one-sided $(1 - \alpha)$ lower confidence intervals for the binomial distribution with parameters n_A and n_B and N samples. Finally, we invoke Corollary 1 and first compute the robust radius according to Eq. (13), based on p_A, p_B the smoothing noise parameter σ and the number of poisoned training samples r. If the resulting radius R is larger than the magnitude of the backdoor samples δ, the prediction is certified, i.e. the backdoor attack has failed on this particular sample. Algorithm 2 shows the pseudocode for the DNN inference with RAB.

6.1.3. Model-deterministic Test-time Augmentation. One caveat in directly applying Equation (16) is the mismatch of the training and test distribution — during training, all examples are perturbed with sampled noise, whereas the test example is without noise. In practice, we see that this mismatch significantly decreases the test accuracy. One natural idea is to also add noise to the test examples, however, this requires careful design (e.g., simply drawing k independent noise vectors and applying them to Equation (16) will lead to a less powerful bound). We thus modify the inference function given a learned model h_k in the following way. Instead of directly classifying an unperturbed input x_{test}, we use the hash value of the trained h_k model parameters as the random seed and sample $u_k \sim \mathcal{N}_{hash(h_k)}(0, \sigma^2 \mathbb{1}_d)$. In practice, we use SHA256 hashing [52] of the trained model file. In this way, the noise we add is a deterministic function of the trained model, which is equivalent to altering the inference function in a deterministic way, $\tilde{h}_k(x_{test}) = h_k(x_{test} + u_k)$. We show in the experiments that this leads to significantly better prediction performance in practice. Note that the reason for using a hash function instead of random sampling every time is to ensure that the noise generation process is deterministic, so the choice of different hash functions is flexible.

6.2. K-Nearest Neighbors

If the base classifier h is a K-nearest neighbor classifier, we can evaluate the corresponding smoothed classifier *exactly* and efficiently, in polynomial time, if the smoothing noise is drawn from a Gaussian distribution. In other words, for this type of model, we can eliminate the need to approximate the expectation value via Monte Carlo sampling and evaluate the classifier exactly. Finally, it is worth remarking that bypassing the need to do Monte Carlo sampling ultimately results in a considerable speed-up as it avoids the expensive training of independent models as is required for generic models including DNNs.

A K-NN classifier works in the following way: Given a training set $\mathcal{D} = \{(x_i, y_i)_{i=1}^n\}$ and a test example x, we first calculate the similarity between x and each x_i, $s_i := \kappa(x_i, x)$ where κ is a similarity function. Given all these similarity scores $\{s_i\}_i$, we choose the K most similar training examples with the largest similarity score $\{x_{\sigma_i}\}_{i=1}^K$ along with corresponding labels $\{y_{\sigma_i}\}_{i=1}^K$. The final prediction is made according to a majority vote among the top-K labels.

Similar to DNNs, we obtain a smoothed K-NN classifier by adding Gaussian noise to training points and evaluate the expectation with respect to this noise distribution

$$q(y|\, x,\, \mathcal{D}) = \mathbb{P}\left(K\text{-NN}(x,\, \mathcal{D} + D) = y\right) \tag{17}$$

where $D = (D^{(1)},\, \ldots,\, D^{(n)}) \sim \prod_{i=1}^{n} \mathcal{N}(0,\, \sigma^2 \mathbb{1}_d)$. The next theorem shows that (17) can be computed exactly and efficiently when we measure the similarity with respect to euclidean distance quantized into finite number similarity of levels.

Theorem 3. *Given n training instances, a C-multiclass K-NN classifier based on quantized euclidean distance with L similarity levels, smoothed with isotropic Gaussian noise can be evaluated exactly with time complexity $\mathcal{O}(K^{2+C} \cdot n^2 \cdot L \cdot C)$.*

Proof (sketch). The first step to computing (17) is to notice that we can summarize all possible arrangements $\{x_{\sigma_i} + D^{(\sigma_i)}\}_{i=1}^{K}$ of top-K instances leading to a prediction by using tally vectors $\gamma \in [K]^C$. A tally vector has as its k-th element the number of instances in the top-K with label k, $\gamma_k = \#\{y_{\sigma_i} = k\}$. In the second step, we partition the event that a tally vector γ occurs into events where an instance i with similarity β is in the top-K but would not be in the top-$(K-1)$. These first two steps result in a summation over $\mathcal{O}(K^C \cdot n \cdot L \cdot C)$ terms. In the last step, we compute the probabilities of the events $\{\text{tally}\, \gamma \wedge \kappa(x_i + D^{(i)},\, x) = \beta\}$ with dynamic programming in $\mathcal{O}(n \cdot K^2)$ steps, resulting in a final time complexity of $\mathcal{O}(K^{2+C} \cdot n^2 \cdot L \cdot C)$. □

If $K = 1$, an efficient algorithm can even achieve time complexity linear in the number of training samples n. We refer the reader to Appendix A.2 for details and the algorithm.

7. Experimental Results

In this section, we present an extensive experimental evaluation of our approach and provide a benchmark for certified robustness for DNN and KNN classifiers on different datasets. In addition, we consider three different types of backdoor attack patterns, namely one-pixel, four-pixel, and blending-based attacks. At a high level, our experiments reveal the following set of observations:

- RAB is able to achieve comparable robustness on benign instances compared with vanilla trained models, and achieves non-trivial *certified accuracy* under a range of realistic backdoor attack settings.
- There is a gap between the certified accuracy provided by RAB and empirical robust accuracy achieved by the state-of-the-art empirical defenses against backdoor attacks without formal guarantees, which serves as the upper bound of the certified accuracy; however, such a gap is reasonably small and we are optimistic that future research can further close this gap.
- RAB's efficient KNN algorithm provides a very effective solution for tabular data.
- Simply applying randomized smoothing to RAB is not effective and careful optimizations (e.g., deterministic test-time augmentation) are necessary.

7.1. Experiment Setup

In this paper, we follow the popular transfer learning setting for poisoning attacks [16], [41], [42], [44], [62] in our experiments, specifically [43]. We first use models initialized with pretrained weights obtained from a clean dataset, and then finetune the model with a subset of training data containing backdoored instances. Preliminary experiments and existing work [51] showed that it is difficult to successfully inject backdoors if only a subset of parameters is finetuned. As a result, we always finetune the entire set of model parameters.

7.1.1. Datasets and Model. We consider four different datasets, namely the MNIST dataset [24] consisting of 60,000 images of handwritten digits from 0-9, the CIFAR-10 dataset [23] which includes 50,000 images of 10 different classes of natural objects such as horse, airplane, automobile, etc. Furthermore, we perform evaluations on the high-resolution ImageNette dataset [19] which is a 10-class subset of the original large-scale ImageNet dataset [11]. Finally, we evaluate the K-NN model on a tabular dataset, namely the UCI Spambase dataset [12], which consists of bag-of-words feature vectors on E-mails and determines whether the message is spam or not. The dataset contains 4,601 data cases, each of which is a 57-dimensional input. We use 0.1% of the MNIST and CIFAR-10 training data to finetune our models; on ImageNette and Spambase, we use 1% for finetuning. For evaluations on DNNs, we choose the CNN architecture from [15] on MNIST and the ResNet used in [10] on CIFAR-10, whereas for ImageNette, we use the standard ResNet-18 [18] architecture.

7.1.2. Training Protocol. We set the number of sampled noise vectors (i.e. augmented datasets) to $N = 1,000$ on MNIST and CIFAR, and $N = 200$ on ImageNette, leading to an ensemble of $1,000$ and 200 models, respectively. The added smoothing noise is sampled from the Gaussian distribution with location parameter $\mu = 0$ and scale $\sigma = 0.5$ for MNIST and Spambase. For CIFAR-10 and ImageNette we use $\mu = 0$ and set the scale to $\sigma = 0.2$. The confidence intervals for the binomial distribution are calculated with an error rate of $\alpha = 0.001$. For the KNN models, we use $K = 3$ neighbors and set the number of similarity levels to $L = 200$, meaning that the similarity scores according to euclidean distance are quantized into 200 distinct levels.

7.1.3. Baselines of Empirical Backdoor Removal Based Defenses. Since this is the first paper providing rigorous certified robustness against backdoor attacks, there is no baseline that allows a comparison of the certified accuracy. We remark that a technical report [49] directly applies the randomized smoothing technique to certify robustness against backdoors without evaluation or analysis. However,

TABLE 1: Evaluation on **DNNs** with different datasets. We use $\sigma = 0.5$ for MNIST and $\sigma = 0.2$ for CIFAR-10 and ImageNette. “Vanilla” denotes DNNs without RAB training and “RAB-cert” is the certified accuracy of RAB. The highest empirical robust accuracies are **bolded**. The robust accuracy scores are evaluated only on *successfully backdoored instances*.

	Backdoor Pattern	Acc. on Benign Instances		Empirical Robust Acc.								**Certified Robust Acc.**
		Vanilla	RAB	Vanilla	RAB	AC [6]	Spectral [48]	Sphere [45]	NC [50]	SCAn [47]	Mixup [3]	RAB-cert
MNIST	One-pixel	92.7%	92.6%	0%	41.2%	64.3%	3.4%	3.1%	**76.2%**	45.6%	34.5%	23.5%
	Four-pixel	92.7%	92.6%	0%	40.7%	56.9%	2.8%	2.1%	**79.9%**	45.4%	33.2%	24.1%
	Blending	92.9%	92.6%	0%	39.6%	**63.6%**	3.0%	1.8%	63.0%	44.7%	28.3%	23.1%
CIFAR-10	One-pixel	59.9%	56.7%	0%	**42.9%**	31.4%	31.2%	16.5%	15.7%	12.9%	26.5%	24.5%
	Four-pixel	59.4%	56.8%	0%	**42.8%**	28.9%	31.4%	15.0%	16.8%	16.5%	31.8%	24.1%
	Blending	60.5%	56.8%	0%	**42.8%**	27.4%	28.0%	16.5%	16.6%	15.8%	30.0%	24.1%
ImageNette	One-pixel	93.0%	91.6%	0%	38.6%	44.7%	47.8%	29.6%	**69.9%**	35.2%	55.1%	15.9%
	Four-pixel	93.7%	91.5%	0%	38.4%	54.2%	52.8%	42.1%	**67.9%**	49.7%	51.6%	12.6%
	Blending	94.8%	91.8%	0%	29.9%	46.3%	18.4%	31.0%	**66.7%**	33.3%	56.3%	9.2%

TABLE 2: Evaluation on **KNNs** with $K = 3$ on the UCI Spambase **tabular dataset**. We use $\sigma = 0.5$ for Spam. “Vanilla” denotes DNNs without RAB training and “RAB-cert” is the certified accuracy of RAB. The highest empirical robust accuracies are **bolded**. The robust accuracy scores are evaluated only on *successfully backdoored instances*.

	Backdoor Pattern	Accuracy on Benign Instances		Empirical Robust Acc.						**Certified Robust Acc.**
		Vanilla	RAB	Vanilla	RAB	AC [6]	Spectral [48]	Sphere [45]	SCAn [47]	RAB-cert
UCI Spambase	One-pixel	98.7%	98.4%	0%	**54.6%**	9.0%	9.6%	2.4%	10.5%	36.4%
	Four-pixel	98.7%	98.4%	0%	**50.0%**	9.6%	9.6%	3.0%	11.2%	33.3%
	Blending	98.7%	98.4%	0%	**58.3%**	8.1%	8.1%	1.7%	9.9%	41.7%

as we will show in Section 7.2.4, directly applying randomized smoothing without deterministic test-time augmentation does not provide high certified robustness.

We will, on the other hand, compare our empirical robust accuracy with the state-of-the-art empirical defenses. We briefly review these defenses in the following.

Activation clustering (AC) [6] extracts the activation of the last hidden layer of a trained model and uses clustering analysis to remove training instances with anomalies. We use the default parameter setting provided in the Adversarial Robustness Toolbox (ART) [36]. **Spectral Signature (Spectral)** [48] uses matrix decomposition on the feature representations to detect and remove training instances with anomalies. We again use the default parameter setting provided in ART. **Sphere** [45] performs dimensionality reduction and removes instances with anomalies in the lower dimensions. The top-15% anomaly instances are removed. **Neural Cleanse (NC)** [50] first reverse-engineers a “pseudo-trigger” for each class. Then, to detect and remove anomaly instances, the distances between each instance with and without the pseudo-trigger are compared, and the most similar ones are recognized as anomaly instances. We use pixel-level distance as the distance metric, 100 epochs for trigger generation, and an initial $\lambda = 0.01$ for MNIST and $\lambda = 0.0001$ for CIFAR and ImageNette. **Statistical Contamination Analyzer (SCAn)** [47] first performs an EM algorithm to decompose two subgroups over a small clean dataset. Then, for each class in the train set, the parameters of a mixture model for all the data are estimated, before we calculate the likelihood for anomaly detection. To identify the backdoored instances, we recognize the smaller set in the most anomalous class as the backdoored instances. **Mixup** [3], following the data augmentation technique in the paper, we use a 4-way mixup training algorithm to train the model over the train set. The convex coefficients are drawn from a Dirichlet distribution with $\alpha = 1.0$.

The initial goal of all these approaches, with the exception of Mixup, is to **detect** backdoored instances, i.e., to determine whether there exists a trigger. To apply them as a defense (i.e., to train a clean model despite the existence of backdoored data), we make adaptations either following the original paper (AC, Spectral, Sphere and NC) or by our design (SCAn) so that we remove training data with anomalies detected by these approaches and retrain a clean model. Some detection cannot be adapted to the defense task, such as [56], and are not included in the comparison.

7.1.4. Evaluation Metrics.

We evaluate the model accuracy trained on the backdoored dataset with vanilla training and RAB training strategies. In particular, we evaluate both the model performance on benign instances (benign accuracy) and backdoored instances for which the attack was successful against the vanilla model (empirical robust accuracy). With RAB, we are also able to calculate the **certified accuracy**, which means that the RAB model not only certifies that the prediction is the same as if it were trained on the clean dataset, but also that the prediction is equal to the ground truth. The certified accuracy is defined below.

$$\text{Certified Acc.} = \frac{1}{n}|\{x_i \colon R_i > \|\delta\|_2 \wedge \hat{y}_i = y_i\}| \qquad (18)$$

where R_i is the robus radius according to Eq. (12), $\hat{y}_i$ is the predicted label, and y_i is the ground truth for input x_i.

We emphasize that we only evaluate the backdoored test instances for which the attack is successful against the vanilla trained models, which is why the vanilla models always have 0% empirical robust accuracy on these backdoored instances in Table 1. This is to evaluate against the effective backdoor attacks and better illustrate the comparison between RAB-trained models with vanilla and baseline backdoor defense models (empirical robust accuracy). Such empirical robust accuracy of different methods serves as an upper bound for the certified accuracy.

7.1.5. Backdoor Patterns. We evaluate RAB against three representative backdoor attacks, namely a one-pixel pattern in the middle of the image, a four-pixel pattern, and blending a random, but fixed, noise pattern to the entire image [8]. We control the perturbation magnitude of the attack via the L_2-norm of the backdoor patterns, setting $\|\delta\|_2 = 0.1$ for all attacks where δ is the backdoor pattern. On MNIST, we inject 10% backdoored instances and 5% for CIFAR and ImageNette respectively. If not described differently, the attack goal is to fool the model into predicting "0" on MNIST, "airplane" on CIFAR and "tench" on ImageNette. In Appendix B.1, we also consider an all-to-all attack goal [15] so that the fooled model will change its prediction conditioned on the original label.

It is possible to use different backdoor patterns via optimization and other approaches. However, since our goal is to provide *certified* robustness against backdoor attacks, a task that is by definition agnostic to the specific backdoor pattern but only depends on the magnitude of the pattern and the number of backdoored training instances, we mainly focus on these representative backdoor patterns. In addition, we only evaluate the backdoor attack to poison the dataset, while other attacks that interfere with the training process are not evaluated [38], as RAB is a robust training pipeline against training data manipulation based poisoning attacks.

7.2. Certified Robustness of DNNs against Backdoor Attacks

In this section we evaluate RAB against backdoor attacks on different models and datasets. We present both the certified robust accuracy of RAB, as well we the empirical robust accuracy comparison between RAB and baseline defenses. Furthermore, we also present several ablation studies to further explore the properties of RAB.

7.2.1. Certified Robustness with RAB. We first evaluate the certified robustness of RAB on DNNs against different backdoor patterns on different datasets. We also present the performance of RAB on benign instances and backdoored instances empirically. Table 1 lists the benchmark results on MNIST, CIFAR-10, and ImageNette, respectively. From the results, we can see that RAB achieves significantly non-trivial certified robust accuracy against backdoor attacks at a negligible cost of benign accuracy; while there are no certified results for any other method. The slight drop in benign accuracy results from training on noisy instances. However, this loss in benign accuracy is less than 3% in most cases and is clearly outweighed by the achieved certified robust accuracy. In particular, RAB achieves over 23% *certified accuracy* on the backdoored instances for MNIST and CIFAR-10, and around 12% for ImageNette. In other words, we can successfully certify for these instances that our model predicts the same result as if it were trained on the clean training set. We run the experiment multiple times and show in Appendix B.7 that the standard deviation is less than 1% in most cases. We also show the abstain rate of certification in Appendix B.6 and observe that it is generally low. If the abstain rate is high, we can perform the similar way as in Cohen et al. [10] to obtain a variation of our theorem to certify the radius by some margin.

7.2.2. Empirical Robustness: without RAB vs. with RAB. In addition to the certificates that RAB can provide, RAB's training process also provides good robustness accuracy *empirically*, without theoretical guarantees.

In Table 1, the "RAB" column reports the empirical robust accuracy — *how often can a malicious input that successfully attacks a vanilla model trick RAB?* We can see that, RAB achieves high empirical robust accuracy, and such empirical robust accuracy achieved by either RAB or other methods serves as an upper bound for the certified robust accuracy provided by RAB under the "RAB-certified" column. It is shown that RAB achieves around 40% empirical robust accuracy on the backdoored instances for MNIST and CIFAR-10, and over 30% for ImageNette. In Appendix B.8, we also try an empirical adversarial attack on the RAB model and observe a similar behavior as on vanilla models.

7.2.3. Comparison with State-of-the-art Empirical Backdoor Defenses. Another line of research is to develop empirical methods to automatically detect and remove backdoored training instances. *How does RAB compare with these methods?* We empirically compare the robustness of RAB with other state-of-the-art baseline methods introduced in Section 7.1.3, as shown in Table 1. we observe that although RAB is not specifically designed for empirical defense, it achieves comparable empirical robust accuracy compared with these baseline methods. RAB outperforms about half of the baselines methods on MNIST and ImageNette and all the baselines on CIFAR-10. Interestingly, our approach performs better on CIFAR-10 than on other tasks while other baselines usually perform badly on CIFAR-10. We attribute this observation to the fact that the benign accuracy on CIFAR-10 is comparably low, so that the baselines based on analyzing feature representations or on model reverse engineering are largely affected and the performance is thus worse. By comparison, RAB only needs to add noise to smooth the training process without analyzing model properties, and is hence less affected by the model viability (similarly, the performance of Mixup is less affected too).

In addition, in Appendix B.1, we additionally evaluate the defenses against a more challenging *all-to-all attack* where many baseline approaches fail, and RAB still achieves good performance. We also show that our approach can be applied to an SVM model for three tabular datasets in Appendix B.4, while existing approaches cannot work well since there is no distinct "activation layer" in a simple SVM model. Furthermore, for very large attack perturbations, the certification will fail as shown in Appendix B.2; however, RAB still achieves non-trivial empirical robustness.

7.2.4. Ablation Study: Impact of Deterministic Test-time Augmentation. We compare the certification accuracy of

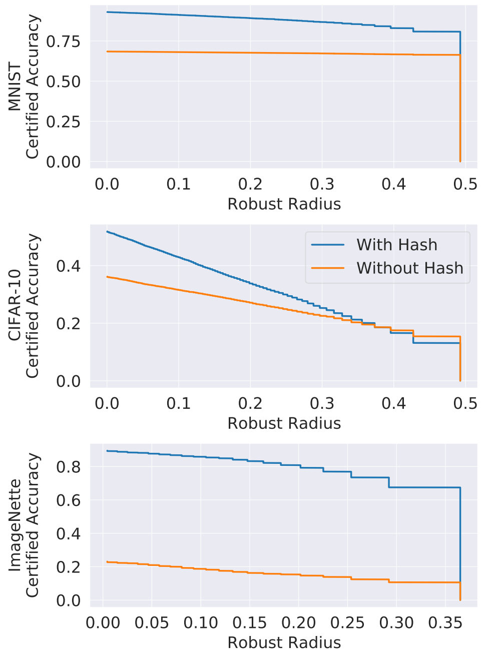

Figure 3: Comparison of the certified accuracy at different radii with and without the proposed deterministic test-time augmentation. The accuracy is evaluated against blending attack with smoothing parameter $\sigma = 0.2$.

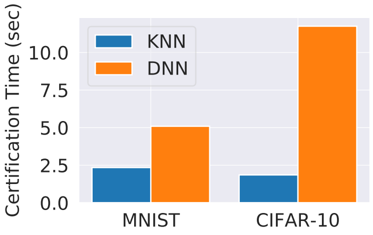

Figure 4: Runtime comparison for certifying one input.

RAB with and without deterministic test-time augmentation in Figure 3. We plot the certified accuracy of all test cases instead of only on successfully attacked cases to show the comparison on the entire dataset. We observe that the certified accuracy significantly improves with the proposed hash function based deterministic test-time augmentation, especially at small certification radii and with a particularly large gap on ImageNette dataset — without the augmentation, the certified accuracy is only around 20%, while it increases to around 80% with the augmentation. This shows that it is important to include the test-time augmentation during inference, and directly adopting randomized smoothing may not provide satisfactory certified accuracy. The detailed empirical and certified robust accuracies are shown in Appendix B.5.

7.3. Certified Robustness of KNN Models

Here we present the benchmarks based on our proposed efficient algorithm for KNN models. We perform experiments on the UCI spambase tabular datasets and show the results for K=3 in Table 2. The NC baseline relies on gradient-based reverse engineering, while Mixup relies on mixing label information during training, so these two methods are not included here. The other baselines use intermediate feature vectors in DNN models, which do not exist in KNN models. Therefore, we use the output prediction vector as the feature vector. From the results, we see that for KNN models, RAB achieves good performance for both empirical and certified robustness and outperforms all the baselines, indicating its advantages for specific domains.

This comparison might seem unfair at first glance, since the considered baselines are based on deep feature representations, which are absent in the KNN case. However, firstly, we emphasize that none of the approaches, including RAB, use deep features for this comparison and have hence access to the same amount of information. Secondly, this comparison reveals an important property of our approach: while the baselines struggle to handle ML models beyond DNN, RAB is applicable to a wider range of models and still yields non-trivial empirical and certified robust accuracy. To enable a comparison for KNN models which is more favorable to the baselines, we consider kernel KNN with a CNN as the kernel function. From the table in Appendix B.3, we see that for this scenario, some baselines indeed outperform RAB.

Figure 4 illustrates the runtime of the exact algorithm for KNN vs. the sampling-based method of DNN. We observe that for certifying one input on KNN with $K = 3$ neighbors, using the proposed *exact* certification algorithm takes only 2.5 seconds, which is around 2-3 times faster than the vanilla RAB on MNIST and 6-7 times faster on CIFAR-10. In addition, the runtime is agnostic to the input size but related to the size of the training set. It would be interesting for future work to design similar efficient certification algorithms for DNNs. Nevertheless, the KNN algorithm is still slower than the algorithm without certification (which is 1000 times faster than the RAB DNN pipeline), and the improvement of running time is still an important future direction.

8. Related Work

In this section, we discuss current backdoor (poisoning) attacks on machine learning models and existing defenses.

Backdoor attacks There have been several works developing optimal poisoning attacks against machine learning models such as SVM and logistic regression [2], [27]. Furthermore, [34] proposes a similar optimization-based poisoning attack against neural networks that can only be applied to shallow MLP models. In addition to these optimization-based poisoning attacks, the backdoor attacks are shown to be very effective against deep neural networks [8], [15]. The backdoor patterns can be either static or generated dynamically [57]. Static backdoor patterns can be as small as one pixel, or as large as an entire image [8].

Empirical defenses against backdoor attacks Given the potentially severe consequences caused by backdoor attacks, multiple defense approaches have been proposed. NeuralCleanse [50] proposes to detect the backdoored models based on the observation that there exists a "short path" to make an instance to be predicted as a malicious one. [7] improves upon the approach by using model inversion to obtain training data, and then applying GANs to generate the "short path" and apply anomaly detection algorithm as in Neural Cleanse. Activation Clustering [6] leverages the activation vectors from the backdoored model as features to detect backdoor instances. Spectral Signature [48] identifies the "spectral signature" in the activation vector for backdoored instances. STRIP [13] proposes to identify the backdoor instances by checking whether the model will still provide a confident answer when it sees the backdoor pattern. SentiNet [9] leverages computer vision techniques to search for the parts in the image that contribute the most to the model output, which are very likely to be the backdoor pattern. In [32], differential privacy has been leveraged as a defense against poisoning attacks. Note that RAB can not guarantee that the trained models are differentially private, although both aim to decrease the model sensitivity intuitively. A further empirical defense against backdoor attacks is proposed in [17] using covariance estimation with the aim of amplifying the spectral signature of backdoored instances.

Certified Defenses against poisoning attacks Another interesting application of randomized smoothing is presented in [40] to certify the robustness against label-flipping attacks and randomize the entire training procedure of the classifier by randomly flipping labels in the training set. This work is orthogonal to ours in that we investigate the robustness with respect to perturbations on the training inputs rather than labels. In a further line of work on provable defenses against poisoning attacks, [26] proposes an ensemble method, deep partition aggregation (DPA). Similar to our work, DPA is related to randomized smoothing, however, in contrast to our work, the goal is to certify the number of poisoned instances for which the prediction remains unaffected. Similarly, [20] use an ensemble technique to certify robustness against poisoning attacks. This is also orthogonal to ours as it certifies the number of poisoned instances, rather than the trigger size. The same certification goal is considered in [21], but is restricted to nearest neighbor algorithms and derives an intrinsic certificate by viewing them as ensemble methods. In addition to these works aiming to certify the robustness of a single model, [60] provides a new way to certify the robustness of an end-to-end sensing-reasoning pipeline. Finally, [54] propose a technique to certify robustness against backdoor attacks within the federated learning framework by controlling the global model smoothness. Furthermore, a technical report also proposes to directly apply the randomized smoothing technique to certify robustness against backdoor attacks without any evaluation or analysis [49]. In addition, as we have shown, directly applying randomized smoothing will not provide high certified robustness bounds. Contrary to that, in this paper, we first provide a unified framework based on randomized smoothing, and then propose the RAB robust training process to provide certified robustness against backdoor attacks based on the framework. We provide the tightness analysis for the robustness bound, analyze different smoothing distributions, and propose the hash function-based model deterministic test-time augmentation approach to achieve good certified robustness. In addition, we analyze different machine learning models with corresponding properties such as model smoothness to provide guidance to further improve the certified robustness.

9. Limitations

One major limitation of RAB is that it introduces non-negligible runtime overhead. To certify the robustness, we need to train and evaluate multiple models (here, 1000 for MNIST/CIFAR-10 and 200 for ImageNette), which is expensive despite the fact that it is parallelizable and can be speeded up with multiple GPUs. Nevertheless, with our polynomial-time KNN algorithm, we have shown a first step towards mitigating the computational cost and leave further endeavors in this direction as future work.

Another limitation is the defender's knowledge of the attack. Indeed, to *certify* the robustness, the defender needs to know 1) an upper bound on the backdoor trigger magnitude (in terms of an L_p norm), 2) an upper bound on the number of poisoned training instances, and, 3) control over the training process. However, to use RAB only as a defense (i.e. without any certificate), the defender only needs to control the training process while 1) and 2) are not needed. The assumption 3) restricts RAB to be a robust training algorithm given an untrusted dataset. In other words, RAB cannot be used to defend against backdoor attacks that interfere with the training process (e.g., [38]).

10. Discussion and Conclusion

In this paper, we aim to propose a unified smoothing framework to certify the model robustness against different attacks. In particular, towards the popular backdoor poisoning attacks, we propose the first robust smoothing pipeline RAB as well as a *model deterministic test-time augmentation* mechanism to certify the prediction robustness against diverse backdoor attacks. In addition, we propose an *exact* algorithm for KNN models without requiring to sample from the smoothing noise distributions. We provide comprehensive benchmarks of certified model robustness against backdoors on diverse datasets, which we believe will provide the *first set* of certified robustness against backdoor attacks for future work to compare with, and hopefully our results and analysis will inspire a new line of research on tighter certified accuracy against backdoor attacks.

Acknowledgement

This work is partially supported by NSF grant No.1910100, NSF CNS 2046726, C3 AI, and the Alfred P. Sloan Foundation. CZ and the DS3Lab gratefully acknowledge the support from the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number MB22.00036 (for European Research Council (ERC) Starting Grant TRIDENT 101042665), the Swiss National Science Foundation (Project Number 200021_184628, and 197485), Innosuisse/SNF BRIDGE Discovery (Project Number 40B2-0_187132), European Union Horizon 2020 Research and Innovation Programme (DAPHNE, 957407), Botnar Research Centre for Child Health, Swiss Data Science Center, Alibaba, Cisco, eBay, Google Focused Research Awards, Kuaishou Inc., Oracle Labs, Zurich Insurance, and the Department of Computer Science at ETH Zurich.

References

[1] A. Athalye, N. Carlini, and D. Wagner, “Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples,” in *International conference on machine learning*. PMLR, 2018, pp. 274–283.

[2] B. Biggio, B. Nelson, and P. Laskov, “Poisoning attacks against support vector machines,” in *Proceedings of the 29th International Coference on Machine Learning*, 2012, p. 1467–1474.

[3] E. Borgnia, V. Cherepanova, L. Fowl, A. Ghiasi, J. Geiping, M. Goldblum, T. Goldstein, and A. Gupta, “Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3855–3859.

[4] X. Cao and N. Z. Gong, “Mitigating evasion attacks to deep neural networks via region-based classification,” in *Proceedings of the 33rd Annual Computer Security Applications Conference*, 2017, pp. 278–287.

[5] N. Carlini and D. Wagner, “Adversarial examples are not easily detected: Bypassing ten detection methods,” in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 2017, pp. 3–14.

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

[7] H. Chen, C. Fu, J. Zhao, and F. Koushanfar, “Deepinspect: a black-box trojan detection and mitigation framework for deep neural networks,” in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 4658–4664.

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

[9] E. Chou, F. Tramer, and G. Pellegrino, “Sentinet: Detecting localized universal attacks against deep learning systems,” in *2020 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2020, pp. 48–54.

[10] J. Cohen, E. Rosenfeld, and Z. Kolter, “Certified adversarial robustness via randomized smoothing,” in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97, 09–15 Jun 2019, pp. 1310–1320.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[12] D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[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 *Proceedings of the 35th Annual Computer Security Applications Conference*. New York, NY, USA: Association for Computing Machinery, 2019, p. 113–125.

[14] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *International Conference on Learning Representations*, 2015.

[15] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, “Badnets: Evaluating backdooring attacks on deep neural networks,” *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[16] T. Gu, B. Dolan-Gavitt, and S. Garg, “Badnets: Identifying vulnerabilities in the machine learning model supply chain,” *arXiv:1708.06733*, 2017.

[17] J. Hayase, W. Kong, R. Somani, and S. Oh, “Spectre: defending against backdoor attacks using robust statistics,” in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 4129–4139. [Online]. Available: https://proceedings.mlr.press/v139/hayase21a.html

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] J. Howard, “Imagenette.” [Online]. Available: https://github.com/fastai/imagenette/

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

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

[22] B. Karlaš, P. Li, R. Wu, N. M. Gürel, X. Chu, W. Wu, and C. Zhang, “Nearest neighbor classifiers over incomplete information: From certain answers to certain predictions,” *Proc. VLDB Endow.*, vol. 14, no. 3, p. 255–267, nov 2020.

[23] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Tech. Rep., 2009.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[25] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, “Certified robustness to adversarial examples with differential privacy,” in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 656–672.

[26] A. Levine and S. Feizi, “Deep partition aggregation: Provable defenses against general poisoning attacks,” in *9th International Conference on Learning Representations*, 2021.

[27] B. Li, Y. Wang, A. Singh, and Y. Vorobeychik, “Data poisoning attacks on factorization-based collaborative filtering,” in *Advances in neural information processing systems*, 2016, pp. 1885–1893.

[28] L. Li, X. Qi, T. Xie, and B. Li, “Sok: Certified robustness for deep neural networks,” *arXiv:2009.04131*, 2020.

[29] L. Li, M. Weber, X. Xu, L. Rimanic, B. Kailkhura, T. Xie, C. Zhang, and B. Li, “Tss: Transformation-specific smoothing for robustness certification,” in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, p. 535–557.

[30] X. Liu, M. Cheng, H. Zhang, and C.-J. Hsieh, “Towards robust neural networks via random self-ensemble,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 369–385.

[31] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, M. E. Houle, D. Song, and J. Bailey, “Characterizing adversarial subspaces using local intrinsic dimensionality,” in *International Conference on Learning Representations*, 2018.

[32] Y. Ma, X. Zhu, and J. Hsu, “Data poisoning against differentially-private learners: Attacks and defenses,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, pp. 4732–4738.

[33] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in *International Conference on Learning Representations*, 2018.

[34] L. Muñoz-González, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli, “Towards poisoning of deep learning algorithms with back-gradient optimization,” in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 2017, pp. 27–38.

[35] J. Neyman and E. Pearson, “On the problem of the most efficient tests of statistical hypotheses. 231,” *Phil. Trans. Roy. Statistical Soc. A*, vol. 289, 1933.

[36] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig *et al.*, “Adversarial robustness toolbox v1. 0.0,” *arXiv:1807.01069*, 2018.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[38] Y. Ren, L. Li, and J. Zhou, "Simtrojan: Stealthy backdoor attack," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 819–823.

[39] J. D. Romano, T. T. Le, W. La Cava, J. T. Gregg, D. J. Goldberg, P. Chakraborty, N. L. Ray, D. Himmelstein, W. Fu, and J. H. Moore, "PMLB v1.0: an open-source dataset collection for benchmarking machine learning methods," *Bioinformatics*, vol. 38, no. 3, pp. 878–880, 2021.

[40] E. Rosenfeld, E. Winston, P. Ravikumar, and Z. Kolter, "Certified robustness to label-flipping attacks via randomized smoothing," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8230–8241.

[41] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 11 957–11 965.

[42] A. Schwarzschild, M. Goldblum, A. Gupta, J. P. Dickerson, and T. Goldstein, "Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 9389–9398.

[43] ——, "Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9389–9398.

[44] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, p. 6106–6116.

[45] J. Steinhardt, P. W. Koh, and P. Liang, "Certified defenses for data poisoning attacks," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 3520–3532.

[46] M. Sun, S. Agarwal, and J. Z. Kolter, "Poisoned classifiers are not only backdoored, they are fundamentally broken," in *arXiv:2010.09080*, 2020.

[47] D. Tang, X. Wang, H. Tang, and K. Zhang, "Demon in the variant: Statistical analysis of {DNNs} for robust backdoor contamination detection," in *30th USENIX Security Symposium*, 2021, pp. 1541–1558.

[48] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *Advances in Neural Information Processing Systems*, 2018, pp. 8000–8010.

[49] B. Wang, X. Cao, N. Z. Gong *et al.*, "On certifying robustness against backdoor attacks via randomized smoothing," *CVPR 2020 Workshop on Adversarial Machine Learning in Computer Vision*, 2020.

[50] 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 *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 707–723.

[51] S. Wang, X. Wang, S. Ye, P. Zhao, and X. Lin, "Defending dnn adversarial attacks with pruning and logits augmentation," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2018, pp. 1144–1148.

[52] Wikipedia contributors, "Sha-2 — Wikipedia, the free encyclopedia," 2020, [Online; accessed 18-March-2020]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=SHA-2&oldid=944705336

[53] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 2018, p. 3905–3911.

[54] C. Xie, M. Chen, P.-Y. Chen, and B. Li, "Crfl: Certifiably robust federated learning against backdoor attacks," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 372–11 382.

[55] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," in *25th Annual Network and Distributed System Security Symposium*. The Internet Society, 2018.

[56] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, "Detecting ai trojans using meta neural analysis," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 103–120.

[57] C. Yang, Q. Wu, H. Li, and Y. Chen, "Generative poisoning attack method against neural networks," *arXiv:1703.01340*, 2017.

[58] G. Yang, T. Duan, J. E. Hu, H. Salman, I. Razenshteyn, and J. Li, "Randomized smoothing of all shapes and sizes," in *International Conference on Machine Learning*. PMLR, 2020, pp. 10 693–10 705.

[59] Z. Yang, B. Li, P.-Y. Chen, and D. Song, "Characterizing audio adversarial examples using temporal dependency," in *International Conference on Learning Representations*, 2019.

[60] Z. Yang, Z. Zhao, H. Pei, B. Wang, B. Karlas, J. Liu, H. Guo, B. Li, and C. Zhang, "End-to-end robustness for sensing-reasoning machine learning pipelines," *arXiv:2003.00120*, 2020.

[61] H. Zhong, C. Liao, A. C. Squicciarini, S. Zhu, and D. Miller, "Backdoor embedding in convolutional neural network models via invisible perturbation," in *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, 2020, pp. 97–108.

[62] C. Zhu, W. R. Huang, H. Li, G. Taylor, C. Studer, and T. Goldstein, "Transferable clean-label poisoning attacks on deep neural nets," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 7614–7623.

Appendix A. Proofs

Here we provide the proofs for the results stated in the main part of the paper. We write $\alpha(\phi) = \alpha(\phi; \mathbb{P}_0)$ and $\beta(\phi) = \beta(\phi; \mathbb{P}_0, \mathbb{P}_1)$ for type-I and -II error probabilities.

A.1. Proof of Theorem 1

Preliminaries and Auxiliary Lemmas: Central to our theoretical results are likelihood ratio tests which are statistical hypothesis tests for testing whether a sample x originates from a distribution X_0 or X_1. These tests are defined as

$$\phi(x) = \begin{cases} 1 & \text{if } \Lambda(x) > t, \\ q & \text{if } \Lambda(x) = t, \\ 0 & \text{if } \Lambda(x) < t. \end{cases} \quad \text{with } \Lambda(x) = \frac{f_{X_1}(x)}{f_{X_0}(x)}, \tag{19}$$

where q and t are chosen such that ϕ has significance α_0, i.e. $\alpha(\phi) = \mathbb{P}_0\left(\Lambda(X) > t\right) + q \cdot \mathbb{P}_0(\Lambda(X) = t) = \alpha_0$.

Lemma A.1. *Let X_0 and X_1 be two random variables with densities f_0 and f_1 with respect to a measure μ and denote by Λ the likelihood ratio $\Lambda(x) = f_1(x)/f_0(x)$. For $p \in [0, 1]$ let $t_p := \inf\{t \geq 0\colon \mathbb{P}(\Lambda(X_0) \leq t) \geq p\}$. Then it holds that*

$$\mathbb{P}\left(\Lambda(X_0) < t_p\right) \leq p \leq \mathbb{P}(\Lambda(X_0) \leq t_p). \tag{20}$$

Proof. We first show the RHS of inequality (20). This follows directly from the definition of t_p if we show that the function $t \mapsto \mathbb{P}(\Lambda(X_0) \leq t)$ is right-continuous. Let $t \geq 0$ and let $\{t_n\}_n$ be a sequence in $\mathbb{R}_{\geq 0}$ such that $t_n \downarrow t$.

Define the sets $A_n := \{x\colon \Lambda(x) \leq t_n\}$ and note that $\mathbb{P}(\Lambda(X_0) \leq t_n) = \mathbb{P}(X_0 \in A_n)$. Clearly, if $x \in \{x\colon \Lambda(x) \leq t\}$ then $\forall n\colon \Lambda(x) \leq t \leq t_n$ and thus $x \in \cap_n A_n$. If on the other hand $x \in \cap_n A_n$ then $\forall n\colon \Lambda(x) \leq t_n \to t$ as $n \to \infty$. Hence, we have that $\cap_n A_n = \{x\colon \Lambda(x) \leq t\}$ and thus $\lim_{n\to\infty} \mathbb{P}(\Lambda(X_0) \leq t_n) = \mathbb{P}(\Lambda(X_0) \leq t)$ since $\lim_{n\to\infty} \mathbb{P}(X_0 \in A_n) = \mathbb{P}(X_0 \in \cap_n A_n)$ for $A_{n+1} \subseteq A_n$. We conclude that $t \mapsto \mathbb{P}(\Lambda(X_0) \leq t)$ is right-continuous and in particular $\mathbb{P}(\Lambda(X_0) \leq t_p) \geq p$. We now show the LHS of inequality (20). For that purpose, consider the set $B_n := \{x\colon \Lambda(x) < t_p - 1/n\}$ and let $B := \{x\colon \Lambda(x) < t_p\}$. Clearly, if $x \in \cup_n B_n$, then $\exists n$ such that $\Lambda(x) < t_p - 1/n < t_p$ and hence $x \in B$. If on the other hand $x \in B$, then we can choose n large enough such that $\Lambda(x) < t_p - 1/n$ and thus $x \in \cup_n B_n$. It follows that $B = \cup_n B_n$. Furthermore, by the definition of t_p and since for any $n \in \mathbb{N}$ we have that $\mathbb{P}(X_0 \in B_n) = \mathbb{P}(\Lambda(X_0) < t_p - 1/n) < p$ it follows that $\mathbb{P}(\Lambda(X_0) < t_p) = \lim_{n\to\infty} \mathbb{P}(X_0 \in B_n) \leq p$ since $B_n \subseteq B_{n+1}$. This concludes the proof. □

Lemma A.2. *Let X_0 and X_1 be random variables taking values in $\mathcal{Z}$ and with probability density functions f_0 and f_1 with respect to a measure μ. Let ϕ^* be a likelihood ratio test for testing the null X_0 against the alternative X_1. Then for any deterministic function $\phi\colon \mathcal{Z} \to [0,1]$ the following implications hold:*

i) $\alpha(\phi) \geq 1 - \alpha(\phi^*) \Rightarrow 1 - \beta(\phi) \geq \beta(\phi^*)$
ii) $\alpha(\phi) \leq \alpha(\phi^*) \Rightarrow \beta(\phi) \geq \beta(\phi^*)$

Proof. We first show (i). Let ϕ^* be a likelihood ratio test as defined in (19). Then, for any other test ϕ we have

$$\begin{aligned}
&1 - \beta(\phi^*) - \beta(\phi) = \\
&= \int_{\Lambda > t} \phi f_1 d\mu + \int_{\Lambda \leq t} (\phi - 1)\, f_1 d\mu + q \int_{\Lambda = t} f_1 d\mu \\
&= \int\limits_{\Lambda > t} \phi \Lambda f_0 d\mu + \int\limits_{\Lambda \leq t} \underbrace{(\phi - 1)}_{\leq 0} \Lambda f_0 d\mu + q \int\limits_{\Lambda = t} \Lambda f_0 d\mu \\
&\geq t \cdot \left[\int\limits_{\Lambda > t} \phi f_0 d\mu + \int\limits_{\Lambda \leq t} (\phi - 1)\, f_0 d\mu + q \int\limits_{\Lambda = t} f_0 d\mu \right] \\
&= t \cdot [\alpha(\phi) - (1 - \alpha(\phi^*))] \geq 0
\end{aligned} \tag{21}$$

with the last inequality following from the assumption and $t \geq 0$. Thus, (i) follows; (ii) can be proved analogously. □

Proof of Theorem 1. We first show the existence of a likelihood ratio test ϕ_A with significance level $1 - p_A$. Let $Z' := (\Omega_x, \Delta) + Z$ and recall that the likelihood ratio Λ between the densities of Z and Z' is given by $\Lambda(z) = \frac{f_{Z'}(z)}{f_Z(z)}$ and let $X' := \Omega_x + X$ and $D' = \Delta + D$. Furthermore, for any $p \in [0,\, 1]$, let $t_p := \inf\{t \geq 0\colon \mathbb{P}(\Lambda(Z) \leq t) \geq p\}$ and

$$q_p = \begin{cases} 0 & \text{if } \mathbb{P}(\Lambda(Z) = t_p) = 0, \\ \frac{\mathbb{P}(\Lambda(Z) \leq t_p) - p}{\mathbb{P}(\Lambda(Z) = t_p)} & \text{otherwise.} \end{cases} \tag{22}$$

Note that by Lemma A.1 we have that $\mathbb{P}(\Lambda(Z) \leq t_p) \geq p$ and

$$\begin{aligned} \mathbb{P}(\Lambda(Z) \leq t_p) &= \mathbb{P}(\Lambda(Z) < t_p) + \mathbb{P}(\Lambda(Z) = t_p) \\ &\leq p + \mathbb{P}(\Lambda(Z) = t_p) \end{aligned} \tag{23}$$

and hence $q_p \in [0,\, 1]$. For $p \in [0,\, 1]$, let ϕ_p be the likelihood ratio test defined in (19) with $q \equiv q_p$ and $t \equiv t_p$. Note that ϕ_p has type-I error probability $\alpha(\phi_p) = 1 - p$. Thus, the test $\phi_A \equiv \phi_{p_A}$ satisfies $\alpha(\phi_A) = 1 - p_A$. It follows from assumption (9) that $\mathbb{P}_{X,\, D}(h(x + X,\, \mathcal{D} + D) = y_A) = q(y_A|\, x,\, \mathcal{D}) \geq 1 - \alpha(\phi_A)$ and thus, by applying the first part of Lemma A.2 to the functions $\phi(z) \equiv \mathbb{1}_{\{h((x,\, \mathcal{D}) + z) = y_A\}}(z)$ and $\phi^* \equiv \phi_A$, it follows that

$$q(y_A|\, x + \Omega_x,\, \mathcal{D} + \Delta) = 1 - \beta(\phi) \geq \beta(\phi_A). \tag{24}$$

Similarly, the likelihood ratio test $\phi_B \equiv \phi_{1-p_B}$ satisfies $\alpha(\phi_B) = p_B$ and, for $y \neq y_A$, it follows from the assumption (9) that $\mathbb{P}_{X,\, D}(h(x + X,\, \mathcal{D} + D) = y) = q(y|\, x,\, \mathcal{D}) \leq p_B = \alpha(\phi_B)$. Thus, applying the second part of Lemma A.2 to the functions $\phi(z) = \mathbb{1}_{\{h((x,\, \mathcal{D}) + z) = y\}}(z)$ and $\phi^* \equiv \phi_B$ yields

$$q(y|\, x + \Omega_x,\, \mathcal{D} + \Delta) = 1 - \beta(\phi) \leq 1 - \beta(\phi_B). \tag{25}$$

Combining (24) and (25) we see that, if $\beta(\phi_A) + \beta(\phi_B) > 1$, then it is guaranteed that $q(y_A|\, x + \Omega_x,\, \mathcal{D} + \Delta) > \max_{y \neq y_A} q(y|\, x + \Omega_x,\, \mathcal{D} + \Delta)$ what completes the proof. □

A.2. Proof of Theorem 2, Corollaries 1, Theorem 3 and Exact KNN Algorithms

We leave the proof of Theorem 2, Corollaries 1, Theorem 3 and exact KNN algorithms to the arXiv version.

Appendix B. Additional Experimental Results

B.1. All-to-all Attacks

In previous evaluations, the attack goal is to fool the model into a specific class. Here, we consider another attack goal that, on seeing the trigger pattern, the model will change its prediction from the i-th class to the $((i+1)\%C)$-th class, where C is the number of classes. Different with the previous goal, the model here will need to recognize both the image and the trigger to make the malicious prediction. Thus, the defenses which assume that the backdoored model makes behavior only based on the backdoor trigger (e.g. NC) will intuitively not have a good performance.

The result of the all-to-all attack is shown in Table B.3. We observe that our approach achieves a similar performance for empirical and certified robustness. The performance on MNIST and ImageNette is slightly better compared with the standard attack, while on CIFAR-10 it decreases a little. As for the baselines, we can observe that the performance of Mixup is also consistent with that on the standard attack. This is understandable as Mixup also performs defense by processing the input and does not rely on model analysis. By comparison, the other baseline approaches based on model analysis does not achieve a good performance here. We owe it to the reason that in all-to-all attacks, the trained model needs to focus on both original image and the trigger pattern, so it is more difficult to detect the backdoors by model analysis than in standard attack where the model only focuses on the trigger pattern.

TABLE B.3: Evaluation on **DNNs** with different datasets with an all-to-all attack goal. We use $\sigma = 0.5$ for MNIST and $\sigma = 0.2$ for CIFAR-10 and ImageNette. "Vanilla" denotes DNNs without RAB training and "RAB-cert" presents certified accuracy of RAB. The highest empirical robust accuracies are **bolded**.

	Backdoor Pattern	Acc. on Benign Instances		Empirical Robust Acc.								Certified Robust Acc.
		Vanilla	RAB	Vanilla	RAB	AC [6]	Spectral [48]	Sphere [45]	NC [50]	SCAn [47]	Mixup [3]	RAB-cert
MNIST	One-pixel	91.5%	90.2%	0%	**51.2%**	17.3%	3.0%	2.8%	28.4%	4.9%	37.1%	24.4%
	Four-pixel	91.6%	91.3%	0%	**60.3%**	16.1%	2.7%	1.8%	30.0%	1.8%	38.7%	39.9%
	Blending	91.5%	91.2%	0%	**59.7%**	15.4%	3.0%	1.8%	30.1%	4.7%	34.6%	39.1%
CIFAR-10	One-pixel	58.4%	52.2%	0%	24.9%	**26.7%**	5.7%	18.2%	13.2%	10.1%	19.7%	10.5%
	Four-pixel	57.5%	52.1%	0%	**25.1%**	11.2%	17.8%	18.3%	17.0%	13.3%	18.7%	11.6%
	Blending	58.3%	52.1%	0%	**24.8%**	10.0%	17.7%	15.9%	12.5%	10.7%	17.0%	10.9%
ImageNette	One-pixel	92.5%	93.0%	0%	43.1%	32.8%	19.6%	41.2%	23.5%	23.5%	**49.2%**	7.8%
	Four-pixel	93.6%	93.0%	0%	37.5%	18.8%	18.8%	43.8%	26.3%	21.7%	**58.3%**	18.7%
	Blending	95.0%	92.9%	0%	44.9%	46.9%	22.9%	34.7%	21.0%	14.3%	**49.0%**	16.3%

TABLE B.4: Evaluation on **DNNs** with different datasets with a large attack perturbation. We use $\sigma = 0.5$ for MNIST and $\sigma = 0.2$ for CIFAR-10 and ImageNette. "Vanilla" denotes DNNs without RAB training and "RAB-cert" presents certified accuracy of RAB. The highest empirical robust accuracies are **bolded**.

	Backdoor Pattern	Acc. on Benign Instances		Empirical Robust Acc.								Certified Robust Acc.
		Vanilla	RAB	Vanilla	RAB	AC [6]	Spectral [48]	Sphere [45]	NC [50]	SCAn [47]	Mixup [3]	RAB-cert
MNIST	Large	86.8%	86.5%	0%	42.3%	65.5%	8.1%	0.6%	**70.9%**	11.9%	20.4%	0%
CIFAR-10	Large	52.1%	44.8%	0%	**27.2%**	20.88%	16.34%	11.96%	25.5%	8.6%	2.4%	0%
ImageNette	Large	84.7%	81.6%	0%	46.4%	62.6%	36.3%	1.5%	**74.9%**	55.5%	59.5%	0%

TABLE B.5: Evaluation on **Kernel KNN** with different datasets. We use $\sigma = 0.5$ for MNIST and $\sigma = 0.2$ for CIFAR-10 and ImageNette. "Vanilla" denotes DNNs without RAB training and "RAB-cert" presents certified accuracy of RAB. The highest empirical robust accuracies are **bolded**.

	Backdoor Pattern	Acc. on Benign Instances		Empirical Robust Acc.								Certified Robust Acc.
		Vanilla	RAB	Vanilla	RAB	AC [6]	Spectral [48]	Sphere [45]	NC [50]	SCAn [47]	Mixup [3]	RAB-cert
MNIST	One-pixel	88.5%	78.2%	0%	35.7%	45.4%	53.0%	48.2%	53.0%	55.8%	**59.5%**	18.0%
	Four-pixel	88.5%	78.1%	0%	36.6%	50.6%	53.6%	48.3%	**69.9%**	55.6%	52.2%	18.8%
	Blending	88.4%	78.4%	0%	36.6%	44.8%	52.4%	47.4%	51.5%	**55.8%**	52.9%	18.8%
CIFAR-10	One-pixel	49.7%	46.5%	0%	21.6%	9.0%	24.9%	15.6%	16.5%	12.9%	**25.1%**	11.3%
	Four-pixel	49.5%	46.6%	0%	21.9%	15.9%	21.7%	**22.7%**	13.4%	15.0%	19.2%	11.7%
	Blending	49.8%	46.6%	0%	20.6%	17.0%	19.6%	15.1%	14.7%	16.8%	**21.8%**	10.5%
ImageNette	One-pixel	90.1%	88.6%	0%	35.3%	**56.8%**	22.2%	28.4%	40.9%	19.3%	31.3%	8.8%
	Four-pixel	90.7%	88.5%	0%	32.0%	**52.2%**	29.6%	41.5%	34.0%	30.8%	27.7%	7.6%
	Blending	91.5%	88.5%	0%	32.1%	**33.3%**	17.2%	2.5%	23.0%	13.8%	21.8%	7.6%

B.2. Larger Perturbation

We consider a larger perturbation consisting of a 4 × 4 trigger pattern with poison rate at 20% and perturbation scale $||\delta_i|| = 4.0$ on MNIST and $||\delta_i|| = 4\sqrt{3}$ on CIFAR-10 and ImageNette (the $\sqrt{3}$ here comes from the fact that we add perturbation on all 3 channels). The results are shown in Table B.4. We can see that such strong perturbation is too large to be within our certification radius, which is a limit of our work. Therefore, the certified robust accuracy is 0. Nevertheless, we can still achieve some non-trivial empirical robustness and is comparable with baselines. This shows that our approach can be applied empirically to defend against strong backdoors with larger perturbation.

B.3. Kernel-KNN

We evaluate the defense on KNNs with a kernel function. The kernel function is learned with the convolution neural network trained on the supervised task and uses the hidden representation of the last layer before output as the kernel output. Note that in this case, our exact KNN certification algorithm cannot be applied since the output with Gaussian variable cannot be analyzed with the kernel function. Therefore, we use the evaluation algorithm as in DNN to evaluate the certification performance. As shown in Table B.5, our approach achieves worse performance than on DNNs, which is understandable since KNN models are known to usually underperform DNN models. On the other hand, we observe that many baselines actually have a better performance than DNN. We view the reason to be that the baselines are based on the detection-and-removal algorithm. We find that the detection will only remove a subset of backdoored instances, so a trained DNN model will still be backdoored; however, any removal of backdoored training data will help the performance of KNN since fewer backdoored instances will be viewed as neighborhood, so the performance may improve. By comparison, RAB will not detect and remove instances and thus will not have a better performance on KNN.

B.4. SVM-based model on tabular data

As our certification for DNN can be applied to any machine learning model, we now evaluate RAB on three tabular data - UCI Spambase dataset (Spambase) [12], and "Adult" and "Agaricus_lepiota" (Mushroom) in the Penn Machine Learning Benchmarks (PMLB) datasets [39]. These datasets are all binary classification tasks. Spambase contains 4,601 data points, with 57-dimensional input; Adult contains 48,842 data points with 14-dimensional input; Mushroom contains 8,145 data points with 22-dimensional

TABLE B.6: Evaluation on **SVM** with different tabular datasets. We use $\sigma = 0.5$ for Spam and $\sigma = 0.2$ for Adult and Mushroom. “Vanilla” denotes DNNs without RAB training and “RAB-cert” presents certified accuracy of RAB. The highest empirical robust accuracies are **bolded**.

	Backdoor Pattern	Acc. on Benign Instances		Empirical Robust Acc.						Certified Robust Acc.
		Vanilla	RAB	Vanilla	RAB	AC [6]	Spectral [48]	Sphere [45]	SCAn [47]	RAB-cert
Spam	One-pixel	91.8%	88.4%	0%	**49.1%**	0%	18.3%	4.8%	12.9%	33.3%
	Four-pixel	91.2%	88.6%	0%	**48.2%**	0%	6.6%	7.4%	11.5%	32.1%
	Blending	92.0%	89.2%	0%	**44.7%**	0%	5.8%	5.8%	11.5%	29.8%
Adult	One-pixel	79.0%	77.2%	0%	**50.7%**	6.3%	15.3%	32.2%	8.4%	17.1%
	Four-pixel	77.4%	73.1%	0%	**53.0%**	5.4%	12.8%	14.4%	7.1%	21.5%
	Blending	78.8%	76.4%	0%	**55.9%**	8.0%	5.0%	11.6%	4.7%	26.1%
Mushroom	One-pixel	87.5%	82.0%	0%	**42.5%**	16.9%	0%	6.4%	17.3%	23.5%
	Four-pixel	86.6%	80.1%	0%	**42.2%**	14.2%	0%	2.8%	13.9%	22.5%
	Blending	87.4%	81.4%	0%	**43.5%**	13.1%	0%	11.1%	14.2%	24.0%

TABLE B.7: Robustness of RAB on **DNNs** with and without test-time augmentation.

	Backdoor Pattern	With Aug		Without Aug	
		RAB	RAB-cert	RAB	RAB-cert
MNIST	One-pixel	41.2%	23.5%	27.0%	12.7%
	Four-pixel	40.7%	24.1%	27.4%	12.8%
	Blending	39.6%	23.1%	26.2%	12.1%
CIFAR-10	One-pixel	42.9%	24.5%	26.9%	15.2%
	Four-pixel	44.4%	25.7%	28.4%	16.4%
	Blending	42.8%	24.1%	27.8%	15.8%
ImageNette	One-pixel	38.6%	15.9%	22.7%	5.1%
	Four-pixel	38.4%	12.6%	22.6%	8.2%
	Blending	29.9%	9.2%	18.7%	4.1%

input. We train a support vector machine (SVM) with RBF kernel using the default setting in scikit-learn toolkit [37]. As for the baselines where activation vectors are required, we use the output prediction vector as its representation, since there are no hidden activation layers in an SVM model.

The result of the SVM dataset is shown in Table B.6. NC is not evaluated because it relies on anomaly detection among different classes, and therefore cannot be applied on these binary classification tasks; Mixup is not evaluated because it cannot be applied in the SVM training algorithm. We can see that our approach still achieves good robustness both empirically and certifiably. Meanwhile, the baseline approaches cannot perform well as they are designed specifically for deep neural networks. In the SVM case where they use the output as the representation vector, the detection performance cannot be good.

B.5. With & Without Test-time Augmentation

Table B.7 shows the comparison of empirical and certified robustness with and without test-time augmentation. We see that the test-time augmentation indeed helps with the model robustness both empirically and certifiably.

B.6. Abstain Rate

Table B.8 shows the abstain rate of RAB against attacks. We see that in general, the abstain rate is relatively low and will not be a serious concern in the pipeline. Note that if the denial-of-service attack is indeed a concern, we can perform a similar way as in [10] to prove a certified radius in which we can certify our defense rather than abstaining the input.

TABLE B.8: The abstain rate of the certification on **DNNs**.

	Backdoor Pattern	Abstain Rate
MNIST	One-pixel	3.32%
	Four-pixel	3.21%
	Blending	3.02%
CIFAR-10	One-pixel	5.59%
	Four-pixel	6.00%
	Blending	5.29%
ImageNette	One-pixel	3.89%
	Four-pixel	4.08%
	Blending	1.90%

TABLE B.9: The mean and standard deviation of the RAB robustness on DNNs with 5 runs.

	Backdoor Pattern	RAB	RAB-cert
MNIST	One-pixel	40.79±0.72%	23.36±0.52%
	Four-pixel	40.27±0.87%	24.37±0.49%
	Blending	40.72±0.65%	23.58±0.88%
CIFAR-10	One-pixel	42.66±0.29%	24.35±0.31%
	Four-pixel	42.56±0.32%	25.25±0.37%
	Blending	42.89±0.21%	23.95±0.17%
ImageNette	One-pixel	38.64±0.80%	15.45±0.94%
	Four-pixel	37.23±0.69%	12.45±0.82%
	Blending	28.74±1.15%	9.20±1.40%

B.7. Multiple Runs

To see the stability of RAB, we run our algorithm 5 times and report the mean and standard deviation in Table B.9. We can see that the standard deviation is relatively small, indicating that our algorithm is stable.

B.8. Adversarial Atacks on RAB Models

In [46], the authors show that if they smooth a backdoored model, the robustified version will still be broken (i.e. with obvious adversarial pattern). We replicate the experiments on the RAB model by performing adversarial attacks against RAB model. In order to do attack, we use the PGD attack where the gradient is calculated by aggregating the gradient on all the trained models. In Figure 5, We show the results on ImageNette with $\varepsilon = 60$ so that the pattern is the most clear. We observe that the adversarial examples look similar with those of unsmoothed model in [46]. Thus, the RAB pipeline is different with the smoothing process; rather, it is similar with an unsmoothed vanilla model.

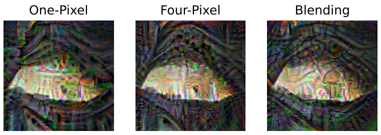

Figure 5: Adversarial examples against backdoored RAB model on the ImageNetted dataset.