7-2024

Certified robust accuracy of neural networks are bounded due to Bayes errors

Ruihan ZHANG

Jun SUN
Singapore Management University, junsun@smu.edu.sg

Citation

Certified Robust Accuracy of Neural Networks Are Bounded Due to Bayes Errors

Ruihan Zhang[✉] and Jun Sun

Singapore Management University, Singapore, Singapore
{rhzhang,junsun}@smu.edu.sg

Abstract. Adversarial examples pose a security threat to many critical systems built on neural networks. While certified training improves robustness, it also decreases accuracy noticeably. Despite various proposals for addressing this issue, the significant accuracy drop remains. More importantly, it is not clear whether there is a certain fundamental limit on achieving robustness whilst maintaining accuracy. In this work, we offer a novel perspective based on Bayes errors. By adopting Bayes error to robustness analysis, we investigate the limit of certified robust accuracy, taking into account data distribution uncertainties. We first show that the accuracy inevitably decreases in the pursuit of robustness due to changed Bayes error in the altered data distribution. Subsequently, we establish an upper bound for certified robust accuracy, considering the distribution of individual classes and their boundaries. Our theoretical results are empirically evaluated on real-world datasets and are shown to be consistent with the limited success of existing certified training results, *e.g.*, for CIFAR10, our analysis results in an upper bound (of certified robust accuracy) of 67.49%, meanwhile existing approaches are only able to increase it from 53.89% in 2017 to 62.84% in 2023.

1 Introduction

Neural networks have achieved remarkable success in various applications, including many security-critical systems such as self-driving cars [24], and face-recognition-based authentication systems [44]. Unfortunately, several security issues of neural networks have been discovered as well. Arguably the most notable one is the presence of adversarial examples. Adversarial examples are inputs that are carefully crafted by adding human imperceptible perturbation to normal inputs to trigger wrong predictions [25]. Their existence poses a significant threat when the neural networks are deployed in security-critical scenarios. For example, adversarial examples can mislead road sign recognition systems of self-driving cars and cause accidents [24]. The increasing adoption of machine learning in security-sensitive domains raises concerns about the robustness of these models against adversarial examples [38].

To defend against adversarial examples, various methods for improving a model's robustness have been proposed. Two main categories are adversarial training [3,58] and certified training [35,45], both of which aim to improve a model's accuracy in the presence of adversarial examples. Adversarial training works by training the network with a mix of normal and adversarial examples, either pre-generated or generated during

(a) Is this a cat?   (b) Eh, it's a dog.
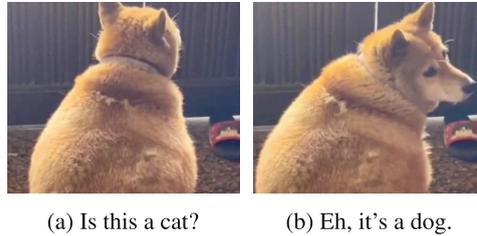
Fig. 1. The picture at left may look like a cat. In fact, it can be the back of a dog.

training. Methods in this category do not provide a formal robustness guarantee [63], leaving the system potentially vulnerable to new types of adversarial attacks [31,49].

In contrast, certified training aims to provide a formal guarantee of robustness. A method in this category typically incorporates robustness verification techniques [60] during training, *i.e.*, they aim to find a valuation of network parameters such that the model is provably robust with respect to the training samples. However, they are found to reduce the model's accuracy significantly [9]. Recent studies have shown that state-of-the-art certified training can result in up to 60% accuracy drop on CIFAR-10 [32] (at vicinity size 8/255). This is unacceptable for many real-world applications. Although numerous researchers attempt to enhance certified training methods, there seems to be an invisible hurdle preventing them from achieving a level of accuracy similar to that of vanilla models. Despite attempts to explore it using the limit of certain abstraction domains [33], in general, whether there is such a theoretical upper bound on certified robust accuracy or not remains an open problem.

In this work, we offer a novel perspective and argue that Bayes errors may be one of the reasons why there is such an invisible hurdle. The Bayes error, in the context of statistics and machine learning, is a fundamental concept related to the inherent uncertainty in any classification system [20]. It represents the minimum error rate for any classifier on a given problem and is determined by the overlap in the probability distributions of the classes to be predicted [13]. Thus, we study whether the Bayes errors put a limit on certified robust accuracy.

To understand how Bayes Error is relevant, we can consider it from the uncertainty in neural network learning. Most existing classifiers learn using a data set which gives a unique and certain label for each input [26]. Yet, this may not be the case in reality. That is, not every input may have a 100% certain label (due to reasons such as information loss during the picture-capturing process). Intuitively, we show a real-world example in Fig. 1. This image looks like a cat, while it is, in fact, also possible to be a dog. The point is that unless we know how this photo was taken, and there is no information loss during the photo taking, there may always be a certain level of uncertainty when we label the data. These uncertainties call for Bayes errors and actually bounds both vanilla and certified robust accuracy.

This work has two objectives. First, we aim to analyse whether the quest for robustness inevitably decreases model accuracy, from the perspective of Bayes errors. This requires examining how the inherent, irreducible error in class probability distributions

influences the robustness of classifiers (with respect to perturbations). We show that given the definition of robustness, the data distribution undergoes a convolution within the vicinity (*i.e.*, the region around an input which is defined by the perturbation budget). Second, we intend to quantify this potential decrease, *i.e.*, what are the upper bounds on the optimal certified robust accuracy? We show that such an upper bound can be derived independently from the classification algorithm. Through a detailed exploration of how each input may contribute to the Bayes errors, our study aims to enhance the understanding of their contribution to classification robustness.

We apply our analysis to multiple benchmark data sets and the corresponding models. From every data set, we observe that the convolved distribution has an increased Bayes error compared with the original distribution. This implies that pursuing robustness would in turn increase uncertainty, and decrease accuracy as we show in our analysis. Second, we contrast the state-of-the-art (SOTA) certified robust accuracy against the upper bound derived using our approach. This is to verify if the bound is empirically effective. We find that the bound is indeed higher than the state-of-the-art certified robust accuracy. We further investigate the relationship between the robustness upper bound and the perturbation vicinity size. When vicinity size grows, we expectedly obtain a decreased upper bound, on every data set used in our study.

2   Preliminary and Problem Definition

In this section, we review the relevant background of this study, including the fundamentals of robustness in machine learning, *e.g.*, its definition and verification. We also recall statistical decision theory, highlighting its relevance to classification. After that, we define our research problem.

In machine learning, the learner, denoted as a function $h : \mathbb{X} \to \mathbb{Y}$, is used to predict outputs $h(\boldsymbol{x}) \in \mathbb{Y}$ based on a (possibly high dimensional) input point $\boldsymbol{x} \in \mathbb{X}$. The quality of h can be measured by a problem-dependent loss function $\ell(h, \mathbf{x}, \mathbf{y})$ [66]. The choice of the loss function depends on the specific problem and data. Common options include the cross-entropy loss for classification and the mean squared error loss for regression. We focus on the classification problem in this work. Classification is the problem of assigning a class to each input [34], *i.e.*, the learner's task is to map an input to a discrete class and the learner is often called a classifier.

Definition 1 (Classifier). *In machine learning, a classifier maps an input $\boldsymbol{x}$ from an input space $\mathbb{X}$ to a discrete class y in the output space $\mathbb{Y}$. The output space $\mathbb{Y}$ is a (typically finite) set of discrete categories. Formally,*

$$h : \boldsymbol{x} \mapsto \hat{y}, \quad \boldsymbol{x} \in \mathbb{X}, \hat{y} \in \mathbb{Y}, \quad \mathbb{Y} = \{\, \mathrm{class}_i \mid i \in \mathbb{Z}^+, i \leq N_y \,\} \tag{1}$$

where N_y is the total number of categories the classifier can assign such that $|\mathbb{Y}| = N_y$.

For example, a spam classifier maps an email to $\{\, \mathrm{spam}, \mathrm{non\text{-}spam} \,\}$. The input vector for an email may embody the length of the message, the frequency of certain keywords in the body of the message, or the vectorised email body [34]. A learning example contains an input and a label. A classifier can learn from labelled

email examples and predict labels for other email examples. The classifier's predictions are then compared with the labels of the email under test to measure the performance of the classifier, *e.g.*, a zero-one misclassification loss may be defined over $\{\text{spam}, \text{non-spam}\} \times \{\text{spam}, \text{non-spam}\}$ by $l(\hat{y}, y) = \mathbf{1}_{\hat{y} \neq y}$. A lower loss on the test sample set indicates a more accurate classifier.

2.1 Robustness in Classification

A classifier may not be robust as small changes in input data might lead to significant changes in the predictions made by a classifier [47]. Consider the spam classifier example. Surprisingly, the removal of a single seemingly unimportant word from an email may switch the classifier's decision from spam to non-spam [55]. This phenomenon highlights the existence of adversarial examples, which are defined as follows.

Definition 2 (Adversarial example [17,25]). *Given a classifier $h : \mathbb{X} \to \mathbb{Y}$ and an input-label pair $(\boldsymbol{x}, y) \in \mathbb{X} \times \mathbb{Y}$, an adversarial example $\boldsymbol{x}' \in \mathbb{X}$ is an input that is similar to $\boldsymbol{x}$ but is classified wrongly, e.g., $h(\boldsymbol{x}') \neq y$. The difference between $\boldsymbol{x}$ and $\boldsymbol{x}'$ can be measured by a distance function d, and we often require the distance between $\boldsymbol{x}'$ and $\boldsymbol{x}$ to be smaller than some threshold ϵ. We assume $\boldsymbol{x}$ is correctly classified, i.e., $h(\boldsymbol{x}) = y$.*

Consider the case of the spam email. If a single word is removed, the Levenshtein distance (a measure of the number of edits needed to change one text into another) is 1. An adversarial example based on such a small change could be used with malicious intent. Even though removing a common word like 'just' does not alter the nature of a spam email, it might be enough to prevent it from being detected by the spam classifier. Therefore, robustness against such attacks is needed such that spam would not evade detection by just changing a few words. Formally, robustness is defined as follows [30].

Definition 3 (Classifier robustness against perturbations). *Given classifier h and example $(\boldsymbol{x}, y) \in \mathbb{X} \times \mathbb{Y}$, we say that h is robust with respect to vicinity $\{\boldsymbol{x}' \mid d(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon\}$, i.e., $\text{Rob}\left(h, \boldsymbol{x}, y; (d, \epsilon)\right)$, only if the following condition is satisfied.*

$$\neg \exists \, \boldsymbol{x}' \in \mathbb{X}. \, d(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon \wedge h(\boldsymbol{x}') \neq y \tag{2}$$

Definition 3 involves the concept of vicinity, which is a subset of the input space, *i.e.* $\subset \mathbb{X}$. It is usually determined by an input and a budget for perturbation. For instance, give an input $\boldsymbol{x}$, we can define its vicinity as $\mathbb{V}_{\boldsymbol{x}} = \{\boldsymbol{x}' \mid d(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon\}$. However, this set representation may be inconvenient sometimes. Thus, we give an equivalent function form as follows.

$$v_{\boldsymbol{x}}(\boldsymbol{x}') = \begin{cases} \left(\int_{\mathbb{V}_{\boldsymbol{x}}} d\boldsymbol{x}''\right)^{-1}, & \text{if } \boldsymbol{x}' \in \mathbb{V}_{\boldsymbol{x}} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$
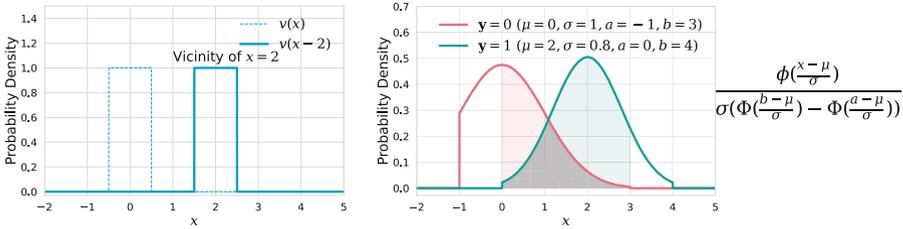
(a) Vicinity function $v(\boldsymbol{x})$ is shown in blue ($\epsilon = 0.5$). To get the vicinity at a specific input $\boldsymbol{x} = 2$, we simply shift $v(\boldsymbol{x})$ along the positive direction of the x-axis by 2.

(b) The grey-shaded area represents the Bayes error, characterising the overlap between the two distributions (truncated normal as expressed). The union of the light colour-shaded area and the grey-shaded area represents the proportion of inputs that *have* uncertainty, *i.e.*, $p(\mathbf{x} \in \mathbb{K}_D)$.

Fig. 2. 1D visualizations of vicinity function and Bayes error. A vicinity function is a rectangular function that returns a constant value if an input is in the vicinity. We use two PDFs of the truncated normal distribution to visualise the Bayes error.

Essentially, Equation (3) can be viewed as a probability density function uniformly defined over the vicinity around an input $\boldsymbol{x}$. Now we shift the x-coordinate by $\boldsymbol{x}$, we get

$$v_{\mathbf{0}}(\boldsymbol{x}' - \boldsymbol{x}) = \begin{cases} \left(\int_{\mathbb{V}_{\mathbf{0}}} d\boldsymbol{x}'' \right)^{-1}, & \text{if } \boldsymbol{x}' - \boldsymbol{x} \in \mathbb{V}_{\mathbf{0}} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Assuming that the vicinity function is translation invariant, we can drop the subscript $\mathbf{0}$, and use a positive constant ϵ_v to represent $\int_{\mathbb{V}_{\mathbf{0}}} d\boldsymbol{x}''$. Thus, the vicinity function $v : \mathbb{X} \to \{ 0, \epsilon_v^{-1} \}$ can be expressed as

$$v(\boldsymbol{x}) = \begin{cases} \epsilon_v^{-1} & \text{if } \boldsymbol{x} \in \mathbb{V}_{\mathbf{0}} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Since these representations are equivalent, we choose either representation based on the contexts. An example of a one-dimensional input's vicinity is shown in Fig. 2a.

Achieving robustness is challenging. Verifying whether $\text{Rob}\left(h, \boldsymbol{x}, y; \mathbb{V}_{\boldsymbol{x}}\right)$ holds for a given classifier h is complicated since examining every example within a vicinity is impractical. Consequently, accurately estimating a classifier's robustness on specific inputs, as well as its robustness on a given data distribution, presents significant challenges. Existing methods for evaluating robustness include empirical evaluation (*i.e.*, adversarial attacks) [14], robustness verification [17,28], and others [57].

Adversarial attacks take one or more steps to search for adversarial examples within a vicinity. Let $\text{AttS}\left(h, \boldsymbol{x}, y; \mathbb{V}\boldsymbol{x}\right)$ denote the **s**uccess of an **att**ack in finding adversarial examples in $\mathbb{V}_{\boldsymbol{x}}$. The failure rate of this attack on classifier h can serve as an estimation for the classifier's expected robustness, as outlined below.

$$\text{AttS}\left(h, \boldsymbol{x}, y; \mathbb{V}\boldsymbol{x}\right) \to \neg\,\text{Rob}\left(h, \boldsymbol{x}, y; \mathbb{V}_{\boldsymbol{x}}\right) \tag{6}$$

Another perspective contends that any non-zero rate of false negatives in the detection of adversarial examples is problematic. To this end, a condition Vrob is to be established, such that given a classifier h, it satisfies

$$\forall \, (\boldsymbol{x}, y) \in \mathbb{X} \times \mathbb{Y}. \; \text{Vrob}\left(h, \boldsymbol{x}, y; \mathbb{V}_{\boldsymbol{x}}\right) \rightarrow \text{Formula 2} \tag{7}$$

This method refers to robustness verification and the condition Vrob is the **v**erification result of **rob**ustness. There are two categories of robustness verification methods, *i.e.*, incomplete deterministic verification [28] and complete deterministic verification [28]. Any deterministic verification method that fulfils Formula 7 qualifies as an incomplete verification. If a method further fulfils Formula 8, it qualifies as a complete verification. In both cases, if the verification result $\text{Vrob}\left(h, \boldsymbol{x}, y; \mathbb{V}_{\boldsymbol{x}}\right)$ is True, *i.e.*, verified, the classifier is considered to have deterministic robustness certification [28] for input $\boldsymbol{x}$ within the vicinity $\mathbb{V}_{\boldsymbol{x}}$, and the average certification likelihood is often called certified robust accuracy [45]. Certified robust accuracy can serve as a lower bound for the classifier's expected robustness.

$$\forall \, (\boldsymbol{x}, y) \in \mathbb{X} \times \mathbb{Y}. \; \text{Vrob}\left(h, \boldsymbol{x}, y; \mathbb{V}_{\boldsymbol{x}}\right) \leftarrow \text{Rob}\left(h, \boldsymbol{x}, y; \mathbb{V}_{\boldsymbol{x}}\right) \tag{8}$$

These verification methods can be used to optimise classifiers during training, and such a practice refers to certified training [28,53], which is defined as follows.

$$\min_{h} \; \mathrm{E}_{(\mathbf{x},\mathbf{y}) \sim D} \left[\sup_{\boldsymbol{x}' \in \mathbb{V}(\mathbf{x}), \; k \neq \mathbf{y}} \left(\ell(h, \boldsymbol{x}', \mathbf{y}) - \ell(h, \boldsymbol{x}', k) \right) \right] \tag{9}$$

Here, the neural network verification methods are used to soundly approximate the worst loss that can be induced by any perturbation within the vicinity of each training sample. However, after years of research [6,46,64], certified training still faces challenges. Existing certified training methods often result in a significant drop in the model's accuracy [10,40]. For instance, the best accuracy achieved by certified training is typically half of that of the standard training on the CIFAR-10 data set [45,51]. Such a significantly reduced accuracy often means that the model is unacceptable in practice.

In summary, to evaluate whether h attains robustness at example $(\boldsymbol{x}, y)$ within the vicinity $\mathbb{V}_{\boldsymbol{x}}$, existing methods include checking $\text{AttS}\left(h, \boldsymbol{x}, y; \mathbb{V}\boldsymbol{x}\right)$ through adversarial attacks or $\text{V}\left(h, \boldsymbol{x}, y; \mathbb{V}_{\boldsymbol{x}}\right)$ through robustness verification. The expected robustness over a given distribution D, denoted by

$$\mathrm{E}_{(\mathbf{x},\mathbf{y}) \sim D} \left[\mathbf{1}_{\text{Rob}\left(h, \mathbf{x}, \mathbf{y}; \mathbb{V}_{\mathbf{x}}\right)} \right] \tag{10}$$

which can be overestimated by attack success rate ($\mathrm{E}_{(\mathbf{x},\mathbf{y}) \sim D}\left[\mathbf{1}_{\text{AttS}}\right]$) or underestimated by certified robust accuracy ($\mathrm{E}_{(\mathbf{x},\mathbf{y}) \sim D}\left[\mathbf{1}_{\text{Vrob}}\right]$). $\mathbf{1}_{\text{condition}}$ is the indicator function that returns 1 if the condition is True, and 0 otherwise.

2.2 Bayes Rules for Distributions

In the following, we introduce the notion of Bayes Error and how it reflects a classification distribution. We consider a scenario where an input $\mathbf{x}$ is to be classified into one

class in $\mathbb{Y}$, in particular, $\mathrm{y} = k$ with prior class probability $P(\mathrm{y} = k)$ where $k \in \mathbb{Y}$. Let $p(\mathbf{x}|\mathrm{y} = k)$ denote the class likelihood, that is, the conditional probability density of $\mathbf{x}$ given that it belongs to class k. The probability that the input $\mathbf{x}$ belongs to a specific class k, namely the posterior probability $p(\mathrm{y} = k|\mathbf{x})$, is given by Bayes' theorem.

$$p(\mathrm{y} = k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathrm{y} = k)P(\mathrm{y} = k)}{p(\mathbf{x})} \tag{11}$$

where $p(\mathbf{x})$ is the probability density function of $\mathbf{x}$, *i.e.*, $p(\mathbf{x}) = \sum_{k \in \mathbb{Y}} p(\mathbf{x}|\mathrm{y} = k)P(\mathrm{y} = k)$. This classifier assigns an input $\mathbf{x}$ to the class with the highest posterior and is called the Bayes classifier, which is the optimal classifier. The classification error associated with the Bayes classifier is outlined as follows.

Definition 4 (Bayes error). *Given a distribution D over $\mathbb{X} \times \mathbb{Y}$, the error associated with the Bayes classifier is called the Bayes error (rate), denoted as β_D. The Bayes error can be expressed [13, 15] as:*

$$\begin{aligned} \beta_D &= \mathrm{E}_{(\mathbf{x},\mathrm{y}) \sim D} \left[1 - \max_k p(\mathrm{y} = k|\mathbf{x}) \right] \\ &= \int \left(1 - \max_k p(\mathrm{y} = k|\mathbf{x} = \boldsymbol{x}) \right) p(\boldsymbol{x}) d\boldsymbol{x} \end{aligned} \tag{12}$$

Besides, since the Bayes classifier is optimal [42], this optimality gives rise to the following definition of the Bayes error [34].

$$\beta_D = \min_{\text{measurable } h} \mathrm{E}_{(\mathbf{x},\mathrm{y}) \sim D} \left[\mathbf{1}_{h(\mathbf{x}) \neq \mathrm{y}} \right] \tag{13}$$

where the Bayes error is defined as the minimum of the errors achieved by measurable functions $h : \mathbb{X} \to \mathbb{Y}$. Hereby, (any) classifier h with an error rate equal to β_D can be called a Bayes classifier.

An example illustrating the Bayes error is given in Fig. 2b. The Bayes error fundamentally reflects the inherent uncertainty in classification tasks. It is the (irreducible) minimal error rate achievable by any classifier for a specific problem, influenced by the overlap amount among the class probability distributions. An input having a certain (deterministic) label can be formally expressed as $\max_k p(\mathrm{y} = k|\mathbf{x} = \boldsymbol{x}) = 1$. We can also represent this using the ceiling $\lceil \cdot \rceil$ or floor $\lfloor \cdot \rfloor$ function within the interval $[0, 1]$. Specifically, the ceiling function returns the smallest integer greater than or equal to the input. Consequently, it returns 1 for any number from 0 (exclusive) up to 1 and returns 0 if the input is 0. This shows that the input's label has uncertainty if $1 - \max_k p(\mathrm{y} = k|\mathbf{x} = \boldsymbol{x}) > 0$ and does not have uncertainty otherwise. We write $\mathbb{K}_D = \{ \boldsymbol{x} \mid 1 - \max_k p(\mathrm{y} = k|\mathbf{x} = \boldsymbol{x}) > 0 \}$ to denote the set of every input whose label has uncertainty. The Bayes error provides a yardstick for other classifiers [18,42], *e.g.*, a classifier may be deemed effective if its error rate approximates the Bayes error.

As highlighted in Eqs. (11) and (12), the calculation of Bayes error is contingent upon knowing the prior distribution. In practical situations, since this distribution is not analytically known, the strategy is to estimate Bayes error using the observable portion of the distribution, *e.g.*, training data characteristics, through approximations [11,19,52,61] or by computing its upper [5,13,21] and lower [1,65,67] bounds.
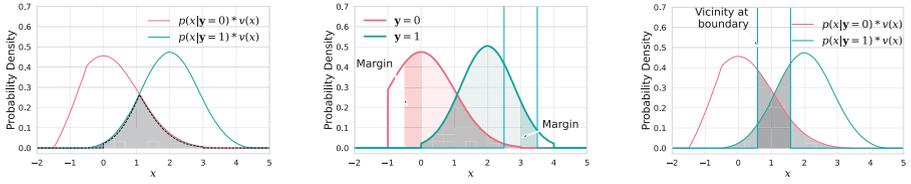
(a) Convolution of distributions from Fig. 2b with the vicinity function from Fig. 2a, resulting in flatter and smoother distributions with increased shaded areas.

(b) Demonstration of the margin contributing to $\zeta_D^\sharp$. In a 1D problem, the width of $\mathbb{K}_D^*$ is essentially the vicinity width.

(c) Union of shaded areas indicating the lower bound of irreducible robustness error and upper bound of certified robust accuracy.

Fig. 3. Visualizing the convolution of distributions, the marginal contribution to Bayes error, and the bounds of robustness error and certified robust accuracy.

Problem Definition. Next, we define the problem that we study. Despite the many proposals on certified training, noticeable suboptimality in robustness persists, especially compared with vanilla accuracy. Our objective is to ascertain whether this limitation comes from insufficient optimisation, or if there exists a fundamental upper bound that inherently limits the certified robust accuracy. Furthermore, if such an upper bound does exist, we aim to investigate how we can compute it, and how we can validate our result.

3 An Upper Bound of Robustness from Bayes Error

In this section, we present a method that attempts to address our research problem defined above, from a Bayes error perspective. Particularly, we hypothesise that the Bayes error plays a vital role in estimating the robustness that can be achieved by any classifier. First, we prove that certified training increases the Bayes error, which poses an upper bound on the robustness that can be achieved by any classifier. Second, we present how the upper bounds of certified robust accuracy can be calculated from a given distribution.

3.1 Certified Training Increases Bayes Error

Certified robustness can be viewed as a way of optimizing the classifier with an altered data distribution instead of the original distribution [37]. This is because due to the requirement of robustness, an input may be forced with a label of some of its neighbors in the vicinity, instead of its original label. In the following, we investigate how the robustness requirement influences the data distribution, further affecting the Bayes error. We hypothesise that the altered distribution worsens Bayes error. We begin by defining a "label-assignment" action that alters a distribution, from a local perspective.

Suppose there is a distribution D over the space $\mathbb{X} \times \mathbb{Y}$. From a local (example) perspective, an example $(\boldsymbol{x}, y) \in \mathbb{X} \times \mathbb{Y}$ assigns its label to a specific domain $\mathbb{S} \subset \mathbb{X}$ ($\mathbb{S}$ can be a vicinity) by directly altering the joint probability in $\mathbb{S}$. Specifically, this

alteration is a process that adds $\Delta p(\mathrm{y} = y, \mathbf{x} = \boldsymbol{x})$ to the original $p(\mathrm{y} = y, \mathbf{x} = \boldsymbol{x})$, and adds $\Delta p(\mathrm{y} = y, \mathbf{x} = \boldsymbol{x}')$ to the original $p(\mathrm{y} = y, \mathbf{x} = \boldsymbol{x}')$ for any example $\boldsymbol{x}' \in \mathbb{S}$, where Δp denotes a change in the joint distribution function (of $\mathbf{x}$, $\mathbf{y}$) such that

$$
\Delta p(\mathrm{y} = k, \mathbf{x} = \boldsymbol{x}) = \begin{cases} 0, & \text{if } k \neq y \\ p_{\mathrm{ori}}(\mathrm{y} = k, \mathbf{x} = \boldsymbol{x}) \left(\frac{1}{\int_{\mathbb{S}} d\boldsymbol{x}'} - 1 \right), & \text{otherwise} \end{cases}
$$

$$
\Delta p(\mathrm{y} = k, \mathbf{x} = \boldsymbol{x}') = \begin{cases} 0, & \text{if } k \neq y \\ \frac{1}{\int_{\mathbb{S}} d\boldsymbol{x}'} p_{\mathrm{ori}}(\mathrm{y} = k, \mathbf{x} = \boldsymbol{x}), & \text{otherwise} \end{cases}
$$

$$(14)$$

We then explain why this label assignment aligns with the robustness criteria. In the context of robustness, for every input in the training set, every neighbour point in the vicinity around the input gets the label of this input. Meanwhile, an input may fall within more than one vicinity. Thus, an input gets labels assigned from multiple neighbours, and each label's influence depends on its source's joint probability. Intuitively, examples with higher joint probability have a stronger influence on its vicinity.

Equation (14) captures the effect of an input's label on its neighbours, from an individual input perspective. We next set out from a distributional perspective which is supposed to match our individual input perspective. When all examples in the original distribution concurrently assign labels to their respective vicinity, the effect is equivalent to convolving this given distribution with the vicinity (function). This convolved distribution represents the target of certified robustness optimization, as captured by Throrem 1.

Theorem 1. *Given a distribution D for classification, optimising for higher certified robustness does not optimise the classifiers to fit D. Rather, it optimises classifiers towards $D * v$, i.e., convolved distribution between D and vicinity $v(\boldsymbol{x})$.*

Proof. Optimising for certified robustness tunes classifiers to have a higher probability of satisfying Formula 2. Therefore, the objective is to maximise

$$
\mathrm{E}_{(\mathbf{x},\mathbf{y}) \sim D} \left[\left\lfloor \int_{\mathbb{X}} v(\mathbf{x} - \boldsymbol{x}') \cdot \mathbf{1}_{\mathrm{y}=h(\boldsymbol{x}')} d\boldsymbol{x}' \right\rfloor \right]
$$
$$
= \sum_{k} \int_{\mathbb{X}} \left\lfloor \int_{\mathbb{X}} v(\boldsymbol{x} - \boldsymbol{x}') \cdot \mathbf{1}_{k=h(\boldsymbol{x}')} d\boldsymbol{x}' \right\rfloor p(\boldsymbol{x}, k) d\boldsymbol{x}
$$

$$(15)$$

Denote $\mu_k(\boldsymbol{x}) = \int_{\mathbb{X}} v(\boldsymbol{x} - \boldsymbol{x}') \cdot \mathbf{1}_{k=h(\boldsymbol{x}')} d\boldsymbol{x}'$, then the objective can be expressed as $\sum_k \int_{\mathbb{X}} \lfloor \mu_{k^*}(\boldsymbol{x}) \rfloor p(\boldsymbol{x}, k) d\boldsymbol{x}$. Suppose μ_k for each $\boldsymbol{x}$ is the variational function we tune to maximise the objective. As the floor function is monotonically increasing, maximising the original objective is equivalent to maximising $\sum_k \int_{\mathbb{X}} \mu_k(\boldsymbol{x}) p(\boldsymbol{x}, k) d\boldsymbol{x}$, which equals

$$
\sum_{k} \int_{\mathbb{X}} \int_{\mathbb{X}} v(\boldsymbol{x} - \boldsymbol{x}') \cdot \mathbf{1}_{k=h(\boldsymbol{x}')} d\boldsymbol{x}' p(\boldsymbol{x}, k) d\boldsymbol{x}
$$
$$
= \sum_{k} \int_{\mathbb{X}} \int_{\mathbb{X}} v(\boldsymbol{x}' - \boldsymbol{x}) p(\boldsymbol{x}, k) d\boldsymbol{x} \cdot \mathbf{1}_{k=h(\boldsymbol{x}')} d\boldsymbol{x}'
$$

$$(16)$$

Observe the convolution form $(p_k * v)(\boldsymbol{x}') = \int_{\mathbb{X}} v(\boldsymbol{x}' - \boldsymbol{x})p(\boldsymbol{x}, k)d\boldsymbol{x}$, and the objective becomes $\sum_k \int_{\mathbb{X}} (p_k * v)(\boldsymbol{x}') \cdot \mathbf{1}_{k=h(\boldsymbol{x}')}d\boldsymbol{x}'$. Thus, the target distribution of optimising for certified robustness is indeed the convolved distribution of the given one.   □

Note that Throrem 1 is not particularly for existing certified training approaches but rather any approach to achieving certified robustness. Hereafter, we use D to denote the original distribution, p to denote the conditional distributions (of each class) from D, D' to denote the convolved distribution, and q to denote the conditional distributions (of each class) from D'. Thus, the Bayes error of D' can be expressed as

$$\beta_{D'} = \mathrm{E}_{(\mathbf{x},\mathbf{y}) \sim D'} \left[1 - \max_k q(\mathbf{y} = k|\mathbf{x}) \right] \tag{17}$$

The subsequent question is to study the Bayes error with respect to the convolved distribution D'. Throrem 2 suggests that Bayes error grows when D is transformed into D', as illustrated in Fig. 3a.

Theorem 2. *Given a distribution D for classification, its convolved distribution D' has an equal or larger Bayes error, i.e., $\beta_D \leq \beta_{D'}$.*

Proof. Consider D is a distribution of random variables $\mathbf{x}$ and $\mathbf{y}$. Let $p_k(\boldsymbol{x})$ be the conditional distribution of $\mathbf{x}$ given $\mathbf{y} = k$. We need to prove that the Bayes error between p_k is less than or equal to the Bayes error between $p_k * v$, where v is a probability density function (PDF). First, let us prove that $((\max_k(p_k)) * v)(\boldsymbol{x}) \geq \max_k((p_k * v)(\boldsymbol{x}))$. Expanding both sides, we get $\int \max_k(p_k(\boldsymbol{x} - \boldsymbol{x}')v(\boldsymbol{x}'))d\boldsymbol{x}'$ at left. We get $\max_k(\int p_k(\boldsymbol{x} - \boldsymbol{x}')v(\boldsymbol{x}')d\boldsymbol{x}')$ at right. We can see that left $\geq \int p_k(\boldsymbol{x} - \boldsymbol{x}')v(\boldsymbol{x}')d\boldsymbol{x}'$ for any k. Therefore, the maximum can be brought out from the integral and thus the left side is proved to be greater than or equal to the right side. Then, we use the equality that the integral of $((\max_k p_k) * v)(\boldsymbol{x})$ is actually the same as the integral of $(\max_k p_k)(\boldsymbol{x})$. This is because v itself is a PDF. Therefore, we get

$$\int \left(1 - (\max_k p_k)(\boldsymbol{x}) \right) d\boldsymbol{x} \leq \int \left(1 - \max_k((p_k * v)(\boldsymbol{x})) \right) d\boldsymbol{x} \tag{18}$$

□

Intuitively, when robustness is required, new labels are assigned to data in the vicinity of the training inputs. But, these new labels sometimes contradict the original labels or contradict themselves. As a result, the convolved distribution invariably exhibits larger uncertainty, represented by an increased Bayes error. For instance, let us consider a separable distribution with a unique boundary. The condition $d(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon$ implies that $\boldsymbol{x}$ and $\boldsymbol{x}'$, near the boundary, should be assigned the same label even if their ground-truth labels are different, leading to a non-zero Bayes error.

3.2   Irreducible Robustness Error and Robustness Upper Bound

We have proved that the optimal robustness is equal to or lower than the optimal accuracy. We now would like to find a quantitative upper bound for robustness. We first

define the irreducible expected error rate across all classifiers regarding robustness, as expressed in Equation (19) where $\zeta_D^\sharp$ represents the irreducible robustness error on distribution D.

$$\zeta_D^\sharp = \inf_{\text{measurable } h} E_{(\mathbf{x},\mathbf{y}) \sim D} \left[1 - \mathbf{1}_{\text{Rob}\left(h,\mathbf{x},\mathbf{y};\mathbb{V}_\mathbf{x}\right)} \right] \tag{19}$$

This concept is analogous to Equation (13), where the Bayes error is described as the irreducible vanilla error rate achievable by any classifier. Then, the upper bound of expected robustness is 1 minus the lower bound of $\zeta_D^\sharp$.

Recall Definition 3, the condition $\text{Rob}(h, \boldsymbol{x}, y; (d, \epsilon))$ holds only if Formula 2 is met, where $(\boldsymbol{x}, y) \in \mathbb{X} \times \mathbb{Y}$. Nevertheless, Formula 2 alone is not a sufficient condition for $\text{Rob}\left(h, \boldsymbol{x}, y; (d, \epsilon)\right)$. According to Definition 2, $\text{Rob}\left(h, \boldsymbol{x}, y; (d, \epsilon)\right)$ also requires that no input in the vicinity of $\boldsymbol{x}$ should be classified incorrectly, as expressed in Formula 20. Formally, $\text{Rob}\left(h, \boldsymbol{x}, y; (d, \epsilon)\right) \iff$ Formula 2 $\wedge$ Formula 20.

$$\neg \exists \left(\boldsymbol{x}', y'\right) \in \mathbb{X} \times \mathbb{Y}. \, p(\boldsymbol{x}', y') > 0 \, \wedge \, d(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon \, \wedge \, h(\boldsymbol{x}') \neq y' \tag{20}$$

Equation (20) suggests that if the ground-truth labels of inputs in the vicinity of $\boldsymbol{x}$ are different from the labels of $\boldsymbol{x}$, then a prerequisite of robustness is missing such that robustness cannot be attained. The conjunction of Formula 2 and 20 clarifies that robustness asks for general correctness across the (local) input domain, rather than just local consistency. From this conjunction, we can derive that for a classifier to attain robustness at an input, it is necessary that the posterior probability associated with this input is entirely certain, which is formally captured in Throrem 3. Further, given a distribution, the proportion of examples with uncertain labels can serve as a lower bound for the proportion of examples without robustness.

Theorem 3. *Given a distribution D over $\mathbb{X} \times \mathbb{Y}$, the irreducible robustness error is greater than or equal to the probability that an input is in $\mathbb{K}_D$.*

$$\zeta_D^\sharp \geq \int \left\lceil 1 - \max_k p(\mathbf{y} = k | \mathbf{x} = \boldsymbol{x}) \right\rceil p(\boldsymbol{x}) d\boldsymbol{x} \geq \beta_D \tag{21}$$

When $\left\lceil 1 - \max_k p(\mathbf{y} = k | \mathbf{x} = \boldsymbol{x}) \right\rceil = 0$, there is one and only one class has a posterior probability of 1 at input $\boldsymbol{x}$, resulting in a non-zero contribution to the Bayes error.

Proof. Assume some classifier h attains robustness at input $\boldsymbol{x}$, and the posterior probability is not certain, *i.e.*, $1 - \max_k p(\mathbf{y} = k \mid \mathbf{x} = \boldsymbol{x}) > 0$. The latter infers that there exists some (non-zero probability) examples of $(\boldsymbol{x}, y_1)$ pair and some $(\boldsymbol{x}, y_2)$ pair, and $y_1 \neq y_2$. The prediction for $\boldsymbol{x}$ differs from at least one of either y_1 and y_2. Formally, the latter condition in our assumption entails $(h(\boldsymbol{x}) \neq y_1 \vee h(\boldsymbol{x}) \neq y_2)$, which then entails

$$\begin{aligned} \exists \boldsymbol{x}' \in \mathbb{V}_{\boldsymbol{x}}. \, h(\boldsymbol{x}') \neq y_1 \vee h(\boldsymbol{x}') \neq y_2 & \quad (\text{because } \boldsymbol{x} \in \mathbb{V}_{\boldsymbol{x}}) \\ \exists \boldsymbol{x}' \in \mathbb{V}_{\boldsymbol{x}} \, \exists y' \in \mathbb{Y}. \, p(\boldsymbol{x}', y') > 0 \, \wedge \, h(\boldsymbol{x}') \neq y' & \quad (\text{Disjunction Elimination}) \end{aligned} \tag{22}$$

Condition (22) contradicts the former condition in our assumption, *i.e.*, Condition 20. Thus, robustness may only be attained if there is no label uncertainty at an input. □

Uncertainty contributes to an irreducible error in both vanilla accuracy and robustness. The irreducible robustness error is at least the Bayes error. We are further interested in refining this boundary in scenarios where we know the value of the Bayes error but lack information about the posterior probabilities. To this end, we develop Corollary 1.

Corollary 1. *Given a distribution D over $\mathbb{X} \times \mathbb{Y}$, its irreducible robustness error is at least as large as the Bayes error multiplied by the number of classes divided by one less than the number of classes,* i.e.,

$$\zeta_D^\sharp \geq \frac{|\mathbb{Y}|}{|\mathbb{Y}| - 1} \beta_D \tag{23}$$

where $|\mathbb{Y}|$ denotes the number of classes.

Proof. we have that $\zeta_D^\sharp \geq \int_{\mathbb{K}_D} p(\boldsymbol{x}) d\boldsymbol{x} = \int_{\mathbb{K}_D}(1 - \max_k p(\mathbf{y} = k|\mathbf{x} = \boldsymbol{x}) + \max_k p(\mathbf{y} = k|\mathbf{x} = \boldsymbol{x})) p(\boldsymbol{x}) d\boldsymbol{x} = \beta_D + \int_{\mathbb{K}_D}(\max_k p(\mathbf{y} = k|\mathbf{x} = \boldsymbol{x})) p(\boldsymbol{x}) d\boldsymbol{x} \geq \int_{\mathbb{K}_D} p(\boldsymbol{x})/|\mathbb{Y}| d\boldsymbol{x} + \beta_D$. Thus, we can prove that $\int_{\mathbb{K}_D} p(\boldsymbol{x}) d\boldsymbol{x} \geq \beta_D |\mathbb{Y}|/(|\mathbb{Y}| - 1)$ $\square$

In Throrem 3 and Corollary 1, the lower bounds for the $\zeta_D^\sharp$ are established based on that a single input needs to have a deterministic label. Still, there are additional conditions that, if unmet, will prevent a classifier from attaining robustness for a given input. For instance, we can expand the certainty requirement from a single input to encompass any input within its vicinity. The input neighbours in the vicinity with uncertain labels can also contribute to the irreducible robustness error. Given an input $\boldsymbol{x}$ such that $\boldsymbol{x} \notin \mathbb{K}_D$, if there exists an $\boldsymbol{x}'$ within this vicinity of $\boldsymbol{x}$ such that $\boldsymbol{x}' \in \mathbb{K}_D$, robustness at $\boldsymbol{x}$ cannot be attained. All such $\boldsymbol{x}$ forms a domain $\mathbb{K}^*{}_D$. $\mathbb{K}^*{}_D$ can be considered as a thin margin around $\mathbb{K}_D$, as shown in Fig. 3b. This expansion results in a more stringent condition. Consequently, we will likely identify a tightened lower bound for $\zeta_D^\sharp$.

Corollary 2. *Given a distribution D over $\mathbb{X} \times \mathbb{Y}$, then*

$$\zeta_D^\sharp \geq 2 \cdot \epsilon_{\text{eff}} \cdot p_{\min} \cdot \left(\int_{\mathbb{K}_D} d\boldsymbol{x} \right)^{\frac{\dim \mathbb{X} - 1}{\dim \mathbb{X}}} + \int_{\mathbb{K}_D} p(\boldsymbol{x}) d\boldsymbol{x} \tag{24}$$

where ϵ_{eff} denotes the radius of the vicinity according to the definition of robustness, e.g., for L^2-perturbation, ϵ_{eff} equals to the radius ϵ. For general perturbations,

$$\frac{\pi^{\dim \mathbb{X}/2}}{\Gamma\left(\frac{\dim \mathbb{X}}{2} + 1\right)} \epsilon_{\text{eff}}^{\dim \mathbb{X}} = \int_{\mathbb{X}} \lceil v(\boldsymbol{x}) \rceil d\boldsymbol{x} \tag{25}$$

Proof. $\mathbb{K}_D^*$ emerges when a perturbation vicinity sweeps along the boundary of $\mathbb{K}_D$ and the isoperimetric inequality suggests that the volume of this marginal domain is minimized when both vicinity and $\mathbb{K}_D$ are $\dim \mathbb{X}$-spheres.

Thus, a lower bound of the volume of $\mathbb{K}_D^*$ can be expressed as the volume difference between two concentric spheres which is again greater than the product of their radius

difference and the surface area of the inner sphere. Thus, the volume of $\mathbb{K}_D^*$ is lower bounded by

$$\epsilon_{\text{eff}} \cdot \text{vol}(\mathbb{K}_D)^{\frac{\dim \mathbb{X}-1}{\dim \mathbb{X}}} \cdot \frac{2\pi^{\dim \mathbb{X}/2}}{\Gamma\left(\frac{\dim \mathbb{X}}{2}\right)} \left(\pi^{-\dim \mathbb{X}/2}\Gamma\left(\frac{\dim \mathbb{X}}{2}+1\right)\right)^{\frac{\dim \mathbb{X}-1}{\dim \mathbb{X}}} \tag{26}$$

where Γ represents the gamma function, and for all positive real numbers $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}\,dt$. We further simplify Equation (26) to $\epsilon_{\text{eff}} \cdot \text{vol}(\mathbb{K}_D)^{(\dim \mathbb{X}-1)/(\dim \mathbb{X})} \cdot \gamma$, where $\gamma \geq 2$ for $\dim \mathbb{X} > 1$ and a lower bound of $\text{vol}(\mathbb{K}_D^*)$ is thus $2 \cdot \epsilon_{\text{eff}} \cdot (\int_{\mathbb{K}_D} d\boldsymbol{x})^{(\dim \mathbb{X}-1)/(\dim \mathbb{X})}$. In very high dimensions, the (minimum) volume of $\mathbb{K}_D^*$ is almost linearly related to the volume of $\mathbb{K}_D$. The irreducible error contributed by the marginal domain to robustness can be expressed as $\int_{\mathbb{K}_D^*} p(\boldsymbol{x})d\boldsymbol{x}$. It is greater than or equal to $\int_{\mathbb{K}_D^*} p_{\min}d\boldsymbol{x}$, where $p_{\min} = \min_{\boldsymbol{x}} p(\boldsymbol{x})$. Thus, this irreducible error $\geq p_{\min} \cdot \text{vol}(\mathbb{K}_D^*)$, contributes to the irreducible robustness error as the first term in Equation (24). This corollary is particularly useful if we know the non-zero $p_{\min}$ of the distribution. □

In short, Throrem 3, Corollary 1, and Corollary 2 suggest how we can get lower bounds of irreducible robustness error $\zeta_D^\sharp$ from the original distribution D, with lower bound from Corollary 2 being the tightest among three. Given a distribution, $\zeta_D^\sharp$ has two sources. One is the examples that have uncertain labels (which contribute to the error directly), and the other is the examples that have neighbours whose labels are uncertain (which contribute to the error indirectly). Additionally, when the Bayes error β_D of a distribution is non-zero, the irreducible error of robustness $\zeta_D^\sharp$ is also non-zero and is greater than the Bayes error.

Theoretically, there is another way to tighten the bound provided by Throrem 3. If we know the convolved distribution D' obtained in Sect. 3.1, the $\zeta_D^\sharp$ can be calculated as $p(\mathbf{x} \in \mathbb{K}_{D'})$, *i.e.*, the probability (in convolved distribution) that input has a deterministic label. Thus,

$$\zeta_D^\sharp = \text{E}_{(\mathbf{x},\mathbf{y})\sim D'}\left[1 - \lfloor \max_k q(\mathbf{y} = k|\mathbf{x})\rfloor\right] \tag{27}$$

Since $D' = D * v$, as vicinity size grows, the Bayes error of D' also grows, and thus the irreducible robustness error $\zeta_D^\sharp$ grows.

The least upper bound of robustness on a given data distribution D can then be written as $1 - \zeta_D^\sharp$, and 1 minus any lower bound of $\zeta_D^\sharp$ presented above serves as an upper bound of robustness on a given data distribution D. These upper bounds are directly derived from the data distribution D and the vicinity function v, independent of any specific classifier.

Although we have been using both Formulae 2 and (20) throughout this subsection, the existing studies only rely on Formula 2 [28] for practical evaluation of certified robust accuracy. Intuitively, we sometimes do not know the true label of a neighbour $\boldsymbol{x}'$ in an input $\boldsymbol{x}$'s vicinity, and thus use the $\boldsymbol{x}$'s label instead. Consequently, the correctness of $\boldsymbol{x}'$ prediction is neglected. Instead, only the consistency between predictions on $\boldsymbol{x}'$ and $\boldsymbol{x}$, as well as the correctness of prediction on $\boldsymbol{x}$, are considered. This simplification could result in a different certified robust accuracy for classifiers and exceed our

upper bounds of robustness on a given data distribution (Throrem 3). To this end, we also present the irreducible robustness error ζ_D in Equation (28) and the corresponding upper bound for such robustness on a given data distribution $1 - \zeta_D$. We use Fig. 3c to illustrate its effect.
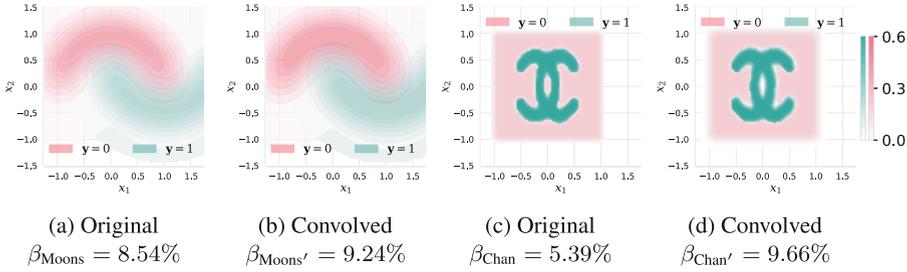
$$\zeta_D = \int_{\mathbb{K}_{D^\dagger}} q(\boldsymbol{x})d\boldsymbol{x} + \int_{\mathbb{X}\backslash\mathbb{K}_{D^\dagger}} \left(1 - \max_k q(\mathbf{y} = k|\mathbf{x} = \boldsymbol{x})\right) q(\boldsymbol{x})d\boldsymbol{x} \qquad (28)$$

where $D^\dagger$ is a distribution obtained from convolving the vicinity function v and the "hardened" distribution of D', *i.e.*, each $p_{\text{hard}}(\mathbf{y} = k_{\max}|\mathbf{x} = \boldsymbol{x}) = q(\boldsymbol{x})$ and for other $k \neq k_{\max}$, $p_{\text{hard}}(\mathbf{y} = k|\mathbf{x} = \boldsymbol{x}) = 0$. Then, $p_{D^\dagger} = p_{\text{hard}} * v$. Recall that q is the conditional distribution of D'. In Equation (28), its first term suggests no input close to the boundary can attain robustness. For inputs not close to the boundary, as indicated by the second term, their optimal robustness on a given data distribution depends on the correctness of the prediction. In terms of Fig. 3c, the first term corresponds to the shaded area bounded by the vicinity, and the second term corresponds to all shaded areas outside the curve. Although Equation (28) has tackled the label-missing challenges in practice, this theoretical evaluation of the irreducible error (in certified robust accuracy) could still rely on the knowledge of distribution. Thus, distribution estimating techniques are also needed when facing sampled data from an unknown distribution.

4 Experiment and Results

In this section, we empirically test our results discussed above by designing and answering three research questions: 1) does certified training always result in a classifier on a distribution with a higher Bayes error; 2) is our computed upper bound of robustness indeed higher than the robustness achieved by all the existing certified training classifiers; and 3) does the upper bound of robustness change when the vicinity increases, and if so how does it change?

The experiments are conducted with four data sets: two synthetic ones (*i.e.*, Moons and Chan [8]) and two standard benchmarks (*i.e.*, FashionMNIST [59] and CIFAR-10 [23]). Moons is used for binary classification with two-dimensional features, where each class's distribution is described analytically with specific likelihood equations, and uses a three-layer Multi-Layer Perceptron (MLP) neural network for classification. The Chan data set, also for binary classification with two-dimensional features, differs in that it does not follow a standard PDF pattern, requiring kernel density estimation (KDE) for non-parametric PDF estimation, and also uses the three-layer MLP. FashionMNIST, a collection of fashion item images, involves a 10-class classification task with 784-dimensional inputs (28×28 pixel grayscale images). Each class has an equal prior probability, and their conditional distributions are estimated non-parametrically using KDE. CIFAR-10 uses images with a resolution of 32×32 pixels. Similar to FashionMNIST, it has a balanced class distribution and is estimated using KDE. We use a seven-layer convolutional neural network (CNN-7) [45] as the classifier of both FashionMNIST and CIFAR-10. We adopt a direct approach [20] to compute the original Bayes error of both FashionMNIST and CIFAR-10 [20].

(a) Original
$\beta_{\text{Moons}} = 8.54\%$

(b) Convolved
$\beta_{\text{Moons}'} = 9.24\%$

(c) Original
$\beta_{\text{Chan}} = 5.39\%$
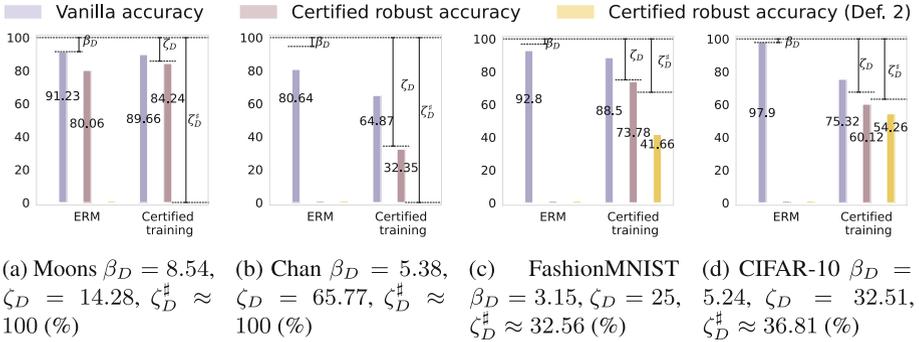
(d) Convolved
$\beta_{\text{Chan}'} = 9.66\%$

Fig. 4. The conditional distribution before and after convolution for (a, b) Moons and (c, d) Chan. For both Moons and Chan, L^∞ size is set at $\epsilon = 0.15$. We also report the Bayes error to show the change of inherent uncertainty in each distribution.

To train the classifiers, two approaches are adopted, *i.e.*, empirical error minimization (ERM [54]) for standard training, and the state-of-the-art (SOTA) small-box method for certified training [35]. The performance of these classifiers is evaluated using two metrics: vanilla accuracy and certified robust accuracy [35]. Note that, certified robust accuracy measures the proportion of predictions that can be certified as robust in terms of satisfying Formula 2.

RQ1: Does the Bayes error indeed grow when certified training is applied? We would like to check if the Bayes error indeed sees a growth when certified training is used. To do that, we first need to obtain the altered distribution used in the context of certified training. As explained in Sect. 3.1, certified training extends the label of an input to its vicinity, and thus results in a convolutional effect across the entire given distribution. Therefore, we can obtain a convolved distribution of each given distribution with each vicinity. Then, we compare the original distribution and the convolved distribution of a given data set, and the Bayes error of the distribution before and after convolution.

We use the Moons and Chan data sets, setting a L^∞ vicinity at $\epsilon = 0.15$. Then, we get the convolved distribution of each data set (using FFT-based convolution, implemented through `scipy.signal.fftconvolve`) and the results are demonstrated in Fig. 4. Observing the comparison shown in Fig. 4(a, b) and (c, d), we can see that the original distribution has gone through a "melting" process, *i.e.*, the peaks of each distribution becomes lower, and the spread increases. For example, in Fig. 4b, the upper moon's centre region (around $x_1 = 0, x_2 = 0.6$) has a higher concentration of inputs from the lower moon than that in (a). This is because convolution with a rectangular function, *e.g.*, vicinity function in our case, is essentially smoothing the original conditional distribution.

To quantify the increased overlap between the density function of each distribution after convolution, we compute their Bayes error. For Moons, the original Bayes error (Fig. 4a) is 8.54%, while the Bayes error after convolution is 9.24%. Similarly, for Chan, Bayes error increases from 5.39% (c) to 9.66% (d). As expected, the Bayes errors do grow, with the growth ranging from 8% to nearly 80%.

(a) Moons $\beta_D = 8.54$, $\zeta_D = 14.28$, $\zeta_D^\sharp \approx 100\,(\%)$

(b) Chan $\beta_D = 5.38$, $\zeta_D = 65.77$, $\zeta_D^\sharp \approx 100\,(\%)$

(c)   FashionMNIST $\beta_D = 3.15$, $\zeta_D = 25$, $\zeta_D^\sharp \approx 32.56\,(\%)$

(d) CIFAR-10 $\beta_D = 5.24$, $\zeta_D = 32.51$, $\zeta_D^\sharp \approx 36.81\,(\%)$

Fig. 5. Upper bounds of robustness/accuracy and the state-of-the-art classifier's performance. The L^∞ vicinity size for certified training /certified robust accuracy for each data set is $\epsilon = 0.15, 0.15, 0.1, 2/255$ for Moons, Chan, FashionMNIST, and CIFAR-10.

We find that convolving with the same vicinity function results in very different growth in the Bayes error. This is likely due to the shape of the original density function. For instance, each moon in the Moons distribution can be approximately seen as a single-modal distribution, and the density function does not have sharp changes. In contrast, the density function of each class's conditional distribution in Chan has sharper value changes at the central region (around $x_1 = 0, x_2 = 0.5$). This may suggest that the Chan distribution exhibits a larger shape change to its original distribution after convolution than Moons. Particularly, in Chan, we observe that the class with the highest probability at the central region changes. Originally, class-0 examples have a higher density in this region. However, after convolution, we can see from Fig. 4d that this region is filled more with class-1 examples than class-0 examples. Essentially, this change shows a significant prediction change in the Bayes classifier. This is likely because convolution has a larger influence on the distributions with features with high (2D) frequencies.

In summary, by comparing the Bayes error before and after distribution alteration, we conclude that the Bayes error does increase when certified training is used, which aligns with Throrem 2. Moreover, the distribution alteration has a larger impact on distributions with high-frequency features than on originally smooth distributions.

RQ2: Is our upper bound of robustness empirically effective? Next, we check whether the computed upper bound of robustness is indeed higher than the existing robustness evaluation in practice. To do that, we apply the closed-form Equation (28) numerically to compute the irreducible robustness error ζ_D for each data set/distribution D. The upper bound of certified robust accuracy is $1 - \zeta_D$. Then, we use ERM and certified training to optimise the corresponding classifier of each data set. As such, we get two trained classifiers for each data set. For each classifier, we compute its performance metrics and compare the classifiers' performance against our upper bounds. We remark that the accuracy and certified robust accuracy may fluctuate when the sample size is not sufficiently large, as seen in Fig. 6. For example, if we are only given five samples,

there is a high chance we get a very high or very low accuracy. For this reason, we gradually increase the sample size of test sets and observe its converged value.
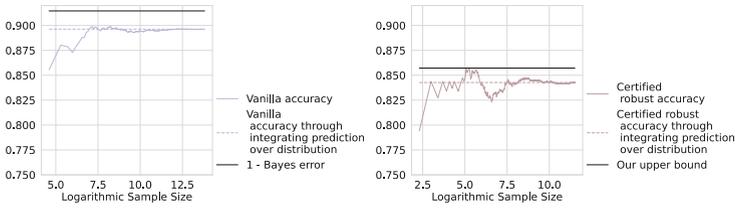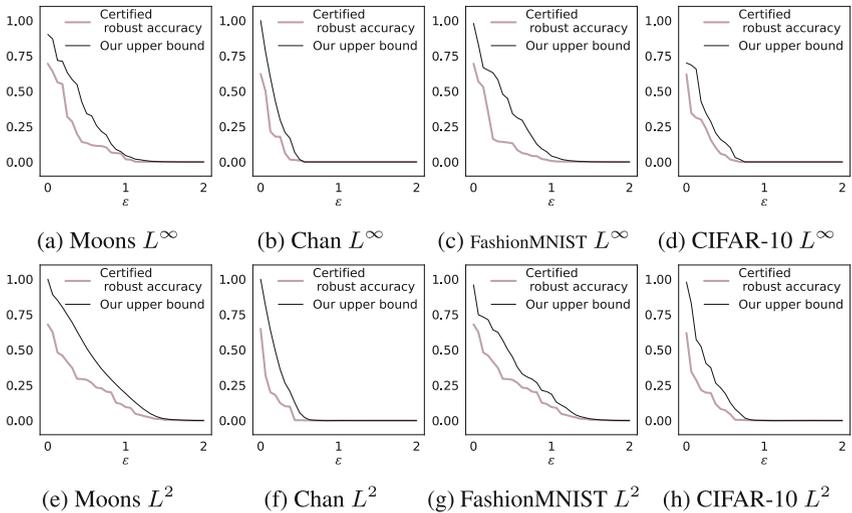

Fig. 6. As sample size grows, the accuracy converges to a value below *1 - Bayes-error*. Similarly, the certified robust accuracy converges to a value below our upper bound of robustness. The figures are computed based on the Moons data set.

The results are shown in Fig. 5. For each data set, the computed value ζ_D is detailed in the caption. The certified robust accuracy is represented by bars in the graph. For example, the MLP for the Moons dataset (seen in Fig. 5a) is trained twice. Initially, it is trained with ERM, achieving a vanilla accuracy of 91.23%, which is nearly the optimal vanilla accuracy of 91.46% (calculated as $1-8.54\%$). Here, the certified robust accuracy is about 80% with an L^∞ vicinity of $\epsilon = 0.15$. When trained a second time with certified training, the MLP's vanilla accuracy slightly decreases to 89.66%, but its certified robust accuracy improves by 5.1%, at 84.24%. The improved certified robust accuracy is below the theoretical upper bound (marked by a dashed line in Fig. 5a, below the annotation ζ_D), which is calculated to be 85.72% $(1-14.28\%)$. Furthermore, the gap between the certified robust accuracy of this classifier and its upper limit is relatively small, approximately 1.5% in absolute percentage points.

Based on the result, we have multiple observations. First, we find that $1 - \zeta_D$ consistently exceeds the certified robust accuracy achieved by state-of-the-art method [35] across various datasets in Fig. 5. This gap, ranging from 1.5% to 7.1%, indicates the potential for further improving classifier robustness within these theoretical limits. For example, the Moons dataset has a small gap, suggesting limited room for improvement, while larger gaps in datasets like the Chan, FashionMNIST, and CIFAR-10 indicate more significant opportunities for increasing the robustness.

Second, we note that ζ_D consistently surpasses the Bayes error β_D by a significant margin for all D. For example, in the Moons dataset, ζ_D is 14.2%, which is 66% higher than its β_D of 8.54%. This implies that robustness against perturbations is challenging, even when the inherent uncertainty of the data is considered. In datasets like Fashion-MNIST and CIFAR-10, despite their low Bayes error of 3.1%-5.2%, their ζ_D are at least six times higher (25.0%-32.7%). This indicates that factors other than inherent data uncertainty are affecting robustness. These factors are likely the newly generated uncertainty from certified training. Moreover, some gaps between ζ_D and β_D are particularly large (*e.g.*, Fig. 5b). Such instances highlight the robustness challenges presented by each dataset can vary. Recall Fig. 4d, the distribution of Chan can be particularly sensitive to convolution with vicinity.

Third, recall that the upper bound ζ_D and certified robust accuracy are based on Formula 2 and Equation (28), and they do not consider the correctness of the label (Definition 2). If we take into consideration the correctness of the examples in the vicinity, we can compute a tighter bound $\zeta_D^\sharp$ based on Equation (20), and certified robust accuracy (from Def. 2) is calculated by sampling a large finite number of neighbours of the input and evaluating their correctness. As the test sample size grows, more examples appear in the vicinity of some training samples, and the likelihood of correctly predicting all of them decreases. This result is also illustrated in the right-most columns in Fig. 5. As observed, certified robust accuracy (from Def. 2) is always lower than $1 - \zeta_D^\sharp$. For instance, the certified robust accuracy of Moons (and Chan) decreases from 84.24% (and 32.35%) respectively to less than 10^{-7}, and that of FashionMNIST (and CIFAR-10) decreases from 73.78% (and 60.12%) to 41.66% (and 54.26%) respectively. Such large reductions indicate a potential need for rethinking the robustness requirement, which may lead to different ways of defining and achieving robustness.

(a) Moons L^∞ (b) Chan L^∞ (c) FashionMNIST L^∞ (d) CIFAR-10 L^∞

(e) Moons L^2 (f) Chan L^2 (g) FashionMNIST L^2 (h) CIFAR-10 L^2

Fig. 7. As epsilon increases, we plot the robustness upper bound change as well as classifiers' certified robust accuracy change in the Moons and Chan dataset.

RQ3: How does the upper bound of robustness vary when the vicinity size grows? In the following, we investigate what can influence the value of irreducible robustness error/upper bound of certified robust accuracy. We already know that when the vicinity grows, it empirically becomes more difficult for a classifier to be robust [35]. The question is then: how about the irreducible robustness error? Is it dependent on the size of the vicinity? If so, how are they correlated? To answer this question, we extend our experiment to cover various vicinity shapes (L^∞ and L^2), and different vicinity sizes (from 0 to 2).

The results are shown in Fig. 7. Each sub-figure in Fig. 7 illustrates the impact of increasing the vicinity size (ϵ) on the upper bound ($1 - \zeta_D$). For instance, in Fig. 7a, we present the change of $1 - \zeta_D$ as well as the classifier's certified robust accuracy after certified training. We observe that for all datasets (Moons, Chan, FashionMNIST, CIFAR-10) and norms (L^∞ and L^2), as the vicinity size grows, both the robustness upper bound and certified robust accuracy decrease monotonically. This indicates an inverse relationship between the upper bound and vicinity size. This finding aligns with our intuition that when vicinity size grows it becomes more difficult for a classifier to be robust. Notably, the CIFAR-10/Chan dataset shows a sharper decline in the upper bound than Moons/FashionMNIST, suggesting that some data distributions may inherently withstand perturbation better, which is consistent with our previous findings. Implementation of our experiment is available on our GitHub page[1].

5   Related Works

This work is closely related to research on Bayes errors and certified training. Computing the Bayes error of a given data distribution has been studied for over half a century [12]. Several works have derived upper and lower bounds of the Bayes error and proposed ways to estimate those bounds. Various f-divergences, such as the Bhattacharyya distance [13] or the Henze-Penrose divergence [7,43], have been studied. Other approaches include directly estimating the Bayes error with f-divergence representation instead of using a bound [36], and computing the Bayes error of generative models learned using normalizing flows [22,48]. More recently, a method has been proposed to evaluate Bayes error estimators on real-world datasets [41], for which we usually do not know the true Bayes error. While these existing studies concentrate on vanilla accuracy, our approach extends the study into the realm of robustness. Besides, some studies may argue that the real-world datasets are well-separated so therefore the Bayes error predicted by the theorems may not be as severe [62]. However, due to the information loss (photo-taking or compression), Bayes errors inevitably exist. For instance, Over 1/3 of CIFAR-10 inputs have been re-annotated by human annotators to have non-fixed labels (CIFAR-10H) [39], indicating non-zero uncertainty. Hence, calculating irreducible error, regardless of severity, holds significance in understanding the inherent limit of certified robustness.

Many certified training techniques have been developed to increase certified robust accuracy, including branch-and-bound [4,16,56], linear relaxation [2,32,46], Lipschitz or curvature verification [29,50], and others [27]. In addition, a number of training techniques have been proposed specifically for improving certified robustness [28], which include warm-up training [45], small boxes [35], and so on. Certified robust accuracy has seen only limited growth over the past decade, prompting research efforts to understand why. Besides the Bayes error perspective, there exists an explanation for this problem from the standpoint of the abstraction domain [33]. However, note that these studies often only focus on the concept of interval arithmetic. Additionally, factors such as the trade-off between certified robustness and vanilla accuracy have also been explored [37,51].

---

[1] https://github.com/cat-claws/irreducible-robustness-error.

6   Conclusion

In this work, we study the limit of classification robustness against perturbations. We are motivated by the observation that the robustness of existing certified classifiers tends to be suboptimal, and hypothesise that there is an irreducible robustness error linked to the classification distribution itself. We formally prove that this irreducible robustness error does exist and is greater than the Bayes error. Further, we present how to calculate the upper bound of robustness based on the data distribution and the vicinity within which we demand robustness. Besides, this work also provides empirical experiments that compute our upper bound on common machine learning data sets. Results show that our robustness upper bound is empirically effective. We conclude that the limit of classification robustness can be well elaborated from the Bayes error perspective and we hope that the upper bound we derive can enlighten future developments on certified training and other robust-classifier training.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Antos, A., Devroye, L., Gyorfi, L.: Lower bounds for Bayes error estimation. IEEE Trans. Pattern Anal. Mach. Intell. **21**(7), 643–645 (1999). https://doi.org/10.1109/34.777375
2. Baader, M., Mueller, M.N., Mao, Y., Vechev, M.: Expressivity of reLU-networks under convex relaxations. In: The Twelfth International Conference on Learning Representations (2024). https://openreview.net/forum?id=awHTL3Hpto
3. Bai, T., Luo, J., Zhao, J., Wen, B., Wang, Q.: Recent advances in adversarial training for adversarial robustness. In: Zhou, Z.H. (ed.) Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21. pp. 4312–4321. International Joint Conferences on Artificial Intelligence Organization (8 2021). https://doi.org/10.24963/ijcai.2021/591, https://doi.org/10.24963/ijcai.2021/591, survey Track
4. Bak, S., Tran, H.D., Hobbs, K., Johnson, T.T.: Improved geometric path enumeration for verifying Relu neural networks. In: Lahiri, S.K., Wang, C. (eds.) Computer Aided Verification, pp. 66–96. Springer International Publishing, Cham (2020)
5. Balagani, K.S., Phoha, V.V.: On the relationship between dependence tree classification error and Bayes error rate. IEEE Trans. Pattern Anal. Mach. Intell. **29**(10), 1866–1868 (2007). https://doi.org/10.1109/TPAMI.2007.1184
6. Balunovic, M., Vechev, M.T.: Adversarial training and provable defenses: bridging the gap. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020). https://openreview.net/forum?id=SJxSDxrKDr

7. Berisha, V., Wisler, A., Hero, A.O., Spanias, A.: Empirically estimable classification bounds based on a nonparametric divergence measure. IEEE Trans. Signal Process. **64**(3), 580–591 (2016). https://doi.org/10.1109/TSP.2015.2477805

8. Chen, Q., Cao, F., Xing, Y., Liang, J.: Evaluating classification model against Bayes error rate. IEEE Trans. Pattern Anal. Mach. Intell. **45**(8), 9639–9653 (2023). https://doi.org/10.1109/TPAMI.2023.3240194

9. Chiang, P., Ni, R., Abdelkader, A., Zhu, C., Studer, C., Goldstein, T.: Certified defenses for adversarial patches. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020). https://openreview.net/forum?id=HyeaSkrYPH

10. Cohen, J., Rosenfeld, E., Kolter, Z.: Certified adversarial robustness via randomized smoothing. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 1310–1320. PMLR, PMLR (09–15 Jun 2019). https://proceedings.mlr.press/v97/cohen19c.html

11. Dučinskas, K., Zikarienundefined, E.: Actual error rates in classification of the t-distributed random field observation based on plug-in linear discriminant function. Informatica **26**(4), 557–568 (Jan 2015)

12. Fukunaga, K., Hostetler, L.: k-nearest-neighbor Bayes-risk estimation. IEEE Trans. Inf. Theory **21**(3), 285–293 (1975). https://doi.org/10.1109/TIT.1975.1055373

13. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press Professional Inc, USA (1990)

14. Ganin, Y., et al.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**(59), 1–35 (2016). http://jmlr.org/papers/v17/15-239.html

15. Garber, F., Djouadi, A.: Bounds on the Bayes classification error based on pairwise risk functions. IEEE Trans. Pattern Anal. Mach. Intell. **10**(2), 281–288 (1988). https://doi.org/10.1109/34.3891

16. Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., Vechev, M.: Ai2: Safety and robustness certification of neural networks with abstract interpretation. In: 2018 IEEE Symposium on Security and Privacy (SP), pp. 3–18 (2018). https://doi.org/10.1109/SP.2018.00058

17. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings (2015). http://arxiv.org/abs/1412.6572

18. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer, 2 edn. (2009). https://hastie.su.domains/Papers/ESLII.pdf

19. Hummels, D.M.: Nonparametric estimation of the Bayes error. Ph.D. thesis, Purdue University (1987). https://docs.lib.purdue.edu/dissertations/AAI8814491/

20. Ishida, T., Yamane, I., Charoenphakdee, N., Niu, G., Sugiyama, M.: Is the performance of my deep network too good to be true? a direct approach to estimating the Bayes error in binary classification. In: The Eleventh International Conference on Learning Representations (2023). https://openreview.net/forum?id=FZdJQgy05rz

21. Kang, H.J., Doermann, D.: Product approximation by minimizing the upper bound of Bayes error rate for Bayesian combination of classifiers. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004, vol. 1, pp. 252–255 Vol.1 (2004). https://doi.org/10.1109/ICPR.2004.1334071

22. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018). https://proceedings.neurips.cc/paper_files/paper/2018/file/d139db6a236200b21cc7f752979132d0-Paper.pdf

23. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
24. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Workshop Track Proceedings. OpenReview.net (2017). https://openreview.net/forum?id=HJGU3Rodl
25. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial machine learning at scale. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings. OpenReview.net (2017). https://openreview.net/forum?id=BJm4T4Kgx
26. LeCun, Y., Cortes, C., Burges, C.: http://yann.lecun.com/exdb/mnist/
27. Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., Jana, S.: Certified robustness to adversarial examples with differential privacy. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 656–672. IEEE (2019)
28. Li, L., Xie, T., Li, B.: Sok: Certified robustness for deep neural networks. In: 44th IEEE Symposium on Security and Privacy, SP 2023, San Francisco, CA, USA, 22-26 May 2023. IEEE (2023). https://arxiv.org/abs/2009.04131
29. Li, Q., Haque, S., Anil, C., Lucas, J., Grosse, R.B., Jacobsen, J.H.: Preventing gradient attenuation in lipschitz constrained convolutional networks. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019). https://proceedings.neurips.cc/paper_files/paper/2019/file/1ce3e6e3f452828e23a0c94572bef9d9-Paper.pdf
30. Lin, W., et al.: Robustness verification of classification deep neural networks via linear programming. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11418–11427 (June 2019)
31. Liu, H., Zhu, X., Lei, Z., Li, S.Z.: Adaptiveface: adaptive margin and sampling for face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
32. Lyu, Z., Guo, M., Wu, T., Xu, G., Zhang, K., Lin, D.: Towards evaluating and training verifiably robust neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4308–4317 (June 2021)
33. Mirman, M., Baader, M., Vechev, M.T.: The fundamental limits of interval arithmetic for neural networks. CoRR **abs/2112.05235** (2021). https://arxiv.org/abs/2112.05235
34. Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of Machine Learning. The MIT Press (2012)
35. Müller, M.N., Eckert, F., Fischer, M., Vechev, M.T.: Certified training: small boxes are all you need. CoRR **abs/2210.04871** (2022). https://doi.org/10.48550/arXiv.2210.04871, https://doi.org/10.48550/arXiv.2210.04871
36. Noshad, M., Xu, L., Hero, A.: Learning to benchmark: Determining best achievable misclassification error from training data. arXiv preprint arXiv:1909.07192 (2019)
37. Pang, T., Lin, M., Yang, X., Zhu, J., Yan, S.: Robustness and accuracy could be reconcilable by (proper) definition. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., Sabato, S. (eds.) International Conference on Machine Learning, ICML 2022, 17–23 July 2022, Baltimore, Maryland, USA. Proceedings of Machine Learning Research, vol. 162, pp. 17258–17277. PMLR (2022). https://proceedings.mlr.press/v162/pang22a.html
38. Papernot, N., McDaniel, P.D., Goodfellow, I.J.: Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. CoRR **abs/1605.07277** (2016). http://arxiv.org/abs/1605.07277
39. Peterson, J.C., Battleday, R.M., Griffiths, T.L., Russakovsky, O.: Human uncertainty makes classification more robust. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)

40. Raghunathan, A., Steinhardt, J., Liang, P.: Certified defenses against adversarial examples. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018). https://openreview.net/forum?id=Bys4ob-Rb

41. Renggli, C., Rimanic, L., Hollenstein, N., Zhang, C.: Evaluating bayes error estimators on real-world datasets with feebee. In: Vanschoren, J., Yeung, S. (eds.) Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual (2021). https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/045117b0e0a11a242b9765e79cbf113f-Abstract-round2.html

42. Ripley, B.D.: Pattern recognition and neural networks. Cambridge Univ. Press (1996). https://doi.org/10.1017/CBO9780511812651

43. Sekeh, S.Y., Oselio, B., Hero, A.O.: Learning to bound the multi-class Bayes error. IEEE Trans. Signal Process. **68**, 3793–3807 (2020). https://doi.org/10.1109/TSP.2020.2994807

44. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1528–1540. CCS '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2976749.2978392

45. Shi, Z., Wang, Y., Zhang, H., Yi, J., Hsieh, C.J.: Fast certified robust training with short warmup. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 18335–18349. Curran Associates, Inc. (2021). https://proceedings.neurips.cc/paper/2021/file/988f9153ac4fd966ea302dd9ab9bae15-Paper.pdf

46. Singh, G., Gehr, T., Püschel, M., Vechev, M.: An abstract domain for certifying neural networks. Proc. ACM Program. Lang. **3**(POPL) (Jan 2019). https://doi.org/10.1145/3290354

47. Szegedy, C., et al.: Intriguing properties of neural networks. In: Bengio, Y., LeCun, Y. (eds.) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings (2014). http://arxiv.org/abs/1312.6199

48. Theisen, R., Wang, H., Varshney, L.R., Xiong, C., Socher, R.: Evaluating state-of-the-art classification models against bayes optimality. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 9367–9377. Curran Associates, Inc. (2021). https://proceedings.neurips.cc/paper_files/paper/2021/file/4e0ccd2b894f717df5ebc12f4282ee70-Paper.pdf

49. Tramer, F., Carlini, N., Brendel, W., Madry, A.: On adaptive attacks to adversarial example defenses. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 1633–1645. Curran Associates, Inc. (2020). https://proceedings.neurips.cc/paper_files/paper/2020/file/11f38f8ecd71867b42433548d1078e38-Paper.pdf

50. Trockman, A., Kolter, J.Z.: Orthogonalizing convolutional layers with the cayley transform. In: International Conference on Learning Representations (2021). https://openreview.net/forum?id=Pbj8H_jEHYv

51. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness may be at odds with accuracy. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019. OpenReview.net (2019). https://openreview.net/forum?id=SyxAb30cY7

52. Tumer, K., Ghosh, J.: Estimating the Bayes error rate through classifier combining. In: Proceedings of 13th International Conference on Pattern Recognition, vol. 2, pp. 695–699 (1996). https://doi.org/10.1109/ICPR.1996.546912

53. Vaishnavi, P., Eykholt, K., Rahmati, A.: Accelerating certified robustness training via knowledge transfer. CoRR **abs/2210.14283** (2022). https://doi.org/10.48550/arXiv.2210.14283

54. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer New York, New York, NY (2000). https://doi.org/10.1007/978-1-4757-3264-1

55. Wang, C., Zhang, D., Huang, S., Li, X., Ding, L.: Crafting adversarial email content against machine learning based spam email detection. In: Proceedings of the 2021 International Symposium on Advanced Security on Software and Systems, pp. 23–28. ASSS '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3457340.3458302

56. Wang, S., Pei, K., Whitehouse, J., Yang, J., Jana, S.: Formal security analysis of neural networks using symbolic intervals. In: 27th USENIX Security Symposium (USENIX Security 18), pp. 1599–1614. USENIX Association, Baltimore, MD (Aug 2018). https://www.usenix.org/conference/usenixsecurity18/presentation/wang-shiqi

57. Weng, T., et al.: Evaluating the robustness of neural networks: an extreme value theory approach. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018). https://openreview.net/forum?id=BkUHlMZ0b

58. Wong, E., Rice, L., Kolter, J.Z.: Fast is better than free: revisiting adversarial training. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020. OpenReview.net (2020). https://openreview.net/forum?id=BJx040EFvH

59. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. CoRR **abs/1708.07747** (2017). http://arxiv.org/abs/1708.07747

60. Xu, K., et al.: Automatic perturbation analysis for scalable certified robustness and beyond. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 1129–1141. Curran Associates, Inc. (2020). https://proceedings.neurips.cc/paper/2020/file/0cbc5671ae26f67871cb914d81ef8fc1-Paper.pdf

61. Yang, S.H., Hu, B.-G.: Discriminative feature selection by nonparametric Bayes error minimization. IEEE Trans. Knowl. Data Eng. **24**(8), 1422–1434 (2012). https://doi.org/10.1109/TKDE.2011.92

62. Yang, Y.Y., Rashtchian, C., Zhang, H., Salakhutdinov, R.R., Chaudhuri, K.: A closer look at accuracy vs. robustness. Adv. Neural Inform. Process. Syst. **33**, 8588–8601 (2020)

63. Zhang, H., Chen, H., Song, Z., Boning, D.S., Dhillon, I.S., Hsieh, C.: The limitations of adversarial training and the blind-spot attack. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019. OpenReview.net (2019). https://openreview.net/forum?id=HylTBhA5tQ

64. Zhang, H., Weng, T.W., Chen, P.Y., Hsieh, C.J., Daniel, L.: Efficient neural network robustness certification with general activation functions. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. NIPS'18, vol. 31, p. 4944-4953. Curran Associates Inc., Red Hook, NY, USA (2018)

65. Zhang, J.G., Deng, H.W.: Gene selection for classification of microarray data based on the bayes error. BMC Bioinform. **8**(1), 370 (Oct 2007). https://doi.org/10.1186/1471-2105-8-370

66. Zhang, T.: Statistical behavior and consistency of classification methods based on convex risk minimization. Ann. Stat. **32**(1), 56–85 (2004). https://doi.org/10.1214/aos/1079120130

67. Zhao, M.J., Edakunni, N., Pocock, A., Brown, G.: Beyond Fano's inequality: bounds on the optimal f-score, ber, and cost-sensitive risk and their implications. J. Mach. Learn. Res. **14**(32), 1033–1090 (2013). http://jmlr.org/papers/v14/zhao13a.html