

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 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, 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. We derive the robustness conditions for different smoothing distributions including Gaussian and uniform distributions. 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 which eliminates the need to sample from a noise distribution for such models. Empirically, we conduct comprehensive experiments for different machine learning models such as DNNs 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 lights 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 [15, 47, 53, 49].

In response, several empirical solutions have been proposed as defenses against evasion attacks [5, 48, 30, 51]. For instance, adversarial training has been proposed to retrain the ML models with generated adversarial examples [32]; quantization has been applied to either inputs or neural network weights to defend against potential adversarial instances [48]. However, recent studies have shown that these defenses are not resilient against intelligent adversaries responding dynamically to the deployed defenses [5, 2].

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 [27]. 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 [24, 10, 50]. Intuitively,

*The first two authors contribute equally to this work.

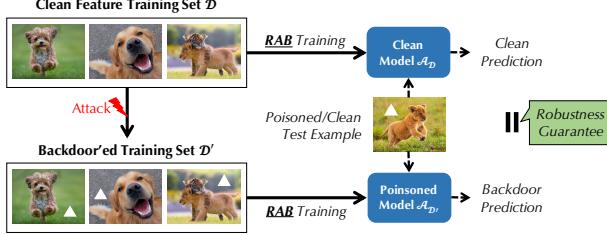


Figure 1: In this paper, we define a robust training process RAB against backdoor attacks. Given a poisoned dataset \mathcal{D}' — produced by adding backdoor patterns Δ to some instances in the dataset \mathcal{D} with clean features — this robust training process guarantees that, for all test examples 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.

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* [45, 14, 17, 26], 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 training-time attack is non-trivial as illustrated in a concurrent work [36], 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 towards test images with the same patterns [16, 8]. Such attacks can be applied to various real-world scenarios such as online face recognition systems [26, 8]. 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 achieve 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 a training set \mathcal{D} , RAB generates N additional “smoothed” training sets $\mathcal{D} + \epsilon_i$ by adding noise ϵ_i ($i \in \{1, \dots, N\}$) 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 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, differentially private DNNs, 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.

Limitations and moving forward. In this paper, we take the first step towards what we believe to be an important, timely problem, namely *certifiable robustness against backdoor attacks*. As an early endeavor on this challenging problem, we believe that it is important for us to make the limitations of the proposed RAB clear — we hope that this paper can provide inspirations for future research to continue improving upon our methods. First, while RAB provides an efficient PTIME algorithm for K nearest neighbors (KNN) classifiers, it still requires the expensive training of multiple models for generic DNN classifiers. Although these training jobs are parallelizable, it is an interesting future direction on how to further decrease the computational complexity for DNN classifiers, or develop efficient PTIME algorithms for other classification methods beyond KNN. Second, RAB focuses on certifying backdoor triggers with limited L_p bounded magnitude. It is interesting to extend RAB to other threat models considering triggers beyond L_p , potentially following how [28] extends randomized smoothing to different semantic transformations. Third, RAB only focuses on *certifiable robustness*, and it is interesting to understand how it can be combined with another orthogonal line of research focusing on empirically detecting backdoors [6, 12, 38]. We believe that such a combination can bring further benefits to both lines of work.

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 and uniform noise distributions. Section 6 analyzes 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 and Section 9 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 [17, 8]. 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 [17, 53]. In particular, it has been shown that such backdoor patterns can be as small as only one or four pixels [43]. On the other hand, Chen et al. [8] show 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 [29, 4], 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.

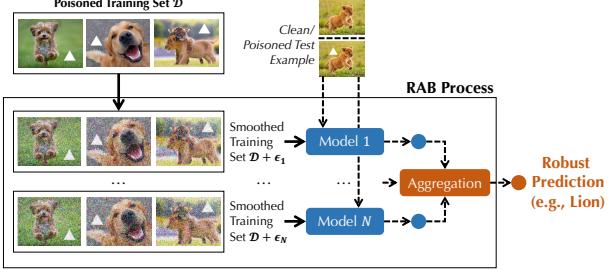


Figure 2: An illustration of the RAB robust training process. Given a poisoned training set $\mathcal{D} + \Delta$ and a training process \mathcal{A} vulnerable to backdoor attacks, RAB generates N smoothed training sets $\{\mathcal{D}_i\}_{i \in [N]}$ and trains N different classifiers \mathcal{A}_i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

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

In this section, 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 a 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 $\mathcal{C} = \{1, \dots, C\}$. A training set \mathcal{D} consists of n (feature, label)-pairs $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$. For a dataset \mathcal{D} and a collection of n feature vectors $d = \{d_1, \dots, d_n\}$, we write $\mathcal{D} + d$ to denote the set $\{(x_1 + d_1, y_1), \dots, (x_n + d_n, y_n)\}$. We view a classifier as a deterministic function that takes as input a tuple with a test instance x and a 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 :

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

We omit the dependence on model parameters throughout this paper and tacitly assume that the learned model is based on parameters obtained from training on the 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 $\Delta(\Omega_x) := \{\delta_1, \dots, \delta_r, 0, \dots, 0\}$, we formalize a backdoor attack as the transformation $(\mathcal{D}, \Omega_x, \tilde{y}) \rightarrow \mathcal{D}_{BD}(\Omega_x, \tilde{y})$ with

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

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

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

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 the backdoor patterns $\Delta(\Omega_x)$ which are present in the dataset, i.e.,

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

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.

3.3 Method Overview

3.3.1 Certified Robustness against Backdoor Attacks

We aim to obtain a 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

$$\sqrt{\sum_{i=1}^r \|\delta_i\|_2^2} < R \quad (5)$$

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 within the RAB framework, it 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 a rigorous analysis and a robustness bound for both Gaussian and uniform smoothing distributions (Section 5). Third, we note that the proposed training process requires to sample 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 the RAB pipeline for two types of models — DNNs and K -NN models.

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

$$q(y|x, \mathcal{D}) = \mathbb{P}_{X,D}(h(x+X, \mathcal{D}+D) = y) \quad (6)$$

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

$$g(x, \mathcal{D}) = \arg \max_y q(y|x, \mathcal{D}). \quad (7)$$

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 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 $\phi: \mathcal{X} \rightarrow [0, 1]$ 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

$$\alpha(\phi; \mathbb{P}_0) = \mathbb{E}_0(\phi(X)), \quad \beta(\phi; \mathbb{P}_1) = \mathbb{E}_1(1 - \phi(X)) \quad (8)$$

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 [34] 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

$$\beta^*(\alpha_0; \mathbb{P}_0, \mathbb{P}_1) = \inf_{\phi: \alpha(\phi; \mathbb{P}_0) \leq \alpha_0} \beta(\phi; \mathbb{P}_1). \quad (9)$$

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 a 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 (6) 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

$$q(y_A | x, \mathcal{D}) \geq p_A > p_B \geq \max_{y \neq y_A} q(y | x, \mathcal{D}). \quad (10)$$

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

$$\beta^*(1 - p_A; \mathbb{P}_0, \mathbb{P}_1) + \beta^*(p_B; \mathbb{P}_0, \mathbb{P}_1) > 1, \quad (11)$$

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 [34] 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 (11). \square

We now make some observations about Theorem 1 to get intuition on the robustness condition (11):

- Different smoothing distributions lead to robustness bounds in terms of different norms. For example, Gaussian noise yields a robustness bound in L_2 norm while Uniform noise leads to other L_p norms.
- The robustness condition (11) 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, 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.

The next proposition shows that it is a necessary condition that Z and $Z + (\Omega_x, \Delta)$ have non-disjoint support in order for Eq. (11) to hold. This sheds light on the dynamics governing the robustness condition (11): if the smoothing distribution Z has compact support, then it is impossible to certify perturbations (Ω_x, Δ) , which induce a shift beyond the support of Z . This result provides us with an idea on the maximal possible robustness bound.

Proposition 1. *If Z and $Z' := (\Omega_x, \Delta) + Z$ have disjoint support, then condition (11) cannot be satisfied.*

Proof. If Z and Z' have disjoint supports, then the null hypothesis Z can be distinguished from the alternative Z' by a single observation and with zero probability of error, since $f_Z(z) = 0 \iff f_{Z'}(z) \neq 0$ for any $z \in \text{supp}(Z) \cup \text{supp}(Z')$. The hypotheses can thus be distinguished in a deterministic way and hence any likelihood ratio test has type II error $\beta(\phi; \mathbb{P}_1) = 1 - \mathbb{E}(\phi(Z')) = 0$. This leads to the contradiction $0 = \beta^*(1 - p_A; \mathbb{P}_0, \mathbb{P}_1) + \beta^*(p_B; \mathbb{P}_0, \mathbb{P}_1) > 1$. \square

For example, suppose $Z = (X, 0)$ with $X \sim \mathcal{U}[0, 1]$. As a consequence of Proposition 1 we immediately see that it is not possible to certify input perturbations Ω_x with absolute value larger than 1, because otherwise Z and $\Omega_x + Z$ would have disjoint supports. Next, we show that our robustness condition is tight in the following sense: If (10) is all that is known about the smoothed classifier g , then there is no perturbation

(Ω_x, Δ) that violates (11). On the other hand, if (11) is violated, then we can always construct a smoothed classifier g^* such that it satisfies the class probabilities (10) 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 (11), then there exists a base classifier h^* such that the smoothed classifier g^* is consistent with the class probabilities (10) 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 which 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 $\Delta(\Omega_x) = \{\delta_1, \dots, \delta_r\} \subset \mathbb{R}^d$ 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 (11), 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 (11) yields a robustness bound in terms of the sum of L_2 -norms of the backdoor patterns.

Corollary 1 (Gaussian Smoothing). Let $\Delta = (\delta_1, \dots, \delta_n)$ 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 $D^{(i)} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2 \mathbb{1}_d)$. Let $y_A \in \mathcal{C}$ such that $y_A = g(x + \Omega_x, \mathcal{D} + \Delta)$ with class probabilities satisfying

$$q(y_A | x + \Omega_x, \mathcal{D} + \Delta) \geq p_A > p_B \geq \max_{y \neq y_A} q(y | x + \Omega_x, \mathcal{D} + \Delta). \quad (12)$$

Then, if the backdoor patterns are bounded by

$$\sqrt{\sum_{i=1}^n \|\delta_i\|_2^2} < \frac{\sigma}{2} (\Phi^{-1}(p_A) - \Phi^{-1}(p_B)), \quad (13)$$

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

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 (13) 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 (13) is given by

$$\|\delta\|_2 < \frac{\sigma}{2\sqrt{r}} (\Phi^{-1}(p_A) - \Phi^{-1}(p_B)). \quad (14)$$

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 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 $D^{(i)} \stackrel{iid}{\sim} \mathcal{U}([a, b])$. Let $y_A \in \mathcal{C}$ such that $y_A = g(x + \Omega_x, \mathcal{D} + \Delta)$ with class probabilities satisfying*

$$q(y_A | x + \Omega_x, \mathcal{D} + \Delta) \geq p_A > p_B \geq \max_{y \neq y_A} q(y | x + \Omega_x, \mathcal{D} + \Delta). \quad (15)$$

Then, if the backdoor patterns satisfy

$$1 - \left(\frac{p_A - p_B}{2} \right) < \prod_{i=1}^n \left(\prod_{j=1}^d \left(1 - \frac{|\delta_{i,j}|}{b-a} \right)_+ \right) \quad (16)$$

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

Similar to the Gaussian case, we can simplify the robustness bound in (16), if we assume that an attacker poisons at most $r \leq n$ training instances with one single pattern δ . In this case, the bound (16) is given by

$$1 - \left(\frac{p_A - p_B}{2} \right) < \left(\prod_{j=1}^d \left(1 - \frac{|\delta_j|}{b-a} \right)_+ \right)^r. \quad (17)$$

We see again that, as the number of infected training samples r gets smaller, this corresponds to a larger bound since the RHS of (17) gets larger. In other words if we know that the attacker injects fewer backdoors, then we can certify a backdoor pattern with larger magnitude.

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

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 σ , model number N

- 1: **for** $k = 1, \dots, N$ **do**
- 2: Sample $\epsilon_{k,1}, \dots, \epsilon_{k,n} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2 \mathbb{1}_d)$.
- 3: $\mathcal{D}_k = \{(x_i + \delta_i + \epsilon_{k,i}, \tilde{y}_i)\}_{i=1}^n\}$.
- 4: $h_k = \text{train_model}(\mathcal{D}_k)$.
- 5: Sample u_k from $\mathcal{N}(0, \sigma^2 \mathbb{1}_d)$ deterministically with random seed based on $\text{hash}(h_k)$.
- 6: **end for**
- 7: **return** Model collection $\{(h_1, u_1), \dots, (h_N, u_N)\}$

Algorithm 2 Certified inference with RAB-trained models.

Require: Test sample x , noise scale σ , models $\{(h_k, u_k)\}_{k=1}^N$, backdoor magnitude $\|\delta\|_2$, number of poisoned training samples r

- 1: $\text{counts} = |\{k: h_k(x + u_k, \mathcal{D} + \epsilon_k) = y\}|$ for $y = 1, \dots, C$
- 2: $y_A, y_B = \text{top two indices in counts}$
- 3: $n_A, n_B = \text{counts}[y_A], \text{counts}[y_B]$
- 4: $p_A, p_B = \text{calculate_bound}(n_A, n_B, N, \alpha)$.
- 5: **if** $p_A > p_B$ **then**
- 6: $R = \frac{\sigma}{2\sqrt{r}} (\Phi^{-1}(p_A) - \Phi^{-1}(p_B))$
- 7: **if** $R \geq \|\delta\|_2$ **then**
- 8: **return** prediction y_A , robust radius R .
- 9: **end if**
- 10: **end if**
- 11: **return** ABSTAIN

examples while keeping the features untouched. Interestingly, one concurrent work explored the possibility of using randomized smoothing to defend against label flipping attack [36]. 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.

6 Instantiating the General Framework with Specific ML Models

In the preceding sections, we presented our approach to certify 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 [21] 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. (13). In the following, we first describe the training process and then the inference algorithm.

6.1.1 RAB Training for DNNs

First, we draw N samples d_1, \dots, 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 $k = 1, \dots, N$ and obtain classifiers h_1, \dots, 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 $\{h_1, \dots, h_N\}$ has to classify an input which 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

$$\hat{q}(y|x, \mathcal{D}) = \frac{n_y}{N}, \quad n_y = |\{k: h_k(x_{test} + u_k, \mathcal{D} + d_k) = y\}| \quad (18)$$

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. (14), 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 (18) 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 apply them to Equation (18) 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[46] 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.

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

$$q(y|x, \mathcal{D}) = \mathbb{P}(K\text{-NN}(x, \mathcal{D} + D) = y) \quad (19)$$

where $D = (D^{(1)}, \dots, D^{(n)}) \sim \prod_{i=1}^n \mathcal{N}(0, \sigma^2 \mathbf{1}_d)$. The next theorem shows that (19) 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 (19) 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 $\mathcal{O}(K^{2+C} \cdot n^2 \cdot L \cdot C)$. \square

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 B 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. The attack patterns are illustrated in Figure 3 which shows that these patterns can be hard to spot by a human, in particular for the one-pixel pattern on high resolution images. 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. On the other hand, as expected, its performance on raw image features such as MNIST and CIFAR-10 is worse than DNNs.
- Simply applying randomized smoothing to RAB is not effective and careful optimizations (e.g., deterministic test-time augmentation) are necessary.

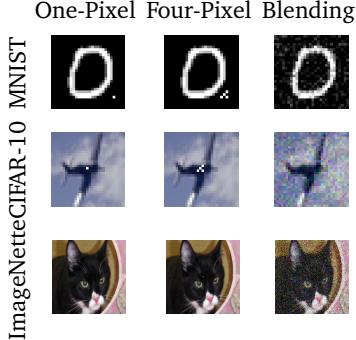


Figure 3: Examples of the applied backdoor patterns.

7.1 Experiment Setup

In this paper, we follow the popular transfer learning setting for poisoning attacks [41, 54, 37, 17, 39] in our experiments, specifically [40]. We first use models initialized with pretrained weights, and then finetune the model with a subset of training data containing backdoored instances.

7.1.1 Datasets and Model

We consider four different datasets, namely the MNIST dataset [23] consisting of 60,000 images of handwritten digits from 0-9, the CIFAR-10 dataset[22] 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 [20] 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 [13], 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 [16] on MNIST and the ResNet used in [10] on CIFAR-10, whereas for ImageNette, we use the standard ResNet-18 [19] 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 Defenses

We compare our approach with three state-of-the-art backdoor defense methods which focus on the empirical defense: Activation clustering (**AC**) [6], Spectral Signature (**Spectral**) [43] and **Sphere** defense [42]. AC extracts the activation of the last hidden layer of a trained model, and uses clustering analysis to removes training instances with anomalies. Spectral leverages matrix decomposition on the feature representations to detect and remove training instances with anomalies. Finally, Sphere performs dimension reduction and removes instances with anomalies in the lower dimensions. We use the Adversarial Robustness Toolbox (ART) [35] for AC and Spectral and implement the Sphere defense in our codebase.

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. “Robust Accuracy” of RAB shows the empirical robust accuracy, and “RAB-certified” presents certified accuracy of RAB. The highest empirical robust accuracies are **bolded**.

	Backdoor Pattern	Accuracy on Benign Instances		Robust Accuracy on Successful Backdoored Instances					
		Vanilla	RAB	Vanilla	RAB	RAB-certified	AC [6]	Spectral [43]	Sphere [42]
MNIST	One-pixel	92.7%	92.6%	0%	41.2%	23.5%	64.3%	3.4%	3.1%
	Four-pixel	92.7%	92.6%	0%	40.7%	24.1%	56.9%	2.8%	2.1%
	Blending	92.9%	92.6%	0%	39.6%	23.1%	63.6%	3.0%	1.8%
CIFAR-10	One-pixel	59.9%	56.7%	0%	42.9%	24.5%	31.4%	31.2%	16.5%
	Four-pixel	59.4%	56.8%	0%	42.8%	24.1%	28.9%	31.4%	15.0%
	Blending	60.5%	56.8%	0%	42.8%	24.1%	27.4%	28.0%	16.5%
ImageNette	One-pixel	93.0%	91.6%	0%	38.6%	15.9%	44.7%	47.8%	29.6%
	Four-pixel	93.7%	91.5%	0%	38.4%	12.6%	54.2%	52.8%	42.1%
	Blending	94.8%	91.8%	0%	29.9%	9.2%	46.3%	18.4%	31.0%

Table 2: Evaluation on KNNs with $K = 3$ on the **tabular dataset** UCI Spambase. We use $\sigma = 0.5$. “Vanilla” denotes KNN without RAB training. “Robust Accuracy” of RAB shows the empirical robust accuracy, and “RAB-certified” presents certified accuracy of RAB. The highest empirical robust accuracies are **bolded**.

	Backdoor Pattern	Accuracy on Benign Instances		Robust Accuracy on Successful Backdoored Instances					
		Vanilla	RAB	Vanilla	RAB	RAB-certified	AC [6]	Spectral [43]	Sphere [42]
Spambase	One-pixel	98.7%	98.4%	0%	54.6%	36.4%	9.0%	9.6%	2.4%
	Four-pixel	98.7%	98.4%	0%	50.0%	33.3%	9.6%	9.6%	3.0%
	Blending	98.7%	98.4%	0%	58.3%	41.7%	8.1%	8.1%	1.7%

Table 3: Evaluation for KNNs with $K = 3$ on **image datasets**, which we expect to perform worse than the DNN results in Table 1. We use $\sigma = 0.5$ for MNIST and $\sigma = 0.2$ for CIFAR-10. “Vanilla” denotes KNN without RAB training. “Robust Accuracy” of RAB is the empirical robust accuracy, and “RAB-certified” is the certified accuracy of RAB. The highest empirical robust accuracies are **bolded**.

	Backdoor Pattern	Accuracy on Benign Instances		Robust Accuracy on Successful Backdoored Instances					
		Vanilla	RAB	Vanilla	RAB	RAB-certified	AC [6]	Spectral [43]	Sphere [42]
MNIST	One-pixel	83.2%	77.3%	0%	7.9%	0.8%	5.7%	5.7%	1.9%
	Four-pixel	83.2%	77.3%	0%	7.9%	0.8%	5.6%	5.6%	1.9%
	Blending	83.2%	77.3%	0%	8.0%	1.6%	5.8%	5.8%	1.8%
CIFAR-10	One-pixel	22.0%	18.8%	0%	4.9%	1.8%	0.1%	0.1%	0.1%
	Four-pixel	22.0%	18.8%	0%	4.9%	1.7%	0.1%	0.1%	0.1%
	Blending	22.0%	18.8%	0%	4.8%	1.6%	0.1%	0.1%	0.1%

Since this is the first paper providing rigorous certified robustness against backdoor attacks, there is no other baseline for comparing the certified accuracy. Note that a technical report [44] directly applies the randomized smoothing technique to certify robustness against backdoors without any evaluation or analysis, and we will show in Section 7.2.5 that directly apply randomized smoothing without deterministic test-time augmentation will not provide high certified robustness.

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.

$$\text{Certified Acc.} = \frac{1}{n} |\{x_i : R_i > \|\delta\|_2 \wedge \hat{y}_i = y_i\}| \quad (20)$$

where R_i is the robust radius according to Eq. (13), \hat{y}_i is the predicted label, and y_i is the ground truth for input x_i . We emphasize that we only evaluate on 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-Table 3. 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 visualize all backdoor patterns on different datasets in Fig. 3. We control the L_2 -norm of the backdoor patterns as the perturbation magnitude. In particular, we select the backdoor magnitude to $\|\delta\|_2 = 0.1$ for all attacks. It is possible to inject different backdoor patterns via optimization and other approaches as well. However, since our goal is to provide the *certified* robustness against backdoor attacks, which is by definition agnostic to the actual backdoor pattern and only depends on the magnitude of the backdoor pattern together with the number of training instances, we mainly focus on these three representative backdoor patterns.

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 as 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 is 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.

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 serve 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.

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 the models trained on a dataset after removing backdoored instances detected by the three state-of-the-art backdoor detection baseline methods [6, 43, 42] introduced in Section 7.1.3. As illustrated in Table 1, RAB achieves comparable empirical robust accuracy compared with these empirical baseline methods.

7.2.4 Certified Accuracy Under different Radii.

We further discuss how different certified radii affect the certified accuracy. In Fig. 4, we present the certified accuracy as a function of the robust radius given different values for the smoothing parameter σ against blending attack. The conclusions on other backdoor patterns are similar. We can see that the certified accuracy decreases with increased radii and, at a certain point, it suddenly goes to zero, which aligns with existing observations on certified robustness against evasion attacks [10]. Furthermore, stronger noise harms the certified accuracy at a small radii, while improving it at a larger robust radius. It is thus essential to choose an appropriate smoothing noise magnitude according to the task. The certified accuracy of KNN is comparatively low due to its simple structure, but it achieves non-trivial certified accuracy at larger radius as we do not need Monte Carlo sampling which would result in a finite sampling error that decreases the certified robustness.

7.2.5 Ablation Study: Impact of Deterministic Test-time Augmentation

We compare the certification accuracy of RAB with and without deterministic test-time augmentation in Figure 5. 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.

7.2.6 Ablation Study: Impact of Training Stability/Smoothness.

In addition to vanilla DNNs, we provide the certified robustness benchmarks for a set of smoothed models to explore factors that would help improve the certified accuracy. In particular, we train *smooth* models such as differentially private DNNs and evaluate RAB on them. In Appendix C, Table B.4 shows the results on MNIST, CIFAR-10 and ImageNet. We do not observe significantly improved certified accuracy on such differentially private models compared to vanilla DNNs. Thus, further improving the certified accuracy is a non-trivial challenge, and differential privacy itself is not enough. It would be interesting to explore other efficient strategies to further smooth the models or improve model stability to improve the certified accuracy.

7.3 Certified Robustness of KNN Models

In this section, we present the benchmarks based on our proposed efficient algorithm for KNN models without requiring to sample from the smoothing noise distribution. We perform experiments on the MNIST, CIFAR-10, and UCI spambase datasets and show the results for $K = 3$ in Tables 2 and 3. From Table 2, we see that the KNN model achieves high certified accuracy on tabular data, which indicates its advantages for specific domains. As expected, from Table 3, we can see that the performance of KNN on the raw MNIST and CIFAR-10 images drops significantly since KNN is too simple for the vision tasks. However, RAB still achieves nontrivial certified accuracy and highest empirical robust accuracy compared with baselines. As for the baseline defense methods, we note that all of them rely on an intermediate feature vector for the input, which does not exist

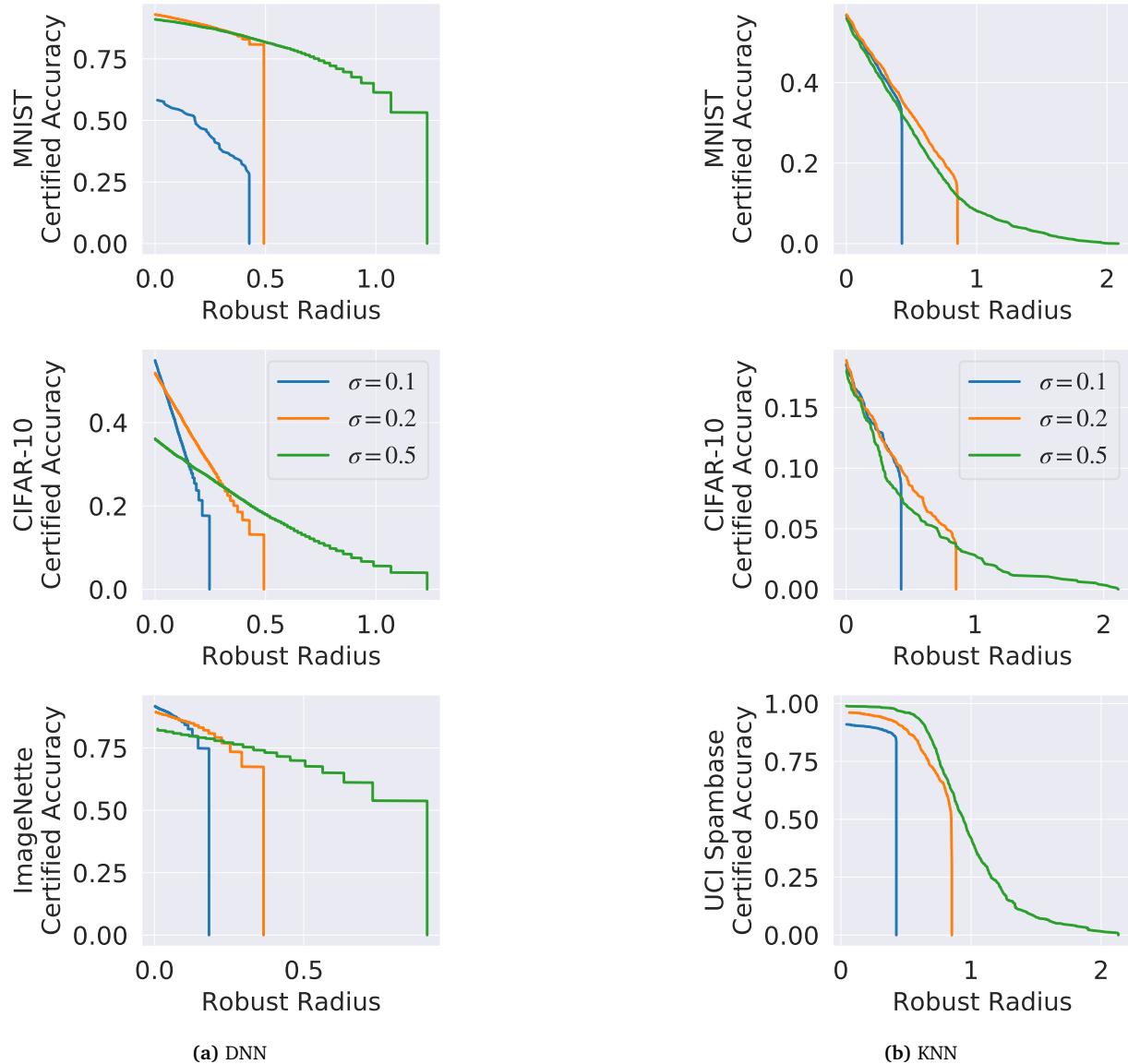


Figure 4: Certified accuracy of DNN and KNN at different radii with different smoothing parameter σ against blending attack.

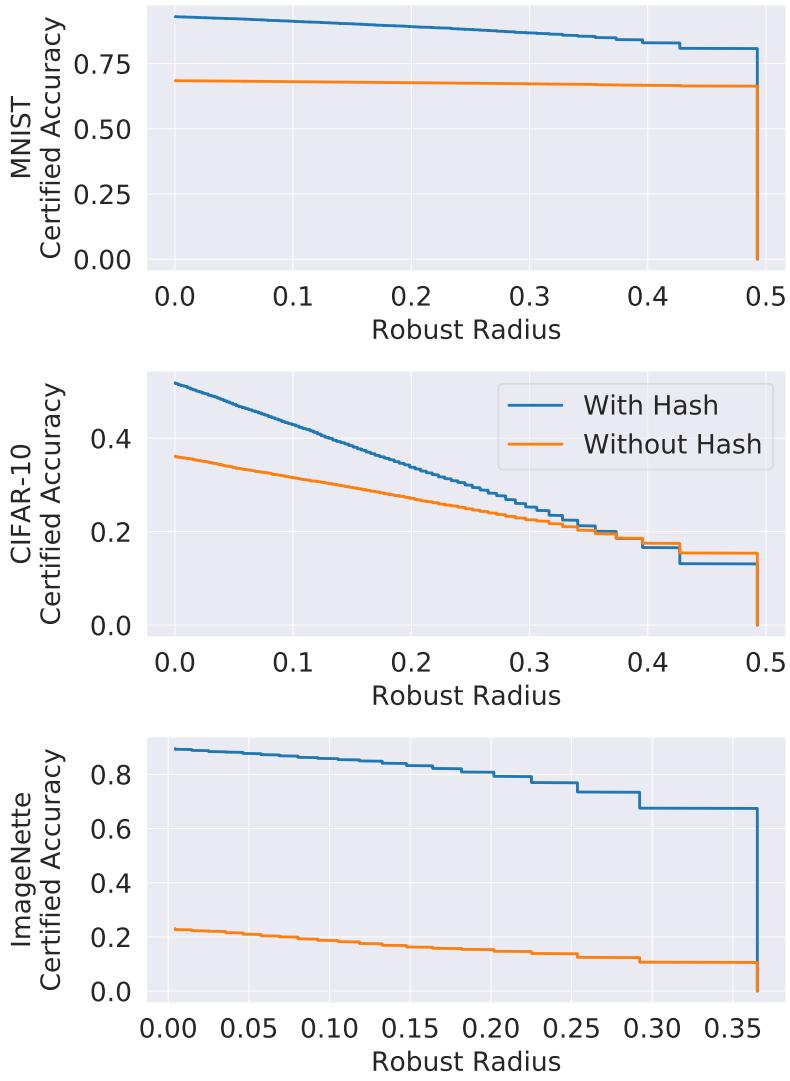


Figure 5: Comparison of the certified accuracy at different radii with and without the proposed deterministic test-time augmentation. The accuracy is evaluated against blending attack with smoothing parameter $\sigma = 0.2$.

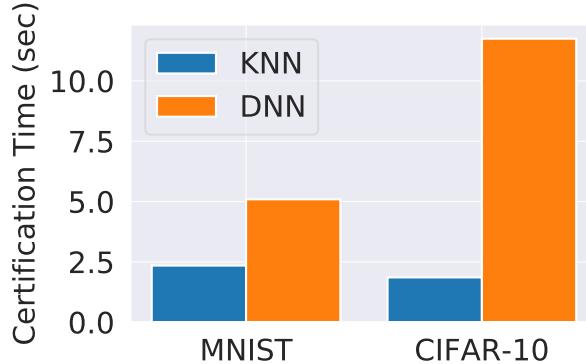


Figure 6: Runtime comparison for certifying one input.

in a KNN model. Therefore, we use the output prediction vector of the model as the representation. Not surprisingly, these approaches cannot achieve a good certification on the KNN models.

Figure 6 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 future work to design similar efficient certification algorithms for DNNs.

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 [3, 26]. Furthermore, [33] 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, 16]. The backdoor patterns can be either static or generated dynamically [49]. Static backdoor patterns can be as small as one pixel, or as large as an entire image [8].

Potential defenses against backdoor attacks Given the potential severe consequences caused by backdoor attacks, multiple defense approaches have been proposed. NeuralCleanse [45] 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 apply 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 [43] identifies the “spectral signature” in the activation vector for backdoored instances. STRIP [14] 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 [31], differential privacy has been leveraged as a defense against poisoning attacks. Note that RAB would not guarantee the trained models are differentially private, although they both aim to decrease the model sensitivity intuitively. A further empirical defense against backdoor attacks is proposed in [18] using covariance estimation with the aim of amplifying the spectral

signature of backdoored training instances. Finally, another interesting application of randomized smoothing is presented in [36] 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, [25] 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. In addition to these works aiming to certify the robustness for a single model, [52] provides a new way to certify the robustness of an end to end sensing-reasoning pipeline. Recently, a technical report also proposes to directly apply the randomized smoothing technique to certify robustness against backdoor attacks without any evaluation or analysis [44]. 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 model robustness.

9 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 particular, we evaluate our certified robustness against backdoors on DNNs and KNN models. 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.

References

- [1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.
- [2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.
- [3] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on Machine Learning*, page 1467–1474, Madison, WI, USA, 2012. Omnipress.
- [4] Xiaoyu Cao and Neil Zhenqiang Gong. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, pages 278–287, 2017.
- [5] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM, 2017.

- [6] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018.
- [7] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz 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*, pages 4658–4664. AAAI Press, 2019.
- [8] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.
- [9] Edward Chou, Florian Tramèr, Giancarlo Pellegrino, and Dan Boneh. Sentinel: Detecting physical attacks against deep learning systems. *arXiv preprint arXiv:1812.00292*, 2018.
- [10] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320, 09–15 Jun 2019.
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [12] Min Du, Ruoxi Jia, and Dawn Song. Robust anomaly detection and backdoor attack detection via differential privacy. In *International Conference on Learning Representations*, 2020.
- [13] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
- [14] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, page 113–125, New York, NY, USA, 2019. Association for Computing Machinery.
- [15] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- [16] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.
- [17] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.
- [18] Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. Spectre: defending against backdoor attacks using robust statistics. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4129–4139. PMLR, 18–24 Jul 2021.
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [20] Jeremy Howard. Imagenette.
- [21] Bojan Karlaš, Peng Li, Renzhi Wu, Nezihe Merve Gürel, Xu Chu, Wentao Wu, and Ce Zhang. Nearest neighbor classifiers over incomplete information: From certain answers to certain predictions. *Proc. VLDB Endow.*, 14(3):255–267, nov 2020.
- [22] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

- [23] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [24] 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)*, pages 656–672, 2019.
- [25] Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defenses against general poisoning attacks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.
- [26] Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. Data poisoning attacks on factorization-based collaborative filtering. In *Advances in neural information processing systems*, pages 1885–1893, 2016.
- [27] Linyi Li, Xiangyu Qi, Tao Xie, and Bo Li. Sok: Certified robustness for deep neural networks. *arXiv preprint arXiv:2009.04131*, 2020.
- [28] Linyi Li, Maurice Weber, Xiaojun Xu, Luka Rimanic, Bhavya Kailkhura, Tao Xie, Ce Zhang, and Bo Li. Tss: Transformation-specific smoothing for robustness certification. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, CCS ’21*, page 535–557, 2021.
- [29] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 369–385, 2018.
- [30] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Michael E. Houle, Dawn Song, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations*, 2018.
- [31] Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. Data poisoning against differentially-private learners: Attacks and defenses. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4732–4738. International Joint Conferences on Artificial Intelligence Organization, 7 2019.
- [32] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.
- [33] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 27–38. ACM, 2017.
- [34] J Neyman and E Pearson. On the problem of the most efficient tests of statistical hypotheses. 231. *Phil. Trans. Roy. Statistical Soc. A*, 289, 1933.
- [35] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, et al. Adversarial robustness toolbox v1. 0.0. *arXiv preprint arXiv:1807.01069*, 2018.
- [36] Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, and Zico Kolter. Certified robustness to label-flipping attacks via randomized smoothing. In *International Conference on Machine Learning*, pages 8230–8241. PMLR, 2020.
- [37] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11957–11965, 2020.

- [38] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models. *arXiv preprint arXiv:2003.03675*, 2020.
- [39] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9389–9398. PMLR, 18–24 Jul 2021.
- [40] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In *International Conference on Machine Learning*, pages 9389–9398. PMLR, 2021.
- [41] Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 6106–6116. Curran Associates Inc., 2018.
- [42] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. Certified defenses for data poisoning attacks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3520–3532, 2017.
- [43] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems*, pages 8000–8010, 2018.
- [44] Binghui Wang, Xiaoyu Cao, Neil Zhenqiang Gong, et al. On certifying robustness against backdoor attacks via randomized smoothing. *arXiv preprint arXiv:2002.11750*, 2020.
- [45] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks*, page 0. IEEE, 2019.
- [46] Wikipedia contributors. Sha-2 — Wikipedia, the free encyclopedia, 2020. [Online; accessed 18-March-2020].
- [47] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, page 3905–3911. AAAI Press, 2018.
- [48] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *25th Annual Network and Distributed System Security Symposium*. The Internet Society, 2018.
- [49] Chaofei Yang, Qing Wu, Hai Li, and Yiran Chen. Generative poisoning attack method against neural networks. *arXiv preprint arXiv:1703.01340*, 2017.
- [50] Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, pages 10693–10705. PMLR, 2020.
- [51] Zhuolin Yang, Bo Li, Pin-Yu Chen, and Dawn Song. Characterizing audio adversarial examples using temporal dependency. In *International Conference on Learning Representations*, 2019.
- [52] Zhuolin Yang, Zhikuan Zhao, Hengzhi Pei, Boxin Wang, Bojan Karlas, Ji Liu, Heng Guo, Bo Li, and Ce Zhang. End-to-end robustness for sensing-reasoning machine learning pipelines. *arXiv preprint arXiv:2003.00120*, 2020.

- [53] Haoti Zhong, Cong Liao, Anna Cinzia Squicciarini, Sencun Zhu, and David 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*, pages 97–108, 2020.
- [54] Chen Zhu, W. Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7614–7623. PMLR, 09–15 Jun 2019.

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 the 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

$$\phi(x) = \begin{cases} 1 & \text{if } \Lambda(x) > t, \\ q & \text{if } \Lambda(x) = t, \\ 0 & \text{if } \Lambda(x) < t. \end{cases} \quad \text{with } \Lambda(x) = \frac{f_{X_1}(x)}{f_{X_0}(x)}, \quad (21)$$

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

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 $t_p := \inf\{t \geq 0 : \mathbb{P}(\Lambda(X_0) \leq t) \geq p\}$. Then it holds that*

$$\mathbb{P}(\Lambda(X_0) < t_p) \leq p \leq \mathbb{P}(\Lambda(X_0) \leq t_p). \quad (22)$$

Proof. We first show the RHS of inequality (22). 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 $A_n := \{x : \Lambda(x) \leq t_n\}$ and note that $\mathbb{P}(\Lambda(X_0) \leq t_n) = \mathbb{P}(X_0 \in A_n)$. Clearly, if $x \in \{x : \Lambda(x) \leq t\}$ then $\forall n : \Lambda(x) \leq t \leq t_n$ and thus $x \in \cap_n A_n$. If on the other hand $x \in \cap_n A_n$ then $\forall n : \Lambda(x) \leq t_n \rightarrow t$ as $n \rightarrow \infty$. Hence, we have that $\cap_n A_n = \{x : \Lambda(x) \leq t\}$ and thus $\lim_{n \rightarrow \infty} \mathbb{P}(\Lambda(X_0) \leq t_n) = \mathbb{P}(\Lambda(X_0) \leq t)$ since $\lim_{n \rightarrow \infty} \mathbb{P}(X_0 \in A_n) = \mathbb{P}(X_0 \in \cap_n A_n)$ 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 (22). For that purpose, consider the set $B_n := \{x : \Lambda(x) < t_p - 1/n\}$ and let $B := \{x : \Lambda(x) < t_p\}$. 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 $\mathbb{P}(\Lambda(X_0) < t_p) = \lim_{n \rightarrow \infty} \mathbb{P}(X_0 \in B_n) \leq p$ since $B_n \subseteq B_{n+1}$. This concludes the proof. \square

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 $\phi : \mathcal{Z} \rightarrow [0, 1]$ 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 (21). Then, for any other test ϕ we have

$$1 - \beta(\phi^*) - \beta(\phi) = \quad (23)$$

$$= \int_{\Lambda > t} \phi f_1 d\mu + \int_{\Lambda \leq t} (\phi - 1) f_1 d\mu + q \int_{\Lambda = t} f_1 d\mu \quad (24)$$

$$= \int_{\Lambda > t} \phi \Lambda f_0 d\mu + \int_{\Lambda \leq t} \underbrace{(\phi - 1)}_{\leq 0} \Lambda f_0 d\mu + q \int_{\Lambda = t} \Lambda f_0 d\mu \quad (25)$$

$$\geq t \cdot \left[\int_{\Lambda > t} \phi f_0 d\mu + \int_{\Lambda \leq t} (\phi - 1) f_0 d\mu + q \int_{\Lambda = t} f_0 d\mu \right] \quad (26)$$

$$= t \cdot [\alpha(\phi) - (1 - \alpha(\phi^*))] \geq 0 \quad (27)$$

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

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 $t_p := \inf\{t \geq 0 : \mathbb{P}(\Lambda(Z) \leq t) \geq p\}$ and

$$q_p = \begin{cases} 0 & \text{if } \mathbb{P}(\Lambda(Z) = t_p) = 0, \\ \frac{\mathbb{P}(\Lambda(Z) \leq t_p) - p}{\mathbb{P}(\Lambda(Z) = t_p)} & \text{otherwise.} \end{cases} \quad (28)$$

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

$$\begin{aligned} \mathbb{P}(\Lambda(Z) \leq t_p) &= \mathbb{P}(\Lambda(Z) < t_p) + \mathbb{P}(\Lambda(Z) = t_p) \\ &\leq p + \mathbb{P}(\Lambda(Z) = t_p) \end{aligned} \quad (29)$$

and hence $q_p \in [0, 1]$. For $p \in [0, 1]$, let ϕ_p be the likelihood ratio test defined in (21) 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 (10) that $\mathbb{E}(p(y_A|x + X, \mathcal{D} + D)) = q(y_A|x, \mathcal{D}) \geq 1 - \alpha(\phi_A)$ and thus, by applying the first part of Lemma A.2 to the functions $\phi \equiv p(y_A|x + X, \mathcal{D} + D)$ and $\phi^* \equiv \phi_A$, it follows that

$$q(y_A|x + \Omega_x, \mathcal{D} + \Delta) = 1 - \beta(\phi) \geq \beta(\phi_A). \quad (30)$$

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 (10) that $\mathbb{E}(p(y|x + X, \mathcal{D} + D)) = q(y|x, \mathcal{D}) \leq p_B = \alpha(\phi_B)$. Thus, applying the second part of Lemma A.2 to the functions $\phi \equiv p(y|x + X, \mathcal{D} + D)$ and $\phi^* \equiv \phi_B$ yields

$$q(y|x + \Omega_x, \mathcal{D} + \Delta) = 1 - \beta(\phi) \leq 1 - \beta(\phi_B). \quad (31)$$

Combining (30) and (31) 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. \square

A.2 Proof of Theorem 2

Proof. We show tightness by constructing a base classifier h^* , such that the smoothed classifier is consistent with the class probabilities (10) for a given (fixed) input (x_0, \mathcal{D}_0) but whose smoothed version is not robust for adversarial perturbations (Ω_x, Δ) that violate (11). Let ϕ_A and ϕ_B be two likelihood ratio tests for testing

the null $Z \sim \mathbb{P}_0$ against the alternative $Z + (\Omega_x, \Delta) \sim \mathbb{P}_1$ and let ϕ_A be such that $\alpha(\phi_A) = 1 - p_A$ and ϕ_B such that $\alpha(\phi_B) = p_B$. Since (Ω_x, Δ) violates (11), we have that $\beta(\phi_A) + \beta(\phi_B) \leq 1$. Let p^* be given by

$$p^*(y|x, \mathcal{D}) = \begin{cases} 1 - \phi_A(x - x_0, \mathcal{D} - \mathcal{D}_0) & y = y_A \\ \phi_B(x - x_0, \mathcal{D} - \mathcal{D}_0) & y = y_B \\ \frac{1 - p^*(y_A|x, \mathcal{D}) - p(y_B|x, \mathcal{D})}{C-2} & \text{o.w.} \end{cases} \quad (32)$$

where the notation $\mathcal{D} - \mathcal{D}_0$ denotes subtraction on the features but not on the labels. Note that for binary classification, $C = 2$ we have that $\phi_A = \phi_B$ and hence p^* is well defined since in this case, by assumption $p_A + p_B = 1$. If $C > 2$, note that it follows immediately from the definition of p^* that $\sum_k p^*(y|x, \mathcal{D}) = 1$. Note that, from the construction of ϕ_A and ϕ_B in the proof of Theorem 1 (Appendix A.1) that (pointwise) $\phi_A \geq \phi_B$ provided $p_A + p_B \leq 1$. It follows that for $y \neq y_A, y_B$ we have $p^*(y|x, \mathcal{D}) \propto \phi_A - \phi_B \geq 0$. Thus, p^* is a well defined (conditional) probability distribution over labels and $h^*(x, \mathcal{D}) := \arg \max_y p^*(y|x, \mathcal{D})$ is a base classifier. Furthermore, to see that the corresponding smoothed classifier q^* is consistent with the class probabilities (10), consider

$$q^*(y_A|x_0, \mathcal{D}_0) = \mathbb{E}(p^*(y_A|x_0 + X, \mathcal{D}_0 + D)) \quad (33)$$

$$= \mathbb{E}(1 - \phi_A(X, D)) = 1 - \alpha(\phi_A) = p_A \quad (34)$$

and

$$q^*(y_B|x_0, \mathcal{D}_0) = \mathbb{E}(p^*(y_B|x_0 + X, \mathcal{D}_0 + D)) \quad (35)$$

$$= \mathbb{E}(\phi_B(X, D)) = \alpha(\phi_B) = p_B. \quad (36)$$

In addition, for any $y \neq y_A, y_B$, we have $q^*(y|x_0, \mathcal{D}_0) = (1 - p_A - p_B)/(C - 2) \leq p_B$ since by assumption $p_A + p_B \geq 1 - (C - 2) \cdot p_B$. Thus, q^* is consistent with the class probabilities (10). In addition, note that $q^*(y_A|x_0 + \Omega_x, \mathcal{D}_0 + \Delta) = 1 - \beta(\phi_A)$ and $\beta(\phi_B) = q^*(y_B|x_0 + \Omega_x, \mathcal{D}_0 + \Delta)$. Since by assumption $1 - \beta(\phi_A) < \beta(\phi_B)$ we see that indeed $y_A \neq q^*(x_0 + \Omega_x, \mathcal{D}_0 + \Delta)$. \square

A.3 Proof of Corollaries 1 and 2

Proof of Corollary 1. We prove this statement by direct application of Theorem 1. Let $Z = (X, D)$ be the smoothing distribution for q and let $\tilde{Z} := (\Omega_x, \Delta) + Z$ and $\tilde{Z}' := (0, -\Delta) + \tilde{Z}$. Correspondingly, let $\tilde{q}(y|x, \mathcal{D}) = q(y|x + \Omega_x, \mathcal{D} + \Delta)$. By assumption, we have that $\tilde{q}(y_A|x, \mathcal{D}) \geq p_A$ and $\max_{y \neq y_A} \tilde{q}(y|x, \mathcal{D}) \leq p_B$. We will now apply Theorem 1 to the smoothed classifier \tilde{q} . By Theorem 1, there exist likelihood ratio tests ϕ_A and ϕ_B for testing \tilde{Z} against \tilde{Z}' such that, if

$$\beta(\phi_A) + \beta(\phi_B) > 1 \quad (37)$$

then it follows that $y_A = \arg \max_y \tilde{q}(y|x, \mathcal{D} - \Delta)$. The statement then follows, since $\tilde{q}(y|x, \mathcal{D} - \Delta) = \arg \max_y \tilde{q}(y|x + \Omega_x, \mathcal{D} + \Delta)$. We will now construct the corresponding likelihood ratio tests and show that (37) has the form (13). Note that the likelihood ratio between \tilde{Z} and \tilde{Z}' at $z = (x, d)$ is given by $\Lambda(z) = \exp(\sum_{i=1}^n \langle d_i, -\delta_i \rangle_\Sigma + \frac{1}{2} \langle \delta_i, \delta_i \rangle_\Sigma)$ where $\Sigma = \sigma^2 \mathbf{I}_d$ and $\langle a, b \rangle_\Sigma := \sum_{i=1}^n a_i b_i / \sigma^2$. Thus, since singletons have probability 0 under the Gaussian distribution, any likelihood ratio test for testing \tilde{Z} against \tilde{Z}' has the form

$$\phi_t(z) = \begin{cases} 1, & \Lambda(z) \geq t. \\ 0, & \Lambda(z) < t. \end{cases} \quad (38)$$

For $p \in [0, 1]$, let

$$t_p := \exp \left(\Phi^{-1}(p) \sqrt{\sum_{i=1}^n \langle \delta_i, \delta_i \rangle_\Sigma} - \frac{1}{2} \sum_{i=1}^n \langle \delta_i, \delta_i \rangle_\Sigma \right) \quad (39)$$

and note that $\alpha(\phi_{t_p}) = 1 - p$ since

$$\alpha(\phi_{t_p}) = 1 - \Phi\left(\frac{\log(t_p) + \frac{1}{2} \sum_{i=1}^n \langle \delta_i, \delta_i \rangle_\Sigma}{\sqrt{\sum_{i=1}^n \langle \delta_i, \delta_i \rangle_\Sigma}}\right) \quad (40)$$

where Φ is the CDF of the standard normal distribution. Thus, the test $\phi_A \equiv \phi_{t_A}$ with $t_A \equiv t_{p_A}$ satisfies $\alpha(\phi_A) = 1 - p_A$ and the test $\phi_B \equiv \phi_{t_B}$ with $t_B \equiv t_{1-p_B}$ satisfies $\alpha(\phi_B) = p_B$. Computing the type II error probability of ϕ_A yields $\beta(\phi_A) = \Phi\left(\Phi^{-1}(p_A) - \sqrt{\sum_{i=1}^n \langle \delta_i, \delta_i \rangle_\Sigma}\right)$. and, similarly, the type II error probability of ϕ_B is given by $\beta(\phi_B) = \Phi\left(\Phi^{-1}(1 - p_B) - \sqrt{\sum_{i=1}^n \langle \delta_i, \delta_i \rangle_\Sigma}\right)$. Finally, we see that $\beta(\phi_A) + \beta(\phi_B) > 1$ is satisfied if and only if $\sqrt{\sum_{i=1}^n \|\delta_i\|_2^2} < \frac{\sigma}{2} (\Phi^{-1}(p_A) - \Phi^{-1}(p_B))$ what concludes the proof. \square

Proof of Corollary 2. We proceed analogously to the proof of Corollary 1 but with a uniform distribution on the feature vectors $D \sim \mathcal{U}([a, b])$ and construct the likelihood ratio tests in the uniform case. Denote by $S := \prod_{i=1}^n S_i$, $S_i := \prod_{j=1}^d [a + \delta_{ij}, b + \delta_{ij}]$ the support of $\tilde{D} := \Delta + D$ and by $S' := \prod_{i=1}^n [a, b]^d$ the support of $\tilde{D}' := D$. Note that the likelihood ratio between \tilde{Z} against \tilde{Z}' at $z = (x, w, v)$ for any $w \in S \cup S'$ is given by

$$\Lambda(z) = \frac{f_{\tilde{Z}'}(z)}{f_{\tilde{Z}}(z)} = \frac{f_{\tilde{W}'}(w)}{f_{\tilde{W}}(w)} = \begin{cases} 0 & w \in S \setminus S', \\ 1 & w \in S \cap S', \\ \infty & w \in S' \setminus S. \end{cases} \quad (41)$$

and that any likelihood ratio test for testing \tilde{Z} against \tilde{Z}' has the form (21). We now construct such likelihood ratio tests ϕ_A, ϕ_B with $\alpha(\phi_A) = 1 - p_A$ and $\alpha(\phi_B) = p_B$ by following the construction in the proof of Theorem 1. Specifically, we compute q_A, t_A such that these type I error probabilities are satisfied. Notice that $p_0 := \mathbb{P}(\tilde{W} \in S \setminus S') = 1 - \prod_{i=1}^n \left(\prod_{j=1}^d \left(1 - \frac{|\delta_{ij}|}{b-a}\right)_+ \right)$ where $(x)_+ = \max\{x, 0\}$. For $t \geq 0$ we have $\mathbb{P}(\Lambda(\tilde{Z}) \leq t) = p_0$ if $t < 1$ and 1 otherwise. Thus $t_p := \inf\{t \geq 0 : \mathbb{P}(\Lambda(\tilde{Z}) \leq t) \geq p\}$ is given by $t_p = 0$ if $p \leq p_0$ and $t_p = 1$ if $p > p_0$. We notice that, if $p_A \leq p_0$, then $t_A \equiv t_{p_A} = 0$. This implies that the type II error probability of the corresponding test ϕ_A is 0 since in this case

$$\beta(\phi_A) = 1 - \mathbb{P}(\Lambda(\tilde{Z}') > 0) - q_A \cdot \mathbb{P}(\Lambda(\tilde{Z}') = 0) \quad (42)$$

$$= 1 - \mathbb{P}(\tilde{D}' \in S') - q_A \cdot \mathbb{P}(\tilde{D}' \in S \setminus S') = 0. \quad (43)$$

Similarly, if $1 - p_B \leq p_0$ then $t_B \equiv t_{p_B} = 0$ and we obtain that the corresponding test ϕ_B satisfies $\beta(\phi_B) = 0$. In both cases $\beta(\phi_A) + \beta(\phi_B) > 1$ can never be satisfied and we find that $p_A > p_0$ and $1 - p_B > p_0$ is a necessary condition. In this case, we have that $t_A = t_B = 1$. Let q_A and q_B be defined as in the proof of Theorem 1

$$q_A := \frac{\mathbb{P}(\Lambda(\tilde{Z}) \leq 1) - p_A}{\mathbb{P}(\Lambda(\tilde{Z}) = 1)} = \frac{1 - p_A}{1 - p_0}, \quad (44)$$

$$q_B := \frac{\mathbb{P}(\Lambda(\tilde{Z}) \leq 1) - (1 - p_B)}{\mathbb{P}(\Lambda(\tilde{Z}) = 1)} = \frac{1 - (1 - p_B)}{1 - p_0}. \quad (45)$$

Clearly, the corresponding likelihood ratio tests ϕ_A and ϕ_B have significance $1 - p_A$ and p_B respectively. Furthermore, notice that

$$\mathbb{P}(\tilde{D}' \in S' \setminus S) = \mathbb{P}(\tilde{D} \in S \setminus S') = p_0 \quad (46)$$

$$\mathbb{P}(\tilde{D}' \in S' \cap S) = \mathbb{P}(\tilde{D} \in S' \cap S) = 1 - p_0 \quad (47)$$

and hence $\beta(\phi_A)$ is given by

$$\beta(\phi_A) = 1 - \mathbb{P}(\Lambda(\tilde{Z}') > 1) - q_A \cdot \mathbb{P}(\Lambda(\tilde{Z}') = 1) \quad (48)$$

$$= 1 - p_0 - q_A \cdot (1 - p_0) = p_A - p_0. \quad (49)$$

and similarly

$$\beta(\phi_B) = 1 - \mathbb{P}(\Lambda(\tilde{Z}') > 1) - q_B \cdot \mathbb{P}(\Lambda(\tilde{Z}') = 1) \quad (50)$$

$$= 1 - p_0 - q_B \cdot (1 - p_0) = 1 - p_B - p_0. \quad (51)$$

Finally, the statement follows, since $\beta(\phi_A) + \beta(\phi_B) > 1$ if and only if $1 - \left(\frac{p_A - p_B}{2}\right) < \prod_{i=1}^n \left(\prod_{j=1}^d \left(1 - \frac{|\delta_{ij}|}{b_j - a_j}\right)_+ \right)$. \square

B Smoothed K-NN Classifiers

We first formalize K -NN classifiers which use quantized Euclidean distance as a notion of similarity. Specifically, let $B_1 = [0, b_1), \dots, B_L := [b_{L-1}, \infty)$ be similarity buckets with increasing $b_1 < b_2 < \dots < b_{L-1}$ and associated similarity levels $\beta_1 > \beta_2 > \dots > \beta_L$. Then for $x, x' \in \mathbb{R}^d$ we define their similarity as $\kappa(x, x') := \sum_{l=1}^L \beta_l \mathbb{1}_{B_l}(\|x - x'\|_2^2)$ where $\mathbb{1}_{B_l}$ is the indicator function of B_l . Given a dataset $D = (x_i, y_i)_{i=1}^n$ and a test instance x , we define the relation

$$x_i \succeq x_j \iff \begin{cases} \kappa(x_i, x) > \kappa(x_j, x) & \text{if } i > j \\ \kappa(x_i, x) \geq \kappa(x_j, x) & \text{if } i \leq j \end{cases} \quad (52)$$

which says that the instance x_i is more similar to x , if either it has strictly larger similarity or, if it has the same similarity as x_j , then x_i is more similar if $i < j$. With this binary relation, we define the set of K nearest neighbours of x as $I_K(x, D) := \{i : |\{j \neq i : x_j \succeq x_i\}| \leq K - 1\} \subseteq [n]$ and summarize the per class votes in I_K as a label tally vector $\gamma_k(x, D) := \#\{i \in I_K(x, D) : y_i = k\}$. The prediction of a K -NN classifier is then given by $K\text{-NN}(x, D) = \arg \max_k \gamma_k(x, D)$.

B.1 Proof of Theorem 3

Proof. Our goal is to show that we can compute the smoothed classifier q with $Z = (0, D)$, $D \sim \prod_{i=1}^n \mathcal{N}(0, \sigma^2 \mathbb{1}_d)$ and defined by the probability

$$q(y | x, D) = \mathbb{P}_D(K\text{-NN}(x, D + D) = y) \quad (53)$$

in time $\mathcal{O}(K^{2+C} \cdot n^2 \cdot L \cdot C)$. For ease of notation, let $X_i := x_i + D^{(i)}$ and $S_i := \kappa(X_i, x)$ and note that $p_i^l := \mathbb{P}(S_i = \beta_l) = F_{d, \lambda_i}(\frac{b_l}{\sigma^2}) - F_{d, \lambda_i}(\frac{b_{l-1}}{\sigma^2})$ where F_{d, λ_i} is the CDF of the non-central χ^2 -distribution with d degrees of freedom and non-locality parameter $\lambda_i = \|x_i + \delta_i - x\|_2^2 / \sigma^2$. Note that we can write (53) equivalently as $\mathbb{P}_D(\arg \max_{k'} \gamma_{k'}(x, D + D)) = y$ and thus $q(y | x, D) = \sum_{\gamma \in \Gamma_k} \mathbb{P}_D(\gamma(x, D + D) = \gamma)$ with $\Gamma_k := \{\gamma \in [K]^C : \arg \max_{k'} \gamma_{k'} = k\}$. Next, we notice that the event that a tally vector γ occurs, can be partitioned into the events that lead to the given γ and for which instance i has similarity β_l and is in the top- K but not in the top- $(K - 1)$. We define these to be the boundary events $\mathcal{B}_i^l(\gamma) := \{\forall c : \#\{j \in I_c : X_j \succeq X_i\} = \gamma_c, S_i = \beta_l\}$ where $I_c = \{i : y_i = c\}$. The probability that a given tally vector γ occurs is thus given by $\mathbb{P}_D(\gamma(x, D + D) = \gamma) = \sum_{i=1}^n \sum_{l=1}^L \mathbb{P}(\mathcal{B}_i^l(\gamma))$.

For fixed i we notice that the for different classes, the events $\{\#\{j \in I_c : X_j \succeq X_i\} = \gamma_c\}$ are pairwise independent, conditioned on $\{S_i = \beta_l\}$. Writing $P_c^l(i, \gamma)$ for the conditional probability $\mathbb{P}(\#\{i \in I_c : y_i = c\} = \gamma_c | S_i = \beta_l)$

Table B.4: Evaluation on smoothed DNN models trained with DP-SGD.

	Attack Approach	Attack Setting	Attack Success Rate	σ	Clean/Backdoor Acc	Certified Acc at $R = 0.2/0.5/1.0/2.0$	Certified Rate at $R = 0.2/0.5/1.0/2.0$
MNIST	One-pixel 	$\ \delta_i\ _2 = 1.0, r_p = 0.1$	0.004	1.0	0.997 / 0.997	0.997 / 0.994 / 0.974 / 0.726	0.999 / 0.994 / 0.974 / 0.726
				2.0	0.997 / 0.997	0.995 / 0.994 / 0.976 / 0.540	0.997 / 0.994 / 0.976 / 0.540
	Four-pixel 	$\ \delta_i\ _2 = 1.0, r_p = 0.1$	0.006	1.0	0.997 / 0.998	0.998 / 0.998 / 0.994 / 0.904	0.998 / 0.998 / 0.994 / 0.904
				2.0	0.994 / 0.999	0.998 / 0.998 / 0.995 / 0.926	0.998 / 0.998 / 0.995 / 0.926
	Blending 	$\ \delta_i\ _2 = 1.0, r_p = 0.1$	0.004	1.0	0.997 / 0.998	0.998 / 0.998 / 0.986 / 0.916	0.999 / 0.998 / 0.986 / 0.916
				2.0	0.992 / 0.998	0.998 / 0.998 / 0.962 / 0.493	0.998 / 0.998 / 0.962 / 0.493
CIFAR	One-pixel 	$\ \delta_i\ _2 = 1.0, r_p = 0.1$	0.002	1.0	0.996 / 0.998	0.998 / 0.997 / 0.994 / 0.905	0.998 / 0.997 / 0.994 / 0.905
				2.0	0.994 / 0.998	0.998 / 0.997 / 0.994 / 0.905	0.998 / 0.997 / 0.994 / 0.905
	Four-pixel 	$\ \delta_i\ _2 = 1.0, r_p = 0.1$	0.006	1.0	0.997 / 0.997	0.995 / 0.994 / 0.985 / 0.735	0.997 / 0.995 / 0.985 / 0.735
				2.0	0.996 / 0.997	0.995 / 0.994 / 0.981 / 0.774	0.995 / 0.994 / 0.981 / 0.774
	Blending 	$\ \delta_i\ _2 = 1.0, r_p = 0.1$	0.004	1.0	0.997 / 0.998	0.998 / 0.998 / 0.986 / 0.916	0.999 / 0.998 / 0.986 / 0.916
				2.0	0.992 / 0.998	0.998 / 0.998 / 0.962 / 0.493	0.998 / 0.998 / 0.962 / 0.493
ImageNet	One-pixel 	$\ \delta_i\ _2 = 1.0, r_p = 0.1$	0.057	0.5	0.718 / 0.574	0.564 / 0.552 / 0.538 / 0.440	0.939 / 0.916 / 0.881 / 0.715
				1.0	0.689 / 0.528	0.520 / 0.510 / 0.482 / 0.401	0.919 / 0.893 / 0.844 / 0.715
	Four-pixel 	$\ \delta_i\ _2 = 1.0, r_p = 0.1$	0.141	0.5	0.771 / 0.770	0.764 / 0.755 / 0.729 / 0.671	0.974 / 0.950 / 0.902 / 0.800
				1.0	0.750 / 0.730	0.720 / 0.712 / 0.694 / 0.645	0.958 / 0.944 / 0.920 / 0.848
	Blending 	$\ \delta_i\ _2 = 1.0, r_p = 0.1$	0.273	0.5	0.729 / 0.604	0.589 / 0.577 / 0.554 / 0.477	0.950 / 0.926 / 0.878 / 0.740
				1.0	0.680 / 0.494	0.477 / 0.461 / 0.438 / 0.364	0.919 / 0.893 / 0.849 / 0.726
ImageNet		$\ \delta_i\ _2 = 0.1, r_p = 0.02$	0.306	0.5	0.771 / 0.781	0.772 / 0.764 / 0.742 / 0.669	0.970 / 0.950 / 0.901 / 0.794
				1.0	0.756 / 0.748	0.745 / 0.735 / 0.723 / 0.662	0.972 / 0.960 / 0.938 / 0.837
	One-pixel 	$\ \delta_i\ _2 = 1.0, r_p = 0.1$	0.337	0.5	0.729 / 0.605	0.589 / 0.570 / 0.548 / 0.454	0.943 / 0.921 / 0.879 / 0.721
				1.0	0.684 / 0.495	0.481 / 0.476 / 0.452 / 0.395	0.926 / 0.911 / 0.862 / 0.749
	Four-pixel 	$\ \delta_i\ _2 = 1.0, r_p = 0.1$	0.129	0.5	0.771 / 0.783	0.780 / 0.772 / 0.758 / 0.693	0.972 / 0.953 / 0.918 / 0.817
				1.0	0.753 / 0.734	0.727 / 0.720 / 0.704 / 0.658	0.963 / 0.950 / 0.924 / 0.841
	Blending 	$\ \delta_i\ _2 = 1.0, r_p = 0.1$	0.088	0.5	0.765 / 0.581	0.552 / 0.500 / 0.442 / 0.221	0.773 / 0.681 / 0.589 / 0.264
				1.0	0.617 / 0.319	0.288 / 0.288 / 0.230 / 0.153	0.695 / 0.695 / 0.573 / 0.397
		$\ \delta_i\ _2 = 0.1, r_p = 0.02$	0.030	0.5	0.834 / 0.786	0.768 / 0.731 / 0.690 / 0.458	0.857 / 0.802 / 0.744 / 0.476
				1.0	0.688 / 0.592	0.567 / 0.567 / 0.505 / 0.399	0.760 / 0.760 / 0.652 / 0.490

yields $\mathbb{P}(\mathcal{B}_i^l(\gamma)) = p_i^l \cdot \prod_{c=1}^C P_c^l(i, \gamma)$ and hence $q(y|x, \mathcal{D}) = \sum_{\gamma \in \Gamma_k} \sum_{i=1}^n \sum_{l=1}^L p_i^l \cdot \prod_{c=1}^C P_c^l(i, \gamma)$ which requires $\mathcal{O}(K^C \cdot n \cdot L \cdot C)$ evaluations of P_c^l . The next step is to compute the probabilities P_c^l . For that purpose we need to open up the binary relation \succeq . Suppose first that $y_i \neq c$. Then the event that exactly γ_c instances of class c satisfy $X_j \succeq X_i$ is the same as the event that for some $r \leq \gamma_c$ exactly r instances with index larger than i have similarity strictly larger than X_i and exactly $\gamma_c - r$ instances with index smaller than i have similarity larger or equal than X_i . Now suppose that $y_i = c$. Then, the event that exactly γ_c instances of the same class c satisfy $X_j \succeq X_i$ is the same as the event that for some $r \leq \gamma_c$ exactly r instances with index larger than i have similarity strictly larger than X_i and exactly $\gamma_c - r - 1$ instances with index smaller than i have similarity larger or equal than X_i . This reasoning allows us to write P_c^l in terms of functions

$$R_c^l(i, r) := \mathbb{P}(|\{j \in I_c : S_j > \beta_l, j > i\}| = r) \quad (54)$$

$$Q_c^l(i, r) := \mathbb{P}(|\{j \in I_c : S_j \geq \beta_l, j < i\}| = r) \quad (55)$$

as

$$P_c^l(i, \gamma) = \begin{cases} \sum_{r=0}^{\gamma_c} R_c^l(i, r) \cdot Q_c^l(i, \gamma_c - r) & y_i \neq c \\ \sum_{r=0}^{\gamma_c-1} R_c^l(i, r) \cdot Q_c^l(i, \gamma_c - r - 1) & y_i = c. \end{cases}$$

The functions R_c^l and Q_c^l exhibit a recursive structure that we wish to exploit to get an efficient algorithm. Specifically, we write $\alpha_i^l := \mathbb{P}(S_i \leq \beta_l) = \sum_{s=l}^L p_i^l$, and $\bar{\alpha}_i^l := 1 - \alpha_i^l$ and use the following recursion

$$R_c^l(i-1, r) = \begin{cases} R_c^l(i, r) & y_i \neq c \\ \bar{\alpha}_i^l \cdot R_c^l(i, r-1) + \alpha_i^l \cdot R_c^l(i, r) & y_i = c \end{cases}$$

starting at $i = n$ and $r = 0$ and with initial values $R_c^l(i, -1) = 0$, $R_c^l(n, 0) = 1$ and $R_c^l(n, r) = 0$ for $r \geq 1$.

Similarly,

$$Q_c^l(i+1, r) = \begin{cases} Q_c^l(i, r) & y_i \neq c \\ \bar{\alpha}_i^{l+1} \cdot Q_c^l(i, r-1) \\ + \alpha_i^{l+1} \cdot Q_c^l(i, r) & y_i = c \end{cases} \quad (56)$$

starting the recursion at $i = 1$ and $r = 0$ and with initial values $Q_c^l(i, -1) = 0$, $Q_c^l(1, 0) = 1$ and $Q_c^l(1, r) = 0$ for $r \geq 1$. Evaluating P_c^l requires $\mathcal{O}(K)$ calls to R_c^l and Q_c^l each. The computation of R_c^l and Q_c^l can be achieved in $\mathcal{O}(n \cdot K)$ if the values α_i^l are computed beforehand and stored separately (requiring $\mathcal{O}(n \cdot L)$ computations). Finally, the entire computation has a time complexity of $\mathcal{O}(K^{C+2} \cdot n^2 \cdot L \cdot C)$. \square

C Differentially Private DNN Models

Based on the intuition that a “smoothed” model is expected to be more robust against attacks, in this section, we provide a way to explore this assumption and further provide guidance on how to improve the certified robustness of machine learning models against backdoor attacks. To assess the effects of the model smoothness on the certified robustness bound, we augment the model training process with differentially private stochastic gradient descent (DP-SGD) [1]. Intuitively, this process will add another level of “smoothness” to the model training, which is encoded in the model parameters. However, as mentioned in the main text, here we do not observe significantly improved certified accuracy. It would be interesting to explore other efficient strategies to further smooth the models and ultimately improve the performance.