QCRS: IMPROVE RANDOMIZED SMOOTHING USING QUASI-CONCAVE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Randomized smoothing is currently the state-of-the-art method that provides certified robustness for neural networks. However, it often cannot achieve an adequate certified region on real-world datasets. One way to obtain a larger certified region 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 Cifar10 and ImageNet. The results show that the proposed method significantly enhances the certified radii with low computational overhead.[1]

1 INTRODUCTION

Although deep learning has achieved tremendous success in various fields (Wang et al., 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 et al., 2014) or voices (Carlini & 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 et al., 2020) and preprocessing-based methods (Samangouei et al., 2018; Das et al., 2018). Though effective, empirical defenses cannot guarantee robustness.

Different from empirical defenses, theoretical defenses (certified defense), such as mixed-integer programming (Tjeng et al., 2018), interval bound propagation (Ehlers, 2017; Gowal et al., 2018), and randomized smoothing (Cohen et al., 2019; Lecuyer et al., 2019; Yang et al., 2020), can provide provable defense that theoretically and quantitatively guarantee robustness. The guarantee ensures that there are no adversarial examples within a specific ball with a radius r. Among these 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 region for the class and promise that there is no adversarial example within this region.

Although randomized smoothing is effective, it suffers from two main disadvantages. First, randomized smoothing uses a constant-variance Gaussian filter for every data point when building a smoothed classifier. This makes the certified region 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 region. As a result, when evaluating certified accuracy using the radius-accuracy curve that illustrates the certified accuracy under different radii, a truncation fall often occurs. This is called truncation effect or waterfall effect (Súkeník et al., 2021), which shows the conservation aspect in randomized smoothing. Other issues such as fairness

[1] Under review. Code will be made available after acceptance.

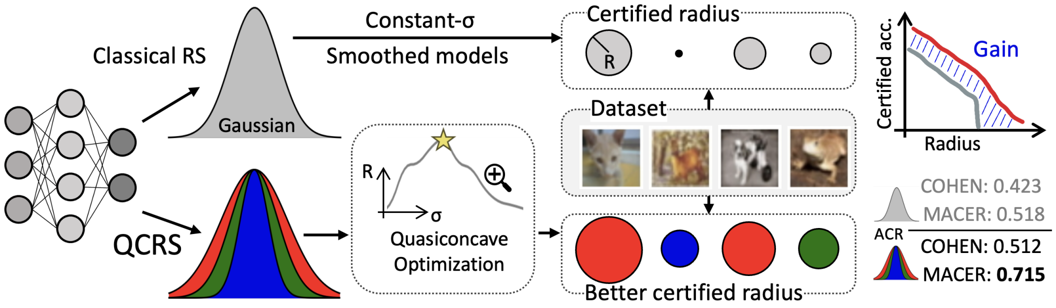


Figure 1: Overview of the proposed QCRS algorithm. QCRS optimizes each smoothed classifier using quasiconcave optimization and provides much better certified radii than the classical randomized smoothing. In this paper, we discuss quasiconcavity on the certified radius w.r.t. σ. i.e., $R(\sigma)$.

(Mohapatra et al., 2021), dimension (Kumar et al., 2020b), and time-efficiency (Chen et al., 2022) also limit the application of randomized smoothing.

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-specific methods that can assign different Gaussian filters to different data points. Those methods try 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, based on quasiconcavity, we develop a novel algorithm called **Quasiconvexity-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 region 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 prove that the sigma-radius curves are quasiconcave for most data points. In addition, we also show that the necessary condition for quasiconcavity is more general and easier to satisfy than the conditions proposed by prior work. In our experiments, $\sim 99\%$ data points satisfy our proposed quasiconcavity condition.
- Based on the observed quasiconcavity property, we propose a novel and efficient input-specific algorithm QCRS to improve the traditional randomized smoothing. QCRS enhances the certified radii and alleviates the truncation effect.
- We conduct extensive experiments, showing the effectiveness of the proposed method on CIFAR-10 and ImageNet. In addition, we combine QCRS with a training-based method and achieve the state-of-the-art certified radii.

2 Related Works

Randomized smoothing utilizes a spatial low-pass Gaussian filter to construct a smoothed model (Cohen et al., 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, Yang et al. (2020); Zhang et al. (2020); Levine & 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 & 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 (Cohen et al., 2019).

Improving RS during inference phase. Different from training-based method, some works utilized different smoothing methods to enhance the certified region. Chen et al. (2021) proposed a multiple-start search algorithm to find the best parameter for building smoothed classifiers. Súkeník et al. (2021) demonstrated the curse of dimensionality for input-dependent smoothing and provided a practical input-specific method to deal with that issue. Alfarra et al. (2022) adopted a memory-based approach to optimize the Gaussian filter of each input data. 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. (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. See Section 4.1 for more detailed description on these methods.

3 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 et al., 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. (2019) (COHEN) proposed a provable method to calculate the certifiable robust region as follows:

$$R = \frac{\sigma}{2} \cdot [\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})], \quad p_A = Pr[f(x+\epsilon) = e_A] \text{ and } p_B = Pr[f(x+\epsilon) = e_B], \tag{3}$$

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 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. That is, any data point inside this region would be predicted as class A by the smoothed classifier. In practice, Cohen et al. (2019) replace $\overline{p_B}$ 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 region in this data point according to COHEN.

Randomized smoothing returns the highest-probability class predicted by the base classifier when perturbations ϵ are added to x. Therefore, smoothed classifier g can be regarded as a spatial smoothing measure of the original base classifier using a Gaussian kernel $\mathcal{G}$, *i.e.,* $f = g \star \mathcal{G}$. Randomized smoothing constructs smoothed classifier to provide certifiable robustness guarantee.

4 QCRS METHODOLOGY

4.1 OBSERVATION AND MOTIVATION

Traditional randomized smoothing suffers from limited certified region and truncation effect, which degrade the certification performance. Several existing methods try to address these issues. Some focus on training the base model to enlarge certified radii, while others use a different Gaussian kernel $\mathcal{G}$ 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 y, if most neighboring points belong to the same class y, we can use $\mathcal{G}$ with a larger variance to convolute x. In contrast, if the neighborhood is full of different class samples, $\mathcal{G}$ needs a small variance to prevent misclassification. Below, we first describe some input-specific search algorithms used in prior work (Alfarra et al., 2022; Chen et al., 2021).

Alfarra et al. (2022) assume that sigma-radius curves are concave and use gradient-based convex optimization along with some relaxation and approximation to find the σ value that provides maximum certified radii. However, in our observation, almost all sigma-radius curves are not concave. We randomly select 200 images from CIFAR-10 dataset and compute the certified radius with respect to σ for each image (Fig. 2). Among these 200 images, 164 of them can provide valid certified radii, and the other 36 images do not have certified regions. We check the concavity numerically for these 164 curves, i.e., check $Hessian(R) \leq 0$; unfortunately, only 11 images satisfy concavity. That is, 93.29% images are not concave. Thus, the gradient-based convex optimization method may not work well in this task.

Instead of depending on the assumption of concavity, Chen et al. (2021) use a multi-start searching algorithm to optimize σ. However, the multi-start procedure incurs high computational overhead. In this work, we observe an intriguing quasiconcave property on the sigma-radius curves, as Fig. 2 shows. The quasiconcave sigma-radius curves accounts for $\sim$ 99%. Quasiconcavity is a much more general property than those used by prior works. It helps us design a more effective and efficient optimization algorithm than existing methods.

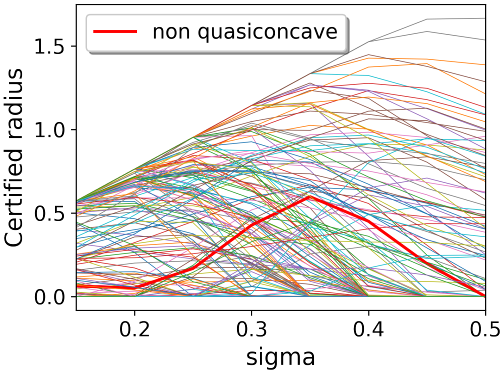


Figure 2: We evaluate the 164 images from CIFAR-10 that can be certified by COHEN, and check their concavity and quasiconcavity numerically: Concave: 6.7%; Quasiconcave: 99.4%.

4.2 QUASICONVEXITY

Quasiconvexity is a generalization of convexity, defined as follows:

Definition 1 *(quasiconvexity and quasiconcavity (Boyd et al., 2004)). A function h is quasiconvex if* $\operatorname{dom} h$ *is convex and for any $\theta \in [0,1]$ and $x, y \in \operatorname{dom} h$,*

$$h(\theta x + (1-\theta)y) \leq \max\{h(x), h(y)\}.$$

Similarly, a function h is quasiconcave if

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

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

$$h(\theta x + (1-\theta)y) < \max\{h(x), h(y)\}.$$

Similarly, a function h is strictly quasiconcave if

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

Quasiconcavity indicates that all values in a segment are not less than the minimum of the endpoints. In this paper, we mainly use strict quasiconcavity. Below, we list lemmas on strict quasiconcavity that we will use later.

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

Lemma 2 *Suppose h is strictly quasiconcave, and let x^* be the optimal solution. Then, the following two statements hold:*

$$\nabla h(x) > 0, \textit{ for } x \in (-\infty, x^*)$$

$$\nabla h(x) < 0, \textit{ for } x \in (x^*, \infty)$$

Lemma 2 illustrates that the gradient must be positive in the left side of the optimal solution.

4.3 DESIGN

In this section, we show quasiconcavity related to sigma-radius curves. Consider $R(\sigma) = \sigma \cdot \Phi^{-1}(\underline{p_A}(\sigma))$. We want to get $\sigma^* = \arg\max_\sigma R(\sigma)$. This σ^* is the optimal solution to maximize $R(\sigma)$. First, we differentiate the objective $R(\sigma)$:

$$\nabla_\sigma R(\sigma) = \frac{\partial R(\sigma)}{\partial \sigma} = \Phi^{-1}(\underline{p_A}(\sigma)) + \sigma \cdot \frac{\partial \Phi^{-1}(\underline{p_A}(\sigma))}{\partial \underline{p_A}(\sigma)} \cdot \frac{\partial \underline{p_A}(\sigma)}{\partial \sigma} \tag{4}$$

According to Lemma 2, if equation 4 is positive for $\sigma < \sigma^*$ and negative for $\sigma > \sigma^*$, the sigma-radius curve is strictly quasiconcave. However, there are some sigma values that can not be certified by randomized smoothing, i.e., $\{\sigma | \underline{p_A}(\sigma) < 0.5\}$. We need to exclude these sigma values because the corresponding smoothed classifiers can not provide any certification. Therefore, we define a new condition based on Lemma 2 as follows:

Definition 2 *(σ-SQC condition) Given a σ^* that satisfies $\nabla R(\sigma^*) = 0$ and $R(\sigma^*) > 0$, we call the sigma-radius curve satisfies σ-strict quasiconcave condition (σ-SQC condition), if for any $\{\sigma | R(\sigma) > 0\}$, $\nabla R(\sigma)$ satisfy the following:*

$$\underset{\sigma<\sigma^*}{Pr}[\nabla R(\sigma) > 0] + \underset{\sigma>\sigma^*}{Pr}[\nabla R(\sigma) < 0] = 2.$$

Intuitively, it illustrates that the slope of sigma-radius curve is positive in the left hand side of optimal solution and negative in the right hand side. Note that this condition is weaker and more general compared to the concentration assumption used in (Li et al., 2022), which restricts the distribution of data points. In addition, it is also weaker to the assumption of concavity (Alfarra et al., 2022). Since σ-SQC condition is weaker, we expect that more data points would satisfy this assumption. In our experiment, there are roughly 99% data points satisfy σ-SQC condition, while only 6.7% data points satisfy the concavity assumption.

We assume that a data point satisfies σ-SQC condition. According to Lemma 2, if we detect that the gradient of a point is positive, we can assert that the 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. If the sigma-radius curve satisfies σ-SQC condition, Algorithm 1 finds the optimal sigma efficiently, which is the global optimal solution according to 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 $\nabla R(\sigma)$ is 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 σ-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.

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.

Therefore, the proposed method is time-efficient compared to Chen et al. (2021); Alfarra et al. (2022). Alfarra et al. (2022) use 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, and the results are shown in Fig 3. The gradient values vary dramatically when using low MC sampling numbers. Therefore, a low MC sampling number may not accurately estimate gradients, which would affect the gradient-based optimization. On the other hand, the proposed QCRS only utilizes the gradient sign, which is much more stable than the gradient value as Fig. 3 shows. The sign hardly changes when the MC sampling number exceeds 500.

Algorithm 1 Bisection Randomized Smoothing

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 $\sigma \leftarrow \arg\max_{\sigma \in \{\hat{\sigma}, \sigma_0\}} R(\sigma)$
```

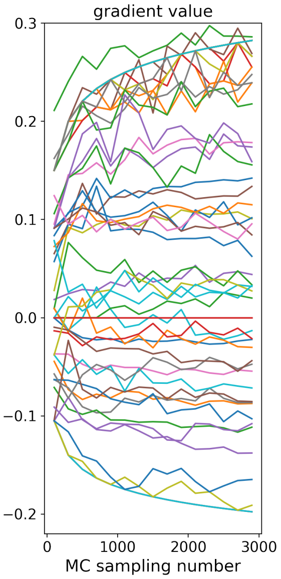


Figure 3: Gradient values with respect to different MC sampling numbers.

4.4 Implementation details

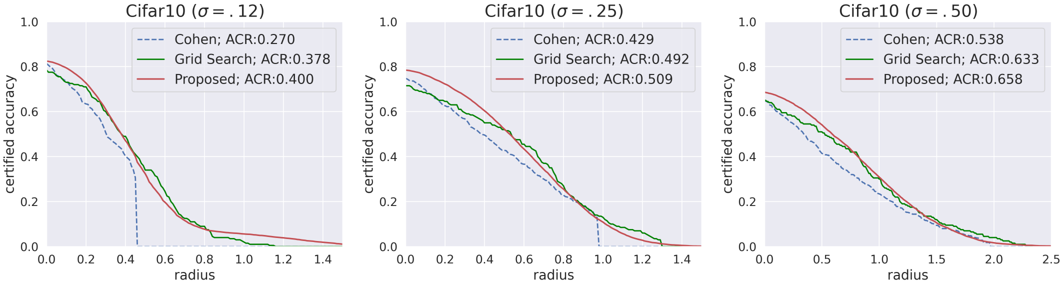
Following prior work, we use ResNet110 for CIFAR-10 and ResNet50 for ImageNet. We use 500 as the MC sampling number to estimate gradients in Algorithm 1. The suboptimal (grid interval) ε is 0.02, and τ (the step to compute gradient) is ± 0.05 in Algorithm 1. Regarding grid search, we use 24 points for CIFAR-10 and 8 points for ImageNet. The searching region is 0.08 to 0.50 for $\sigma = 0.12$, 0.15 to 0.7 for $\sigma = 0.25$, and 0.25 to 1.0 for $\sigma = 0.50$.

5 Experimental results

We evaluate the proposed QCRS and present the experimental results on CIFAR-10 (Krizhevsky 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. Following Zhai et al. (2019), we use average certified radius (ACR) as a 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.

5.1 CIFAR-10

Fig 4 compares the radius-accuracy curves for different methods on the CIFAR-10 dataset. We also show the corresponding ACR, which is also the area under the radius-accuracy curve, in the figure. Table 1 shows the ACR of different methods along with the corresponding runtime cost. The proposed method outperforms the original randomized smoothing (Cohen et al., 2019) significantly. The main performance gain comes from the reduced truncation effect (the waterfall effect) on the radius-accuracy curve. Specifically, QCRS improves Cohen's method by 48%, 18%, and 22% for $\sigma = \{0.12, 0.25, 0.50\}$, respectively. We also compare QCRS to grid search and show the results in Fig. 4 The number of searching points is 24 for each grid search. Since grid search is extremely computationally expensive, we only test the images with $id = 0, 49, 99, ..., 9999$ in CIFAR-10. Although we use 24 points in grid search, which costs 24 times more in runtime than QCRS, we can see that QCRS still outperforms grid search. This is because QCRS is more time-efficient so the searching interval can be much larger than that in grid search. In addition, QCRS guarantees to achieve the optimal as grid search does if σ-SQC condition holds. In terms of the computational cost,



(a) $\sigma = 0.12$ (b) $\sigma = 0.25$ (c) $\sigma = 0.50$

Figure 4: 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.

Table 1: ACR and Time Cost for CIFAR-10.

ACR	$\sigma = .12$	$\sigma = .25$	$\sigma = .50$	Time Cost (Sec.)
COHEN	0.270	0.429	0.538	6.50±0.021
Grid Search	0.378	0.492	0.633	155.80±0.50
QCRS	**0.400**	**0.509**	**0.658**	6.96±0.017

as Table 1 shows, the proposed method only takes about 7% additional inference time compared to the original method proposed by Cohen et al. (2019).

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 for fair comparisons. However, randomized smoothing has random components such as MC sampling, and different works may have subtle parameter selection differences. Although these factors do not affect the results significantly, they still cause small variances in the certification results. Thus, we present the original COHEN baseline results reported in the two papers that we compare to and demonstrate their relative improvement for fair comparisons (Table 2). We can see that the original Cohen's result from these works are different but close. We demonstrate the relative improvement on the certified accuracies under different radii of DSRS and DDRS. As Table 2 shows, for the certified accuracy under radius at 0.5, DSRS and DDRS improve COHEN by 4.9% and 20.0%, 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.

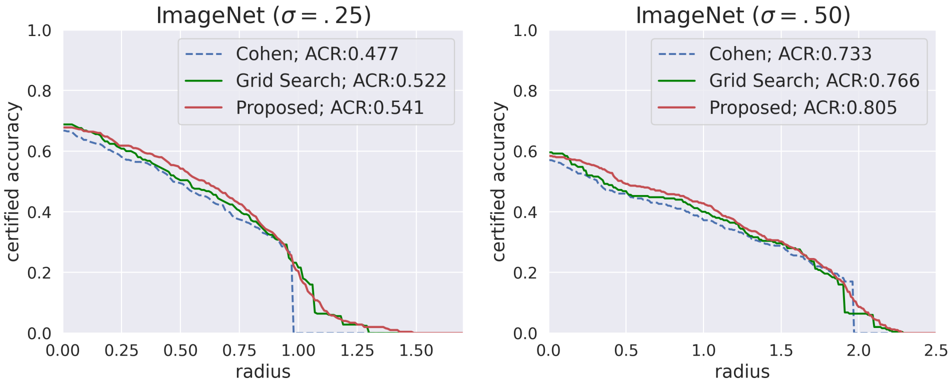
5.2 IMAGENET

Fig 5 shows the results on ImageNet.[2] Following Cohen et al. (2019), only 500 images with $id = 0, 99, 199, ..., 49999$ in the validation set were used. 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 compared to COHEN . Regarding grid search, similar to CIFAR-10, grid search method is computationally expensive, so we set the number of searching points to be 11 on ImageNet. As mentioned earlier, although grid search can provide the optimal certified radius if the cost does not matter, the searching region and precision is limited in practical application. That is the reason why the proposed method can be slightly superior to the brute-force grid search method in Fig 5, while the runtime only accounts for 10% of grid search.

[2] We did not use the model with $\sigma = 1.0$. It is because Mohapatra et al. (2021) has proven that the large σ causes serious fairness issue, σ must be limited in randomized smoothing.

Table 2: Certified accuracy under different radii r of DSRS, DDRS, Grid Search, and the proposed QCRS. ("+%" indicates the relative improvement compared to COHEN.)

	Certified Accuracy								
Certified radii r	0.25	0.5	0.75	1.0	1.25	1.5	1.75	2.0	2.25
COHEN (Li et al., 2022)	0.56	0.41	0.28	0.19	0.15	0.10	0.08	0.04	0.02
+DSRS (Li et al., 2022)	0.57	0.43	0.31	0.21	0.16	0.13	0.08	0.06	0.04
(+%)	1.8%	4.9%	10.7%	10.5%	6.7%	30.0%	0.0%	50%	100%
COHEN (Alfarra et al., 2022)	0.58	0.40	0.29	0.20	0.13	0.07	0.03	0.00	0.00
+DDRS (Alfarra et al., 2022)	0.65	0.48	0.38	0.28	0.17	0.08	0.03	0.01	0.00
(+%)	12.1%	20.0%	31.0%	**40.0%**	30.8%	14.3%	0.0%	NA	0.0%
COHEN (Ours)	0.55	0.41	0.32	0.23	0.15	0.09	0.05	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
(+%)	5.5%	24.4%	31.2%	30.4%	20.0%	**33.3%**	**40%**	NA	NA
COHEN (Ours)	0.55	0.41	0.32	0.23	0.15	0.09	0.05	0.00	0.00
+QCRS (Proposed)	0.64	0.54	0.43	0.31	0.20	0.11	0.05	0.02	0.01
(+%)	**16.4%**	**31.7%**	**34.4%**	34.8%	**33.3%**	22.2%	0.0%	NA	NA



(a) $\sigma = 0.25$

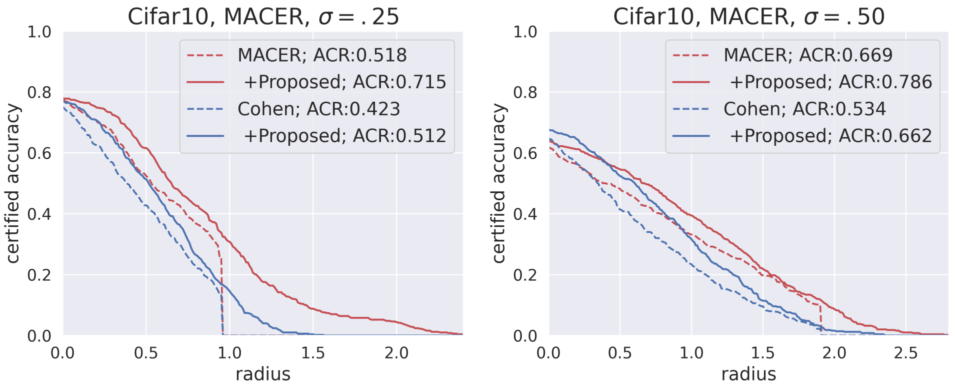
(b) $\sigma = 0.50$

Figure 5: The comparison between COHEN , grid search, and the proposed QCRS on ImageNet. Following Cohen et al. (2019), 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.

5.3 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. We evaluate QCRS on different training weight. 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 models trained by MACER and observe significant improvement in terms of the certified radius. Fig 6 illustrates the results, and Table 3 shows the detailed cross comparison. The last row and the last column show the relative improvement, and the direction is according to the annotated arrow. The bottom right value in the tables are the overall improvement. As Table 3 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%. Therefore, 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 $+47.2\%$.

On the other hand, we can observe that the proposed method and MACER improves the original COHEN to 0.512 and 0.518, respectively. That is to say, the proposed method can enlarge the certified region to the extent that MACER does, but it does not need any training procedure. Note



(a) $\sigma = 0.25$

(b) $\sigma = 0.50$

Figure 6: The performance of QCRS incorporating training-based methods. We use MACER model with (a) $\sigma = 0.25$ and (b) $\sigma = 0.50$. QCRS provides similar improvement to MACER with little computational overhead. Combining QCRS and MACER provides state-of-the-art certified radii.

Table 3: QCRS ACR results incorporating MACER. The models are trained using COHEN or MACER with (a) $\sigma = .25$ and (b) $\sigma = .50$. The arrows illustrate the comparison direction.

(a) $\sigma = 0.25$

	Training		⇒
Test	COHEN	MACER	Improve
Original	0.423	0.518	+22.5%
QCRS	0.512	0.715	+39.7%
⇓ Improve	+21%	+38%	**+69%**

(b) $\sigma = 0.50$

	Training		⇒
Test	COHEN	MACER	Improve
Original	0.534	0.669	+25.3%
QCRS	0.662	0.786	+18.7%
⇓ Improve	+24%	+17.5%	**+47.2%**

that nowadays dataset becomes larger and larger, re-training may be computationally prohibited. Thus, the proposed method benefits from its efficient workflow. It enlarges certified radius with negligible cost.

6 CONCLUSION

In this work, we exploit and prove the quasiconcavity of the sigma-radius curve. σ-SQC condition is general and easy to satisfy. Therefore, most data points ($\sim 99\%$) conform to this condition. Based on σ-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 the 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. Our method overcomes the trade-off in the RS inference phase between clean and robust accuracies on the radius-accuracy curve and eliminates 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.

References

Motasem Alfarra, Adel Bibi, Philip HS Torr, and Bernard Ghanem. Data dependent randomized smoothing. In *Uncertainty in Artificial Intelligence (UAI)*, pp. 64–74. PMLR, 2022.

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

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-totext. In *2018 IEEE security and privacy workshops (SPW)*, pp. 1–7. IEEE, 2018.

Chen Chen, Kezhi Kong, Peihong Yu, Juan Luque, Tom Goldstein, and Furong Huang. Insta-rs: Instance-wise randomized smoothing for improved robustness and accuracy. *arXiv preprint arXiv:2103.04436*, 2021.

Ruoxin Chen, Jie Li, Junchi Yan, Ping Li, and Bin Sheng. Input-specific robustness certification for randomized smoothing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6295–6303, 2022.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning (ICML)*, pp. 1310–1320. PMLR, 2019.

Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E Kounavis, and Duen Horng Chau. 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*, pp. 196–204, 2018.

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

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

Jongheon Jeong, Sejun Park, Minkyu Kim, Heung-Chang Lee, Doguk Kim, and Jinwoo Shin. Smoothmix: Training confidence-calibrated smoothed classifiers for certified adversarial robustness. In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009.

Aounon Kumar, Alexander Levine, Soheil Feizi, and Tom Goldstein. Certifying confidence via randomized smoothing. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:5165–5177, 2020a.

Aounon Kumar, Alexander Levine, Tom Goldstein, and Soheil Feizi. Curse of dimensionality on randomized smoothing for certifiable robustness. In *International Conference on Machine Learning (ICML)*, pp. 5458–5467. PMLR, 2020b.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 656–672. IEEE, 2019.

Alexander Levine, Aounon Kumar, Thomas Goldstein, and Soheil Feizi. Tight second-order certificates for randomized smoothing. *arXiv preprint arXiv:2010.10549*, 2020.

Alexander J Levine and Soheil Feizi. Improved, deterministic smoothing for l_1 certified robustness. In *International Conference on Machine Learning (ICML)*, pp. 6254–6264. PMLR, 2021.

Linyi Li, Jiawei Zhang, Tao Xie, and Bo Li. Double sampling randomized smoothing. In *International Conference on Machine Learning (ICML)*, pp. 13163–13208. PMLR, 2022.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Jeet Mohapatra, Ching-Yun Ko, Lily Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel. Hidden cost of randomized smoothing. In *International Conference on Artificial Intelligence and Statistics*, pp. 4033–4041. PMLR, 2021.

Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision (IJCV)*, 115(3):211–252, 2015.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations (ICLR)*, 2018.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Peter Súkeník, Aleksei Kuvshinov, and Stephan Günnemann. Intriguing properties of input-dependent randomized smoothing. *arXiv preprint arXiv:2110.05365*, 2021.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2013.

Vincent Tjeng, Kai Y Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. In *International Conference on Learning Representations (ICLR)*, 2018.

Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning (ICML)*, pp. 10693–10705. PMLR, 2020.

Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. Macer: Attack-free and scalable robust training via maximizing certified radius. In *International Conference on Learning Representations (ICLR)*, 2019.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12104–12113, 2022.

Dinghuai Zhang, Mao Ye, Chengyue Gong, Zhanxing Zhu, and Qiang Liu. Black-box certification with randomized smoothing: A functional optimization based framework. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:2316–2326, 2020.

A Appendix

A.1 Convergence Analysis

First, we analyze the convergence of the gradient-descent-based methods (Alfarra et al., 2022). Without loss of generality, we discuss convexity here.

Theorem 1 *Suppose a function $R(\sigma)$ is L-smooth for some $L > 0$ with respect to σ. Then, if we run gradient descent for t iterations, it converges as follows (Nesterov et al., 2018):*

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

Theorem 2 *Suppose a function $R(\sigma)$ is L-smooth and μ-strongly convex for some $L, \mu > 0$ with respect to σ, and $\hat{\sigma}$ is the optimal sigma. Then, if we run gradient descent for t iterations, it converges as follows (Nesterov et al., 2018):*

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

Theorem 1 shows the convergence rate under the convex and L-smooth condition. On the other hand, Theorem 2 shows the convergence rate under the L-smooth and μ-convex condition, which is faster but stricter than Theorem 1.

If we want to achieve δ-optimal for σ, i.e., $|\sigma^* - \sigma| \leq \delta$, Theorem 2 demonstrates that R with L-smoothness and μ-strong concavity can guarantee the convergence rate of $\mathcal{O}((\frac{L-\mu}{L+\mu})^t)$, where t is the number of iterations. On the other hand, according to Theorem 1, R with L-smoothness can not guarantee δ-optimal.

Next, we analyze the convergence rate of the proposed method.

Theorem 3 *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:*

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

We prove Theorem 3 as follows:

Proof 1 *Let σ_t be the σ under t iterations. Suppose that R satisfies σ-SQC condition, and there exists a $\sigma^* \in [\sigma_{min}, \sigma_{max}]$. Then, for the first iteration $\sigma_1 = \frac{\sigma_{max}+\sigma_{min}}{2}$, we have*

$$\frac{\sigma_{max} - \sigma_{min}}{2} \geq |\sigma_1 - \sigma^*|,$$

because σ_1 is the midpoint of σ_{min} and σ_{max}. Without loss of generality, we assume $\sigma_{min} \leq \sigma^* \leq \sigma_1$. Thus, according to Algorithm 1, $\sigma_2 = \frac{\sigma_{min}+\sigma_1}{2}$, and

$$\frac{\sigma_{max} - \sigma_{min}}{2^2} \geq |\sigma_2 - \sigma^*|.$$

If we run t iteration, we can conclude that

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

■

Therefore, to achieve δ-optimal, the convergence rate of the proposed method is $\mathcal{O}((\frac{1}{2})^t)$.

Compared with the gradient-descent-based methods DDRS (Alfarra et al., 2022), the proposed method uses much a looser assumption (quasiconcavity), and the convergence rate is $\mathcal{O}((\frac{1}{2})^t)$. DDRS is based on the concave assumption (stricter than quasiconcavity). In addition, only concave assumption can not guarantee any convergence for δ-optimal. Even though L-smoothness holds, which guarantees the convergence for gradient descent, the convergence rate is only $\mathcal{O}(\frac{1}{t})$, and it still cannot achieve δ-optimal. DDRS cannot achieve δ-optimal without L-smooth and μ-strongly concave assumption. Only if both L-smoothness and μ-strong concavity hold, the gradient-descent-based methods can provide $\mathcal{O}((\frac{L-\mu}{L+\mu})^t)$ convergence. That is, the proposed can achieve the optimal sigma using much faster convergence rate and looser data assumption than gradient descent methods such as DDRS (Alfarra et al., 2022).

A.2 Computing the Time Cost

We use NVIDIA GeForce® RTX 3090 and AMD Ryzen 5 5600X with 32GB DRAM to run the time cost experiments in Table 1. For the original RS, it roughly takes 6.5 seconds to certify a datapoint. For the proposed method, it takes 6.96 seconds to compute the optimal smoothed classifier and certify a datapoint. The overhead cost is roughly 7%.

Next, we briefly analyze the computational complexity compared with 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., $|\sigma - \sigma^*| < 0.006$). For each iteration, we need to compute $1,000$ forwards. Thus, for each datapoint, we roughly need additional $6,000$ forwards. The standard RS needs $100,000$ forwards, so the overhead of the proposed QCRS is 6%.

We also briefly analyze the computational complexity compared with Insta-RS (Chen et al., 2021), DDRS (Alfarra et al., 2022), and DSRS (Li et al., 2022). DDRS and DSRA had not provided the code when we submitted this paper. Thus, we cannot compare the time cost directly. For the proposed method and DDRS, the former uses an algorithm of $\mathcal{O}((1/2)^t)$ convergence rate, and the latter uses an algorithm of $\mathcal{O}(1/t)$ convergence rate (assume gradient descent with L-smoothness). In addition, DDRS maintains a memory bank and uses back-propagation several times, which costs a lot. Therefore, we can expect that the time cost of the proposed method is much less than DDRS. On the other hand, compared with DSRS, the author said the running time of DSRS is roughly the same as Cohen's method. In this paper, we show that the proposed method takes about 7% additional inference time. Thus, it is also roughly the same as Cohen's method. Insta-RS adopts multi-start gradient descent, so it must cost a lot.

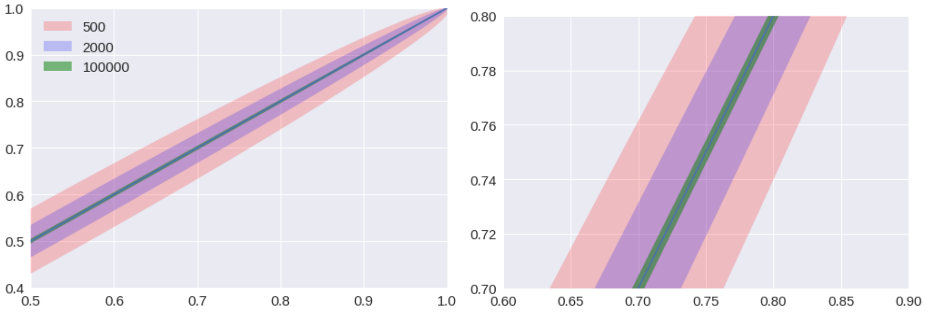
A.3 Quasiconcavity Measurement

Figure 2 is based on standard RS (Cohen). We only consider standard RS in this paper. We sample 20 sigma values to plot Figure 2, listed below: 0.15, 0.18, 0.2 , 0.21, 0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.3, 0.31, 0.32, 0.33, 0.35, 0.4, 0.45, 0.5. Because the model in Figure 2 is trained using $\sigma = 0.25$, the valid sigma values (those can produce a positive certified radius) should be around 0.25. Thus, we increase the sampling density around $\sigma = 0.25$ to check the quasiconcavity.

Regarding Figure 2, we use numerical measurement to verify the quasiconcave condition (according to Lemma 2, we just need to check the sign of gradient on the right/left hand side of optimal σ). Since we want to achieve the 0.01-optimal sigma, we check the quasiconcavity based on the points on the 0.01-grid (gird with $\delta = 0.01$ line-to-line spacing). Therefore, we sample σ in the step size of 0.01. If we decrease δ to check quasiconcavity, the δ-optimal optimization becomes more accurate but the quasiconcave condition is stricter. There is a trade-off to choose δ.

A.4 Gradient Stability

The number of MC sampling affects the estimation of $\underline{p_A}(\sigma)$ significantly. As Fig. 7 shown, if the sampling number is 500, the possible interval is the red region with confidence level $1 - \alpha$. The red region is very large, resulting in the uncertainty for the estimation of $\underline{p_A}(\sigma)$. That is, the estimation of $\underline{p_A}(\sigma)$ is very unstable. Due to expansive computational costs, prior work relied on backpropagation usually uses very low sampling numbers. Therefore, we assert that their computed gradient is unstable, which may lead to poor optimization for σ.



(a) MC sampling (b) Zoom in

Figure 7

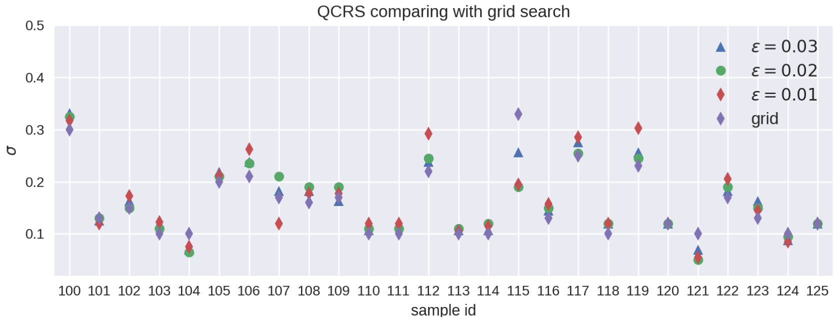


Figure 8

A.5 Error on sigma

We assume the optimal sigma found by grid search is the ground truth optimal. Thus, we compare the optimal sigma found by QCRS and grid search. We randomly select some images, and Fig 8 illustrates the results. The sigma found by QCRS is close to those found by grid search.