Towards Large Certified Radius in Randomized Smoothing Using Quasiconcave Optimization

Bo-Han Kung, Shang-Tse Chen

National Taiwan University
d10922019@csie.ntu.edu.tw, stchen@csie.ntu.edu.tw

Abstract

Randomized smoothing is currently the state-of-the-art method that provides certified robustness for deep neural networks. However, due to its excessively conservative nature, this method of incomplete verification often cannot achieve an adequate certified radius on real-world datasets. One way to obtain a larger certified radius is to use an input-specific algorithm instead of using a fixed Gaussian filter for all data points. Several methods based on this idea have been proposed, but they either suffer from high computational costs or gain marginal improvement in certified radius. In this work, we show that by exploiting the quasiconvex problem structure, we can find the optimal certified radii for most data points with slight computational overhead. This observation leads to an efficient and effective input-specific randomized smoothing algorithm. We conduct extensive experiments and empirical analysis on CIFAR-10 and ImageNet. The results show that the proposed method significantly enhances the certified radii with low computational overhead.

Introduction

Although deep learning has achieved tremendous success in various fields (Wang, Bochkovskiy, and Liao 2022; Zhai et al. 2022), it is known to be vulnerable to adversarial attacks (Szegedy et al. 2013). This kind of attack crafts an imperceptible perturbation on images (Goodfellow, Shlens, and Szegedy 2014) or voices (Carlini and Wagner 2018) to make the AI system predict incorrectly. Many adversarial defense methods have been proposed to defend against adversarial attacks. Adversarial defenses can be categorized into empirical defenses and theoretical defenses. Common empirical defenses include adversarial training (Madry et al. 2017; Shafahi et al. 2019; Wong, Rice, and Kolter 2020) and preprocessing-based methods (Samangouei, Kabkab, and Chellappa 2018; Das et al. 2018). Though effective, empirical defenses cannot guarantee robustness.

Different from empirical defenses, theoretical defenses provide robustness verification with mathematical guarantees (Li, Xie, and Li 2023). The verification must be *sound*, thereby preventing false positives. However, while it has the potential to be *complete*, ensuring no false negatives, it may also be *incomplete*. One the other hand, theoretical defenses can be categorized into *deterministic* verification and *probabilistic* verification. Deterministic verification methods, such as mixed-integer programming (MIP) (Tjeng, Xiao, and Tedrake 2018), interval bound propagation (IBP)(Ehlers 2017; Weng et al. 2018; Gowal et al. 2018; Mueller et al. 2022) and Lipschitz-bounded networks (Singla and Feizi 2021; Singla, Singla, and Feizi 2022; Xu, Li, and Li 2022), offer theoretical robustness guarantees for a DNN model. The verification in these cases is deterministic. In contrast, probabilistic verification methods, such as randomized smoothing (Cohen, Rosenfeld, and Kolter 2019; Lecuyer et al. 2019; Yang et al. 2020), provide a provable defense that ensures that there are no adversarial examples within a specific ball with a radius R. Notably, this guarantee is accompanied by a degree of randomness that is independent of input images.

Among those methods, only randomized smoothing (RS) can scale to state-of-the-art deep neural networks and real-world datasets. Randomized smoothing first builds a smoothed classifier for a given data point via a Gaussian filter and Monte Carlo sampling, and then it estimates a confidence lower bound for the highest-probability class. Next, it determines a certified radius for the class and promise that there is no adversarial example within this radius.

Although randomized smoothing is effective, it suffers from two main disadvantages. First, randomized smoothing applies the same constant-variance Gaussian filter to every data point when constructing a smoothed classifier. This makes the certified radius dramatically underestimated. Second, randomized smoothing adopts a confidence lower bound (Clopper-Pearson lower bound) to estimate the highest-probability class, which also limits the certified radius. As a result, when evaluating radius-accuracy curve, a truncation fall often occurs (see the gray curve in the upper right of Fig. 1). This is called *truncation effect* or waterfall effect (Súkeník, Kuvshinov, and Günnemann 2021), which shows the conservation aspect in randomized smoothing. Other issues such as fairness (Mohapatra et al. 2021), dimension (Kumar et al. 2020b), and time-efficiency (Chen et al. 2022) also limit its application.

To alleviate truncation effect and improve the certified radii, a more precise workflow is necessary. Prior work (Chen et al. 2021; Alfarra et al. 2022) proposed input-

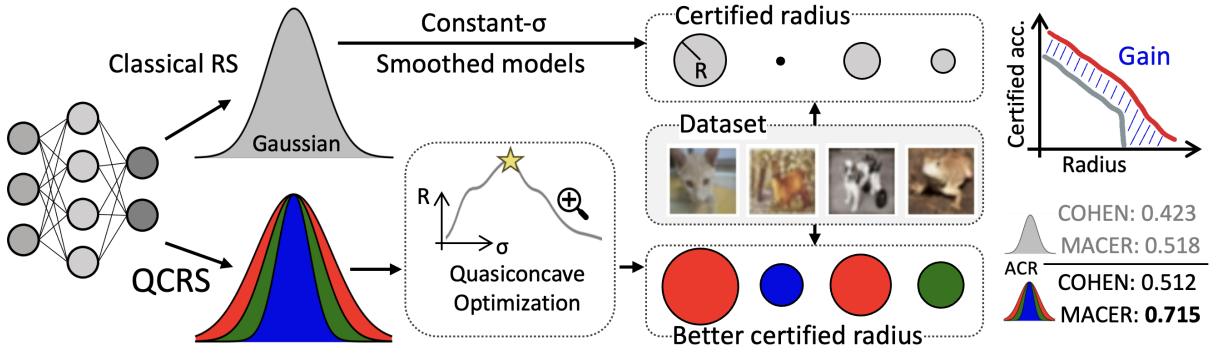


Figure 1: Overview of the proposed QCRS algorithm. QCRS finds the better sigma value for each smoothed classifier using quasiconcave optimization. Thus, it provides better certified radii than the classical randomized smoothing. In this paper, we discuss quasiconcavity on the certified radius w.r.t. σ. i.e., $R(\sigma)$.

specific methods that can assign different Gaussian filters to different data points. Those methods attempt to optimize the radius by finding the optimal variance σ^2 of the Gaussian filter. In this work, we first delve into randomized smoothing and discover a useful property called quasiconcavity for the sigma-radius curve. Next, we define a probabilistic quasiconcavity assumption and then develop a novel algorithm called **Quasiconcavity-based Randomized Smoothing (QCRS)** that optimizes certified radii with respect to sigma. The overview of QCRS is illustrated in Fig. 1. QCRS significantly improves the certified radii with little computational overhead compared to existing methods (Chen et al. 2021; Alfarra et al. 2022). The proposed QCRS enjoys the advantages of both performance and time-efficiency. The main technical contributions are summarized as follows:

- We discover and empirically demonstrate that the sigma-radius curves are quasiconcave for most data points. In our experiments, approximately 99% of the data points satisfy our proposed quasiconcavity condition.
- Based on the observed quasiconcavity property, we propose a novel and efficient input-specific algorithm called QCRS. This algorithm aims to enhance certified radii and alleviate the truncation effect in randomized smoothing.
- Through extensive experiments, we demonstrate the effectiveness of our proposed QCRS method on the CIFAR-10 and ImageNet datasets. Furthermore, by combining QCRS with a training-based method, we achieve state-of-the-art certified radii.

Related Work

Randomized smoothing utilizes a spatial low-pass Gaussian filter to construct a smoothed model (Cohen, Rosenfeld, and Kolter 2019). Based on the Neyman-Pearson lemma, this smoothed model can provide a provable radius R to guarantee robustness for large-scale datasets. To improve randomized smoothing, some works (Yang et al. 2020; Zhang et al. 2020; Levine and Feizi 2021) proposed general methods using different smoothing distribution for different ℓ_p balls, while others tried to provide a better and tighter certification (Kumar et al. 2020a; Levine et al. 2020).

Improving RS during training phase. To further enlarge the radius R, some works used training-based method (Salman et al. 2019; Zhai et al. 2019; Jeong et al. 2021; Anderson and Sojoudi 2022). These models were specifically designed for randomized smoothing. For example, MACER (Zhai et al. 2019) made the computation of certified radius differentiable and add it to the standard cross-entropy loss. Thus, the average certified radius of MACER outperforms the Gaussian-augmentation model that was used by the original randomized smoothing.

Improving RS during inference phase. Different from training-based method, some works utilized different smoothing methods to enhance the certified region. Chen *et al.,* (Chen et al. 2021) proposed a multiple-start search algorithm to find the best parameter for building smoothed classifiers. Alfarra *et al.,* (Alfarra et al. 2022) adopted a memory-based approach to optimize the Gaussian filter of each input data. Chen *et al.,* (Chen et al. 2022) proposed an input-specific sampling acceleration method to control the sampling number and provides fast and effective certification. Li *et al.,* (Li et al. 2022) proposed double sampling randomized smoothing that utilizes additional smoothing information for tighter certification. These inference-time methods are the most relevant to our work. We will provide a more detailed description of these methods later.

Preliminaries

Let $x \in \mathbb{R}^d$ be a data point, where d is the input dimension. $\mathcal{C} = \{1, 2, ..., c\}$ is the set of classes. $F : \mathbb{R}^d \to \mathbb{R}^c$ is a general predictor such as neural networks. We define the base classifier as

$$f(x) = e_\xi; \quad \xi = \arg\max_j F_j(x), \tag{1}$$

where e_j denotes a one-hot vector where the j^{th} component is 1 and all the other components are 0. The smoothed classifier (Cohen, Rosenfeld, and Kolter 2019) $g : \mathbb{R}^d \to \mathcal{C}$ is defined as

$$g(x) = \arg\max_{c \in \mathcal{C}} Pr[f(x+\epsilon) = e_c], \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I), \tag{2}$$

where $\mathcal{N}$ is Gaussian distribution and ϵ is a noise vector sampled from $\mathcal{N}$. Cohen *et al.,* (COHEN) proposed a provable

method to calculate the certifiable robust radius as follows:

$$R = \frac{\sigma}{2} \cdot [\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})], \tag{3}$$

$p_A = Pr[f(x+\epsilon) = e_A]$, and $p_B = Pr[f(x+\epsilon) = e_B]$, where A is the highest-probability class of the smoothed classifier, and B is the runner-up class. $\underline{p_A}$ and $\overline{p_B}$ are the Clopper-Pearson lower/upper bound of $\underline{p_A}$ and p_B, which can be estimated by Monte Carlo (MC) sampling with a confidence level $1-\alpha$. R indicates the certified radius. Any data point inside this radius would be predicted as class A by the smoothed classifier. That is, $g(x+\delta) = g(x)$ for all $||\delta||_2 \leq R$. In practice, COHEN replaces $\overline{p_B}$ with $1-\underline{p_A}$, so equation 3 usually is reformulated as $R = \sigma \cdot \Phi^{-1}(\underline{p_A})$. If $\underline{p_A} < 0.5$, it indicates that there is no certified radius in this data point according to COHEN.

The smoothed classifier g is constructed from the base classifier f by introducing perturbations ϵ to x. Therefore, the smoothed classifier g can be regarded as a spatial smoothing measure of the original base classifier f using a Gaussian kernel $\mathcal{G}$, i.e., $g = f \star \mathcal{G}$, where $\star$ is the convolution operator. From the signal-processing perspective, g is the Weierstrass transformation of f. That is, the smoothed classifier g is constructed by applying a low-pass filter $\mathcal{G}$ on f. Randomized smoothing constructs smoothed classifier to provide certifiable robustness guarantee.

QCRS Methodology

Observation and Motivation

As mentioned earlier, several existing methods attempt to address the truncation effect. Some focus on training the base model to enlarge certified radii, while others use a different Gaussian kernel $\mathcal{G}(\sigma)$ for each image to construct g. We follow the later approach and propose an input-specific algorithm that finds the optimal $\mathcal{G}$ for most data points. Intuitively, for a data point x of class c_A, if most neighboring points belong to the same class c_A, we can use $\mathcal{G}$ with a larger variance to convolute the data space of f. In contrast, if the neighborhood is full of different class samples, $\mathcal{G}$ needs a small variance to prevent misclassification. The proposed method draws inspiration from this concept and aims to optimize the selection of $\mathcal{G}(\sigma)$. Below, we will discuss some input-specific search algorithms that have been utilized in prior works. These algorithms contribute to the development of our approach.

DDRS (Alfarra et al. 2022) assumes that sigma-radius curves, $R(\sigma) = \sigma \cdot \Phi^{-1}(\underline{p_A}(\sigma))$, are concave and use gradient-based convex optimization along with some relaxation and approximation to find the optimal σ value. However, in our experiments, almost all sigma-radius curves are not concave. We select 200 images from CIFAR-10 dataset and compute the certified radii with respect to σ for each image. Among these 200 images, as least 189 images do not satisfy concavity. Thus, the gradient-based convex optimization method may not work well in this task. Instead of depending on the assumption of concavity, Insta-RS (Chen et al. 2021) uses a multi-start searching algorithm to optimize σ. However, the multi-start procedure incurs high computational overhead. In our work, we observe an intriguing quasiconcave property on the sigma-radius curves, which helps us develop an effective and efficient algorithm to optimize sigma. In order to assess the generality of the quasiconcave property, we conducted evaluations on two datasets and four distinct models. Table 1 illustrates the generality of concavity and quasiconcavity of the sigma-radius curves. The table demonstrates that quasiconcavity is significantly more prevalent than concavity in real-world datasets. We will delve deeper into this table later.

Dataset	CIFAR-10		ImageNet	
Trained Models	$\sigma = .12$	$\sigma = .25$	$\sigma = .25$	$\sigma = .50$
Concave	16%	6.71%	0%	0%
Quasiconcave	97%	98.17%	92.95%	90.47%

Table 1: We evaluate the generality of the concave and quasiconcave properties of the sigma-radius curve on two different datasets and four different trained models. We find that a significant number of data points do not exhibit a concave sigma-radius curve, while over 90% of the data points demonstrate a quasiconcave sigma-radius curve.

Quasiconcavity

This section lists the definition and two fundamental lemmas about quasiconcavity that will be used in designing our QCRS method. Quasiconcavity is a generalization of concavity, defined as follows:

Definition 1. *(quasiconcavity (Boyd and Vandenberghe 2004)). A function $h : \mathbb{R}^n \to \mathbb{R}$ is quasiconcave if $\operatorname{dom} h$ is convex and for any $\theta \in [0,1]$ and $x, y \in \operatorname{dom} h$,*

$$h(\theta x + (1-\theta)y) \geq \min\{h(x), h(y)\}.$$

Furthermore, a function h is strictly quasiconcave if $\operatorname{dom} h$ is convex and for any $x \neq y$, $x, y \in \operatorname{dom} h$, and $\theta \in (0,1)$:

$$h(\theta x + (1-\theta)y) > \min\{h(x), h(y)\}.$$

In this paper, we mainly use strict quasiconcavity. Below, we list lemmas on strict quasiconcavity that we will use later.

Lemma 1. *If a function h is strictly quasiconcave, then any local optimal solution of h must also be globally optimal.*

Lemma 2. *Suppose $h : \mathbb{R} \to \mathbb{R}$ is differentiable, and let x^* be the optimal solution of h. The function h is strictly quasiconcave if and only if the following two statements hold:*

$$\nabla h(x) > 0, \forall x \in (-\infty, x^*) \text{ and } \nabla h(x) < 0, \forall x \in (x^*, \infty)$$

We defer the proofs of these two lemmas to Appendix D.

Design

In this section, we show quasiconcavity related to sigma-radius curves, i.e., $R(\sigma)$. First, we differentiate $R(\sigma)$:

$$\nabla_\sigma R(\sigma) = \frac{\partial R(\sigma)}{\partial \sigma} = \Phi^{-1}(\underline{p_A}(\sigma)) + \sigma \cdot \frac{\partial \Phi^{-1}(\underline{p_A}(\sigma))}{\partial \sigma}$$

Assume that σ^* exists, where $\sigma^* = \arg\max_\sigma R(\sigma)$. Then, according to Lemma 2, strict quasiconcavity is established

if the gradient $\nabla_\sigma \Phi^{-1}$ adheres to the subsequent upper and lower bounds:

$$\begin{cases} \nabla_\sigma \Phi^{-1} = \frac{\partial \Phi^{-1}(\underline{p_A}(\sigma))}{\partial \sigma} > -\frac{\Phi^{-1}(\underline{p_A}(\sigma))}{\sigma} & \text{for } \sigma < \sigma^* \\ \nabla_\sigma \Phi^{-1} = \frac{\partial \Phi^{-1}(\underline{p_A}(\sigma))}{\partial \sigma} < -\frac{\Phi^{-1}(\underline{p_A}(\sigma))}{\sigma} & \text{for } \sigma > \sigma^* \end{cases}$$

Intuitively, it indicates that $\nabla_\sigma \Phi^{-1}$ has a negative lower bound when $\sigma < \sigma^*$ and a negative upper bound when $\sigma > \sigma^*$.

The computation of a classifier's decision boundary is proven to be NP-hard, rendering the calculation of $\nabla_\sigma \Phi^{-1}$ impractical (Katz et al. 2017). Randomized smoothing employs a series of probabilistic approaches, such as MC sampling, to address this issue. Similarly, instead of evaluating $\nabla_\sigma \Phi^{-1}$, we define a probabilistic condition based on Lemma 2 to determine quasiconcavity of $R(\sigma)$:

Definition 2. *((υ^-, υ^+)-SQC condition) Given an optimal σ^*, we call the sigma-radius curve satisfies (υ^-, υ^+)-Strict Quasiconcave Condition ((υ^-, υ^+)-SQC condition), if for any $\{\sigma | R(\sigma) > 0\}$, $\nabla R(\sigma)$ satisfies the following:*

$$\Pr_{\sigma < \sigma^*}[\nabla R(\sigma) > 0] = \upsilon^-, \quad \Pr_{\sigma > \sigma^*}[\nabla R(\sigma) < 0] = \upsilon^+.$$

Intuitively, if $\upsilon^- = \upsilon^+ = 1$, the condition states that the slope of the sigma-radius curve is positive on the left side of the optimal solution and negative on the right side. Note that this condition is weaker and more general than the concentration assumption used in (Li et al. 2022), which requires additional assumptions on the distribution of data points. It is also weaker than the concavity assumption used in DDRS (Alfarra et al. 2022). Since the (υ^-, υ^+)-SQC condition is weak, we expect that more data points would satisfy this assumption. Thus, we conduct some experiments on CIFAR-10 and ImageNet, which are detailed in Appendix B. For each data point in CIFAR-10 and ImageNet, we first employ grid search to determine the optimal sigma σ^*. Subsequently, we uniformly sample 20 distinct sigma values and evaluate the (υ^-, υ^+)-SQC condition. In terms of concavity, we evaluate the Hessian of these 20 points. The results are illustrated in Table 1. Taking the model with $\sigma = .25$ on CIFAR=10 as an example, we derive the following conclusions: 1) There are at most 98.17% data points satisfy the $(1, 1)$-SQC condition, while only a maximum of 6.7% satisfy the concavity assumption; 2) Among these 98.17% data points, their υ^- and υ^+ are both within the interval $[0.8609, 1]$ using 95% confidence level, determined by the Clopper-Pearson interval. That is, these data points at least satisfy $(0.86, 0.86)$-SQC condition. Notably, in our experiment, the mean values of υ^- and υ^+ are greater than 0.99; 3) If we set υ^- and υ^+ to 0.99, we can expect that 95% of data points will achieve the optimal sigma, as the proposed method requires five iterations to converge ($0.99^5 \approx 0.95$). We will discuss the convergence in the next section.

We assume that a data point satisfies (υ^-, υ^+)-SQC condition, with the corresponding υ^- and υ^+ being close to one. According to Lemma 2, if we detect that the gradient of a point is positive, we can assert that the unique optimal sigma is on its right hand side. Based on these rules, we design a time-efficient algorithm that can achieve optimal σ, shown in Algorithm 1. Algorithm 1 finds the optimal sigma efficiently based on binary search, and the sigma is the globally optimal solution, as demonstrated by Lemma 1. On the other hand, the sigma values within the non-certified interval $\{\sigma | R(\sigma) = 0\}$ must not be the solution. The gradients $\nabla R(\sigma)$ are likely to be zero in the interval because the curve is a horizontal line with $R(\sigma) = 0$ there. This leads to a gradient vanishing issue in Algorithm 1. To circumvent this issue, we utilize momentum M to guide the optimization direction. Algorithm 1 guarantees to find the same optimal solution as grid search if the curve satisfies $(1, 1)$-SQC condition. The time complexity is N for grid search and $\log N$ for Algorithm 1, where N is the number of points on the grid. Therefore, the proposed method is significantly faster than grid search, while both of them can achieve the same optimal σ.

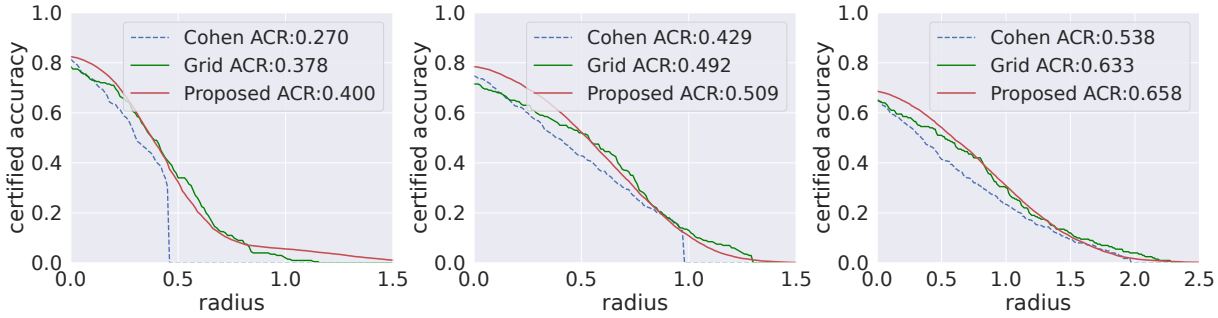
Algorithm 1: Our proposed QCRS method

Input: Searching region σ_{max} and σ_{min}; suboptimal interval ε; original sigma σ_0; gradient step τ
Parameter: momentum $M \leftarrow 0$
Output: The optimal σ

while $\sigma_{max} - \sigma_{min} > \varepsilon$ **do**
  $\sigma \leftarrow (\sigma_{min} + \sigma_{max})/2$
  Calculate the gradient:
    $\nabla_\sigma R(\sigma) \leftarrow R(\sigma + \tau) - R(\sigma - \tau)$
  if $sign(\nabla_\sigma R(\sigma)) > 0$ **then**
    $\sigma_{min} \leftarrow \sigma$; $M \leftarrow 1$
  else if $sign(\nabla_\sigma R(\sigma)) < 0$ **then**
    $\sigma_{max} \leftarrow \sigma$; $M \leftarrow -1$
  else
    if $M \geq 0$ **then**
      $\sigma_{max} \leftarrow \sigma$; $M \leftarrow -1$
    else
      $\sigma_{min} \leftarrow \sigma$; $M \leftarrow 1$
    end if
  end if
end while
$\hat{\sigma} \leftarrow (\sigma_{min} + \sigma_{max})/2$
return $\sigma \leftarrow \arg\max_{\sigma \in \{\hat{\sigma}, \sigma_0\}} R(\sigma)$

Prior work utilizes backpropagation to compute gradients, which is time-consuming, and the computed gradient is unstable due to MC sampling. Therefore, we use forward passes to compute gradient, which takes the difference of two neighboring points. This is because we only care about the gradient *sign* rather than the exact value. On the last stage of Algorithm 1, we employ a rejection policy that compares the resulting σ to the original σ and returns the larger one.

The proposed method is time-efficient compared to Insta-RS (Chen et al. 2021) and DDRS (Alfarra et al. 2022). DDRS uses a low MC sampling number (one or eight) due to expansive computation and may obtain unstable gradients. To verify this, we analyze the value of gradient under different MC sampling number. The detailed results can be found in the Appendix H. The results reveal that the gradient values vary dramatically when using low MC sampling numbers. Thus, a low MC sampling number can not accurately esti-



(a) $\sigma = 0.12$, CIFAR-10 (b) $\sigma = 0.25$, CIFAR-10 (c) $\sigma = 0.50$, CIFAR-10

Figure 2: The comparison between COHEN, grid search, and the proposed QCRS on the CIFAR-10 dataset. The models are trained by Gaussian augmentation with sigma (a) 0.12, (b) 0.25, and (c) 0.50. The proposed QCRS outperforms the baseline method and is very close to grid search. In addition, we can observe the truncation effect on the curves of COHEN.

mate gradients, which would affect the gradient-based optimization. However, the proposed QCRS only uses the gradient sign, which is more stable than the gradient value. We observe that the sign of the gradient is consistent and hardly changes when the MC sampling number exceeds 500.

Convergence Analysis

The proposed QCRS enjoys a convergence guarantee with the convergence rate analyzed below.

Theorem 1. *Suppose (1,1)-SQC condition holds. Given hyper-parameters σ_{min} and σ_{max}, let σ_t be the σ value after t iterations in Algorithm 1. Algorithm 1 converges to optimal σ^* as follows:*

$$\frac{\sigma_{max} - \sigma_{min}}{2^t} \geq |\sigma_t - \sigma^*|.$$

Theorem 1 means the convergence rate of Algorithm 1 is $\mathcal{O}((\frac{1}{2})^t)$. We defer the proof to Appendix D.

On the other hand, gradient-descent-based method requires much stricter assumptions, such as L-smoothness and μ-strongly concavity, to guarantee convergence. When these conditions are satisfied, the convergence rate is as below (Nesterov 2018):

$$|\sigma_t - \sigma^*|^2 \leq (\frac{L-\mu}{L+\mu})^{(t-1)} |\sigma_1 - \sigma^*|^2.$$

The convergence rate is $\mathcal{O}((\frac{L-\mu}{L+\mu})^t)$, where L and μ depend on the data points. Note that only a maximum of 6.7% of data points satisfy the concavity assumption empirically, and even fewer data points satisfy L-smoothness and μ-strongly concavity simultaneously. Thus, some concave optimization methods cannot converge to the optimal sigma values for most data points.

Experimental Results

We evaluate the proposed QCRS and present the experimental results on CIFAR-10 (Krizhevsky, Hinton et al. 2009) and ImageNet (Russakovsky et al. 2015). We also verify that QCRS can be combined with training-based techniques like MACER (Zhai et al. 2019) to produce state-of-the-art certification results. Detailed implementation information is available in Appendix E.[1] We also present additional experiments in Appendix, including an ablation study, error analysis, and more. Following (Zhai et al. 2019), we use the average certified radius (ACR) as the performance metric, defined as: $ACR = \frac{1}{|\mathcal{D}_{test}|} \sum_{x \in \mathcal{D}_{test}} R(x, y; g)$, where $\mathcal{D}_{test}$ is the test dataset, and $R(x, y; g)$ is the certified radius obtained by the smoothed classifier g.

CIFAR-10 and ImageNet

Fig. 2 compares the radius-accuracy curves for different methods on the CIFAR-10 dataset. In the figure, we also show the corresponding ACR, which is the area under the radius-accuracy curve. Table 2 presents the ACR of different methods and their runtime cost. The proposed QCRS outperforms the original randomized smoothing method (Cohen, Rosenfeld, and Kolter 2019) by significant margins: 48%, 18%, and 22% for the models trained with $\sigma = \{0.12, 0.25, 0.50\}$, respectively. The main performance gain comes from reducing the truncation effect (the waterfall effect) on the radius-accuracy curve. We also compare QCRS to grid search. Since grid search is extremely computationally expensive, we only test the images with $id = 0, 50, 100, ..., 9950$ in CIFAR-10. Despite using 24 points in the grid search, which costs approximately 24 times more in runtime than QCRS, QCRS still outperforms grid search. This is due to QCRS being more time-efficient, which enables a broader and more accurate search region compared to grid search. Furthermore, QCRS guarantees to achieve the same optimal as grid search if the $(1, 1)$-SQC condition holds. Regarding the computational cost, the proposed method only takes about 7% additional inference time compared to the original COHEN method, as shown in Table 2.

We also compare the proposed QCRS with two state-of-the-art randomized smoothing methods, DSRS (Li et al. 2022) and DDRS (Alfarra et al. 2022). We follow their setting to evaluate the proposed method on CIFAR-10 for fair comparisons, and defer other minor comparisons and analyses to Appendix C. As Table 3 shows, for the certified accu-

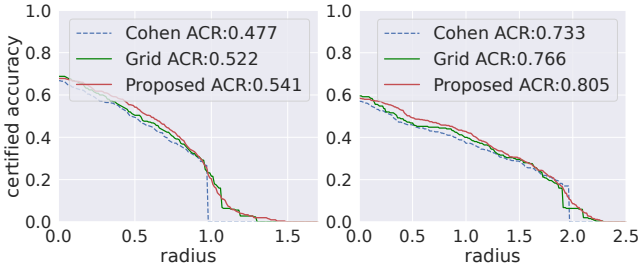
[1]Code: https://github.com/ntuaislab/QCRS

ACR	$\sigma = .12$	$\sigma = .25$	$\sigma = .50$	Time Cost (Sec.)
COHEN	0.270	0.429	0.538	6.50±0.021
DDRS (Alfarra et al. 2022)	0.310	0.448	0.540	7.39±0.016
Grid Search	0.378	0.492	0.633	155.80±0.50
QCRS	**0.400**	**0.509**	**0.658**	6.96±0.017

Table 2: ACR and Time Cost for CIFAR-10.

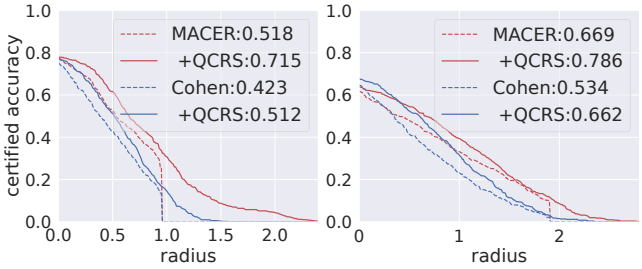
	Certified Accuracy								
Certified radii R	0.25	0.5	0.75	1.0	1.25	1.5	1.75	2.0	2.25
COHEN	0.55	0.41	0.32	0.23	0.15	0.09	0.05	0.00	0.00
DDRS (Alfarra et al. 2022)	0.56	0.44	0.34	0.23	0.15	0.08	0.04	0.00	0.00
DSRS (Li et al. 2022)	0.55	0.42	0.30	0.19	0.09	0.04	0.00	0.00	0.00
Grid Search (24 points)	0.58	0.51	0.42	0.30	0.18	**0.12**	**0.07**	**0.04**	0.01
QCRS (Proposed)	**0.64**	**0.54**	**0.43**	**0.31**	**0.20**	0.11	0.05	0.02	0.01

Table 3: Certified accuracy under different radii R of DSRS, DDRS, Grid Search, and the proposed QCRS.



(a) $\sigma = 0.25$, ImageNet (b) $\sigma = 0.50$, ImageNet

Figure 3: The comparison between COHEN, grid search, and QCRS on ImageNet. Following Cohen, we only use 500 images in the validation set. The models are trained by Gaussian augmentation with $\sigma =$ (a) 0.25 and (b) 0.50.



(a) $\sigma = 0.25$, CIFAR-10 (b) $\sigma = 0.50$, CIFAR-10

Figure 4: The performance of QCRS incorporating training-based methods. We use MACER model with (a) $\sigma = 0.25$ and (b) $\sigma = 0.50$. Both QCRS and MACER provide similar improvements over COHEN, but QCRS incurs little computational overhead. Combining QCRS and MACER provides state-of-the-art certified radii.

racy under radius at 0.5, DSRS and DDRS improve COHEN by 2.4% and 7.3%, respectively. On the other hand, the proposed QCRS improves COHEN by 31.7%. Therefore, among the methods that boost certified radii, QCRS improves COHEN most effectively.

Fig. 3 shows the results on ImageNet. Following COHEN, only 500 images with $id = 0, 100, 200, ..., 49900$ in the validation set were used. We avoided using the $\sigma = 1.0$ model as (Mohapatra et al. 2021) revealed that a large σ leads to serious fairness concerns, necessitating a restriction on σ in randomized smoothing. Additionally, we observed the similar fairness issue in (Mohapatra et al. 2021), to be discussed later. For the model with $\sigma = .25$, the proposed method improves ACR from 0.477 to 0.541, roughly 13.4%. Similarly, for the model with $\sigma = .50$, the proposed method improves ACR from 0.733 to 0.805, roughly 9.8%. In addition, the proposed method overcomes the truncation effect, providing a larger certified radius than COHEN. As for the grid search, similar to CIFAR-10, it is computationally expensive, so we set the number of searching points to be 11 on ImageNet. As mentioned earlier, although the grid search can provide the optimal certified radius if the cost does not matter, its searching region and precision are limited in practical application. That is why the proposed method achieves a slightly superior ACR than the brute-force grid search method in Fig. 3, while the runtime is roughly 11 times faster than it.

MACER

The proposed method focuses on enhancing randomized smoothing while building the smoothed classifier. Thus, it is orthogonal to the approach that aims to boost certified radii during training stage. QCRS can incorporate with training-based methods. The most representative training-based method to enhance certified radius is MACER. We apply the proposed method to the models trained by MACER and observe significant improvement in terms of the certified radius. Fig. 4 illustrates the results of radius-accuracy curves, and Table 4 shows the detailed comparison. As Table 4 shows, for the model trained by $\sigma = .25$, COHEN achieves 0.423 ACR, and MACER enhances this ACR to 0.518, roughly 22.5%. Next, our proposed QCRS improves MACER ACR from 0.518 to 0.715, roughly 38%. Thus, QCRS and MACER together can significantly boost the original Cohen's RS roughly 69%. Similarly, for the model trained by $\sigma = .50$, QCRS and MACER enhance Cohen's RS from 0.534 to 0.786, approximately +47.2%. Furthermore, Table 4 also includes the results of DDRS incorporating MACER. It can be observed that MACER has a positive impact on the performance of the inference-phase randomized smoothing methods. Among the methods evaluated, QCRS incorporating MACER achieves the state-of-the-art

Test	Training COHEN	Training MACER
COHEN	0.423	0.518
DDRS	0.448 (+6%)	0.561 (+8%)
QCRS	0.512 (+21%)	**0.715** (+38%)

(a) $\sigma = 0.25$

Test	Training COHEN	Training MACER
COHEN	0.534	0.669
DDRS	0.540 (+1%)	0.702 (+5%)
QCRS	0.662 (+24%)	**0.786** (+18%)

(b) $\sigma = 0.50$

Table 4: The ACR results of different inference-phase RS incorporating different training-phase RS. The training-phase RS are COHEN or MACER with (a) $\sigma = .25$ and (b) $\sigma = .50$. The inference-phase RS are COHEN, DDRS, or QCRS. The table also includes the improvement ratios compared to COHEN.

performance. On the other hand, QCRS and MACER improves COHEN to 0.512 and 0.518, respectively. That is, QCRS can enlarge the certified radius to the extent that MACER does, but it does not need any training procedure. As datasets become larger and larger today, re-training may be computationally prohibited. Thus, QCRS benefits from its efficient workflow that enlarges the certified radius with negligible cost.

Computational Cost

We briefly analyze the computational cost compared to COHEN. The sigma searching region of Algorithm 1 is $0.5 - 0.12 = 0.38$. Because the convergence rate of Algorithm 1 is $\frac{\sigma_{max}-\sigma_{min}}{2^t} \geq |\sigma_t - \sigma^*|$, if $t \geq 6$, we can achieve 0.006-optimal, i.e., $|\sigma - \sigma^*| < 0.006$. As for the gradient computation, it requires $1,000$ forward propagations for each iteration. Thus, we roughly require additional $6,000$ forward propagations for each data point to achieve the optimal sigma value. The standard RS needs $100,000$ forward propagations, so the overhead of the proposed QCRS is approximately 6%. Empirically, we observe an approximately 7% overhead as Table 2 illustrates.

We compare the computational cost of QCRS with those of DDRS (Alfarra et al. 2022), DSRS (Li et al. 2022), and Insta-RS (Chen et al. 2021). The DDRS method adopts the radius as its objective function and utilizes gradient descent to directly optimize the radius. This approach involves computing the gradient of the radius multiple times to update the sigma value, which requires performing multiple back propagations. As shown in Table 2, certifying an image using DDRS takes roughly 7.39 seconds. This incurs an overhead of 14%, which is twice as high as the proposed QCRS method. As for DSRS, in our experiments, it incurs roughly 100% overhead compared to COHEN. Finally, Insta-RS employs multi-start gradient descent, so its computational cost is the highest among the methods considered in this paper.

airplanes	cars	birds	cats	deer
0.53 ± 0.21	$\mathbf{0.84 \pm 0.41}$	0.37 ± 0.10	0.36 ± 0.11	$\mathbf{0.36 \pm 0.10}$
dogs	**frogs**	**horses**	**ships**	**trucks**
0.50 ± 0.17	0.53 ± 0.08	0.53 ± 0.12	0.42 ± 0.08	0.61 ± 0.14

Table 5: The optimal sigma values for different classes in CIFAR-10. The base model is trained with $\sigma = 0.5$.

Constant Sigma during Deployment

A consistent sigma may be required during deployment, presenting a limitation for input-specific RS. The memory bank strategy proposed by DDRS (Alfarra et al. 2022), compatible with our QCRS, can address this issue. It is crucial to understand QCRS still can apply a consistent sigma during deployment, and it does not undermine the soundness of certification, though it may limit its practical utility in real-world scenarios. Specifically, in binary scenarios, the proposed method enables the customization of an optimal sigma for specific classes, thereby effectively enhancing their certified radii. For further details, please see Appendix J.

Fairness Issue

As discussed in (Mohapatra et al. 2021), randomized smoothing suffers from fairness issue. The unbounded class in the data space dominates as σ increases. We investigate the optimal σ for each class in CIFAR-10. As Table 5 illustrates, class "cars" exhibits a larger optimal sigma value compared to the other classes. This discrepancy arises from the possibility that "cars" represents the unbounded class; thus, an increase in the sigma value leads to a continuous expansion of the radius. Exploring potential solutions to address this issue remains an intriguing direction.

Conclusion

In this work, we exploit and empirically demonstrate the quasiconcavity of the sigma-radius curve. The (υ^-, υ^+)-SQC condition is general and easy to satisfy. Therefore, most data points (approximately 99%) conform to this condition. Based on the (υ^-, υ^+)-SQC condition, we develop an efficient input-specific method called **QCRS** to efficiently find the optimal σ used for building the smoothed classifier, enhancing the traditional randomized smoothing significantly. Unlike former inference-time randomized smoothing methods that suffer from marginal improvement or high computational overhead, the proposed method enjoys better certification results and lower cost. We conducted extensive experiments on CIFAR-10 and ImageNet, and the results show that the proposed method significantly boosts the average certified radius with 7% overhead. QCRS improves ACR, overcoming the trade-off on the radius-accuracy curve and eliminating the truncation effect. In addition, we combine the proposed QCRS with a training-based technique, and the results demonstrate the state-of-the-art average certified radii on CIFAR-10 and ImageNet. A direction for future work is to generalize the proposed method to ℓ_p ball and different distributions. A better training approach for QCRS is also an interesting future research direction.

Acknowledgements

This work was supported in part by the National Science and Technology Council under Grants MOST 110-2634-F002-051, MOST 110-2222-E-002-014-MY3, NSTC 113-2923-E-002-010-MY2, NSTC-112-2634-F-002-002-MBK, by National Taiwan University under Grant NTU-CC-112L891006, and by Center of Data Intelligence: Technologies, Applications, and Systems under Grant NTU-112L900903.

References

Alfarra, M.; Bibi, A.; Torr, P. H.; and Ghanem, B. 2022. Data dependent randomized smoothing. In *Uncertainty in Artificial Intelligence (UAI)*, 64–74. PMLR.

Anderson, B. G.; and Sojoudi, S. 2022. Certified robustness via locally biased randomized smoothing. In *Learning for Dynamics and Control Conference*, 207–220. PMLR.

Boyd, S. P.; and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Carlini, N.; and Wagner, D. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE security and privacy workshops (SPW)*, 1–7. IEEE.

Chen, C.; Kong, K.; Yu, P.; Luque, J.; Goldstein, T.; and Huang, F. 2021. Insta-RS: Instance-wise Randomized Smoothing for Improved Robustness and Accuracy. *arXiv preprint arXiv:2103.04436*.

Chen, R.; Li, J.; Yan, J.; Li, P.; and Sheng, B. 2022. Input-specific robustness certification for randomized smoothing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 6295–6303.

Cohen, J.; Rosenfeld, E.; and Kolter, Z. 2019. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning (ICML)*, 1310–1320. PMLR.

Das, N.; Shanbhogue, M.; Chen, S.-T.; Hohman, F.; Li, S.; Chen, L.; Kounavis, M. E.; and Chau, D. H. 2018. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 196–204.

Ehlers, R. 2017. Formal verification of piece-wise linear feed-forward neural networks. In *International Symposium on Automated Technology for Verification and Analysis*, 269–286. Springer.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Gowal, S.; Dvijotham, K.; Stanforth, R.; Bunel, R.; Qin, C.; Uesato, J.; Arandjelovic, R.; Mann, T.; and Kohli, P. 2018. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*.

Jeong, J.; Park, S.; Kim, M.; Lee, H.-C.; Kim, D.; and Shin, J. 2021. SmoothMix: Training Confidence-calibrated Smoothed Classifiers for Certified Adversarial Robustness. In *ICML 2021 Workshop on Adversarial Machine Learning*.

Katz, G.; Barrett, C.; Dill, D. L.; Julian, K.; and Kochenderfer, M. J. 2017. Reluplex: An efficient SMT solver for verifying deep neural networks. In *International conference on computer aided verification*, 97–117. Springer.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. *Citeseer*.

Kumar, A.; Levine, A.; Feizi, S.; and Goldstein, T. 2020a. Certifying confidence via randomized smoothing. *Advances in Neural Information Processing Systems (NeurIPS)*, 33: 5165–5177.

Kumar, A.; Levine, A.; Goldstein, T.; and Feizi, S. 2020b. Curse of dimensionality on randomized smoothing for certifiable robustness. In *International Conference on Machine Learning (ICML)*, 5458–5467. PMLR.

Lecuyer, M.; Atlidakis, V.; Geambasu, R.; Hsu, D.; and Jana, S. 2019. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, 656–672. IEEE.

Levine, A.; Kumar, A.; Goldstein, T.; and Feizi, S. 2020. Tight Second-Order Certificates for Randomized Smoothing. *arXiv preprint arXiv:2010.10549*.

Levine, A. J.; and Feizi, S. 2021. Improved, Deterministic Smoothing for L_1 Certified Robustness. In *International Conference on Machine Learning (ICML)*, 6254–6264. PMLR.

Li, L.; Xie, T.; and Li, B. 2023. Sok: Certified robustness for deep neural networks. In *2023 IEEE Symposium on Security and Privacy (SP)*, 1289–1310. IEEE.

Li, L.; Zhang, J.; Xie, T.; and Li, B. 2022. Double Sampling Randomized Smoothing. In *International Conference on Machine Learning (ICML)*, 13163–13208. PMLR.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Mohapatra, J.; Ko, C.-Y.; Weng, L.; Chen, P.-Y.; Liu, S.; and Daniel, L. 2021. Hidden cost of randomized smoothing. In *International Conference on Artificial Intelligence and Statistics*, 4033–4041. PMLR.

Mueller, M. N.; Eckert, F.; Fischer, M.; and Vechev, M. 2022. Certified Training: Small Boxes are All You Need. In *International Conference on Learning Representations (ICLR)*.

Nesterov, Y. 2018. *Lectures on convex optimization*, volume 137. Springer.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision (IJCV)*, 115(3): 211–252.

Salman, H.; Li, J.; Razenshteyn, I.; Zhang, P.; Zhang, H.; Bubeck, S.; and Yang, G. 2019. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in Neural Information Processing Systems (NeurIPS)*, 32.

Samangouei, P.; Kabkab, M.; and Chellappa, R. 2018. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations (ICLR)*.

Shafahi, A.; Najibi, M.; Ghiasi, M. A.; Xu, Z.; Dickerson, J.; Studer, C.; Davis, L. S.; Taylor, G.; and Goldstein, T. 2019. Adversarial training for free! *Advances in Neural Information Processing Systems (NeurIPS)*, 32.

Singla, S.; and Feizi, S. 2021. Skew orthogonal convolutions. In *International Conference on Machine Learning (ICML)*, 9756–9766. PMLR.

Singla, S.; Singla, S.; and Feizi, S. 2022. Improved deterministic l2 robustness on CIFAR-10 and CIFAR-100. In *International Conference on Learning Representations (ICLR)*.

Súkeník, P.; Kuvshinov, A.; and Günnemann, S. 2021. Intriguing Properties of Input-dependent Randomized Smoothing. *arXiv preprint arXiv:2110.05365*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*.

Tjeng, V.; Xiao, K. Y.; and Tedrake, R. 2018. Evaluating Robustness of Neural Networks with Mixed Integer Programming. In *International Conference on Learning Representations (ICLR)*.

Wang, C.-Y.; Bochkovskiy, A.; and Liao, H.-Y. M. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*.

Weng, L.; Zhang, H.; Chen, H.; Song, Z.; Hsieh, C.-J.; Daniel, L.; Boning, D.; and Dhillon, I. 2018. Towards fast computation of certified robustness for relu networks. In *International Conference on Machine Learning (ICML)*, 5276–5285. PMLR.

Wong, E.; Rice, L.; and Kolter, J. Z. 2020. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*.

Xu, X.; Li, L.; and Li, B. 2022. Lot: Layer-wise orthogonal training on improving l2 certified robustness. *Advances in Neural Information Processing Systems (NeurIPS)*, 35: 18904–18915.

Yang, G.; Duan, T.; Hu, J. E.; Salman, H.; Razenshteyn, I.; and Li, J. 2020. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning (ICML)*, 10693–10705. PMLR.

Zhai, R.; Dan, C.; He, D.; Zhang, H.; Gong, B.; Ravikumar, P.; Hsieh, C.-J.; and Wang, L. 2019. MACER: Attack-free and Scalable Robust Training via Maximizing Certified Radius. In *International Conference on Learning Representations (ICLR)*.

Zhai, X.; Kolesnikov, A.; Houlsby, N.; and Beyer, L. 2022. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12104–12113.

Zhang, D.; Ye, M.; Gong, C.; Zhu, Z.; and Liu, Q. 2020. Black-box certification with randomized smoothing: A functional optimization based framework. *Advances in Neural Information Processing Systems (NeurIPS)*, 33: 2316–2326.