INCREMENTAL RANDOMIZED SMOOTHING CERTIFICATION

ABSTRACT

Randomized smoothing-based certification is an effective approach for obtaining robustness certificates of deep neural networks (DNNs) against adversarial attacks. This method constructs a smoothed DNN model and certifies its robustness through statistical sampling, but it is computationally expensive, especially when certifying with a large number of samples. Furthermore, when the smoothed model is modified (e.g., quantized or pruned), certification guarantees may not hold for the modified DNN, and recertifying from scratch can be prohibitively expensive.

We present the first approach for incremental robustness certification for randomized smoothing, IRS. We show how to reuse the certification guarantees for the original smoothed model to certify an approximated model with very few samples. IRS significantly reduces the computational cost of certifying modified DNNs while maintaining strong robustness guarantees. We experimentally demonstrate the effectiveness of our approach, showing up to 4.1x certification speedup over the certification that applies randomized smoothing of approximate model from scratch.

1 INTRODUCTION

Ensuring the robustness of deep neural networks (DNNs) to input perturbations is gaining increased attention from both users and regulators in various application domains (Bojarski et al., 2016; Amato et al., 2013; Julian et al., 2018; ISO). Out of many techniques for obtaining robustness certificaties (Kwiatkowska and Zhang, 2023), statistical methods currently offer the greatest scalability. Randomized smoothing (RS) is a popular statistical certification method by constructing a smoothed model g from a base network f under noise (Cohen et al., 2019). To certify the model g on an input, RS certification checks if the estimated lower bound on the probability of the top class is greater than the upper bound on the probability of the runner-up class (with high confidence). RS certification computes the certified accuracy metric of the DNN on the set of test inputs as a proxy for the DNN robustness. However, despite its effectiveness, RS-based certification can be computationally expensive as it requires DNN inference on a large number of corruptions per input.

The high cost of certification complicates the DNN deployment process, which has become increasingly iterative: the networks are often modified post-training to improve their execution time and/or accuracy. Especially, deploying DNNs on real-world systems with bounded computing resources (e.g., edge devices or GPUs with limited memory), have led to various techniques for approximating DNNs. Common approximation techniques include quantization – reducing the numerical precision of weights (Fiesler et al., 1990; Balzer et al., 1991), and pruning – removing weights that have minimal impact on accuracy (Janowsky, 1989; Reed, 1993).

Common to all of these approximations is that the network behavior (e.g., the classifications) remains the same on most inputs, its architecture does not change, and many weights are only slightly changed. When a user seeks to select a robust and accurate DNN from these possible approximations, RS needs to be performed to compute the robustness of all candidate networks. For instance, in the context of approximation tuning, there are multiple choices for approximation where different quantization or pruning strategies are applied at different layers. Tools such as (Chen et al., 2018b;a; Sharif et al., 2019) use approximations iteratively and test the network at each step. To ensure DNN robustness when using such tools, one would need to check certified accuracy, computed using RS on test data in each step. However, performing RS to compute certified accuracy from scratch can take hours as shown in our experiments even for a single network (with only 500 test images). Therefore, a major encumbrance of almost all existing RS-based certification practices in the above setting, is that *the* *expensive certification needs to be re-run from scratch* for each approximate network. Overcoming this main limitation requires addressing the following fundamental problem:

> How can we leverage the information generated while certifying a given network to speed up the certification of similar networks?

This Work. We present the first incremental RS-based certification framework called Incremental Randomized Smoothing (IRS) to answer this question. The primary objective of our work is to improve the sample complexity of the certification process of similar networks on a predefined test set. Improved sample complexity results in overall speedup in certification, and it reduces the energy requirement and memory footprint of the certification. Given a network f and its smoothed version g, and a modified network f^p with its smoothed version g^p, IRS incrementally certifies the robustness of g^p by reusing the information from the execution of RS certification on g.

IRS optimizes the process of certifying the robustness of smoothed classifier g^p on an input x, by estimating *the disparity* ζ_x – the upper bound on the probability that outputs of f and f^p are distinct. Our new algorithm is based on three key insights about disparity:

1. Common approximations yield small ζ_x values – for instance, it is below 0.01 for int8 quantization for multiple large networks in our experiments.
2. Estimating ζ_x through binomial confidence interval requires fewer samples as it is close to 0 – it is, therefore, less expensive to certify with this probability than directly working with lower and upper probability bounds in the original RS algorithm.
3. We can leverage ζ_x alongside the bounds in the certified radius of g around x to compute the certified radius for g^p – thus soundly reusing the samples from certifying g.

We extensively evaluate the performance of IRS when applying several common DNN approximations such as pruning and quantization on state-of-the-art DNNs on CIFAR10 (ResNet-20, ResNet-110) and ImageNet (ResNet-50) datasets.

Contributions. The main contributions of this paper are:

- We propose a novel concept of incremental RS certification of the robustness of the updated smoothed classifier by reusing the certification guarantees for the original smoothed classifier.
- We design the first algorithm IRS for incremental RS that efficiently computes the certified radius of the updated smoothed classifier.
- We present an extensive evaluation of the performance of IRS speedups of up to 4.1x over the standard non-incremental RS baseline on state-of-the-art classification models.

2 Background

Randomized Smoothing. Let $f : \mathbb{R}^m \to \mathcal{Y}$ be an ordinary classifier. A smoothed classifier $g : \mathbb{R}^m \to \mathcal{Y}$ can be obtained from calculating the most likely result of $f(x+\epsilon)$ where $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$.

$$g(x) := \underset{c \in \mathcal{Y}}{\arg\max}\, \mathbb{P}_\epsilon(f(x+\epsilon) = c)$$

The smoothed network g satisfies following guarantee for a single network f:

Theorem 1. *[From (Cohen et al., 2019)] Suppose $c_A \in \mathcal{Y}$, $\underline{p_A}, \overline{p_B} \in [0,1]$. if*

$$\mathbb{P}_\epsilon(f(x+\epsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}_\epsilon(f(x+\epsilon) = c),$$

then $g(x+\delta) = c_A$ for all δ satisying $\|\delta\|_2 \leq \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B}))$, where Φ^{-1} denotes the inverse of the standard Gaussian CDF.

Computing the exact probabilities $P_\epsilon(f(x+\epsilon) = c)$ is generally intractable. Thus, for practical applications, CERTIFY (Cohen et al., 2019) (Algorithm 1) utilizes sampling: First, it takes n_0 samples to determine the majority class, then n samples to compute a lower bound $\underline{p_A}$ to the success probability with confidence $1-\alpha$ via the Clopper-Pearson lemma (Clopper and Pearson, 1934). If $\underline{p_A} > 0.5$, we set $\overline{p_B} = 1 - \underline{p_A}$ and obtain radius $R = \sigma \cdot \epsilon \cdot \Phi^{-1}(\underline{p_A})$ via Theorem 1, else we return ABSTAIN.

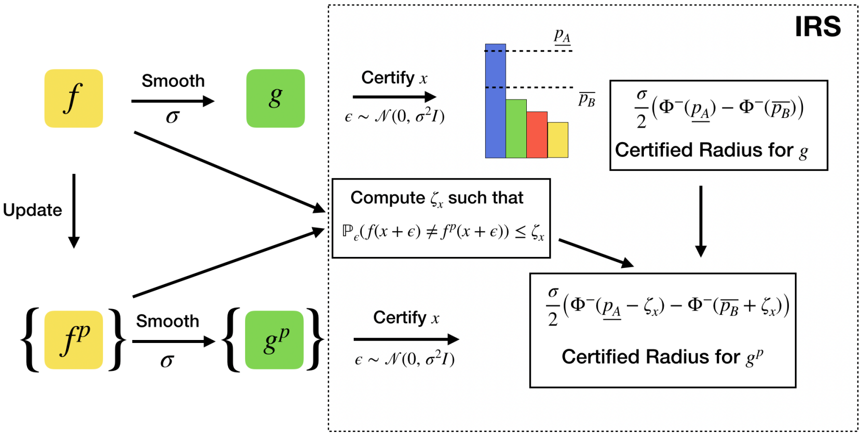

IRS
f
Smooth
σ
g
Certify x
$\epsilon \sim \mathcal{N}(0, \sigma^2 I)$
$\underline{p_A}$
$\overline{p_B}$
$\frac{\sigma}{2}(\Phi^{-}(\underline{p_A}) - \Phi^{-}(\overline{p_B}))$
Certified Radius for g
Update
Compute ζ_x such that
$\mathbb{P}_\epsilon(f(x+\epsilon) \neq f^p(x+\epsilon)) \leq \zeta_x$
$\{f^p\}$
Smooth
σ
$\{g^p\}$
Certify x
$\epsilon \sim \mathcal{N}(0, \sigma^2 I)$
$\frac{\sigma}{2}(\Phi^{-}(\underline{p_A} - \zeta_x) - \Phi^{-}(\overline{p_B} + \zeta_x))$
Certified Radius for g^p

Figure 1: Workflow of IRS from left to right. IRS takes the classifier f and input x. IRS reuses the $\underline{p_A}$ and $\overline{p_B}$ estimates computed for f on x by RS. IRS estimate ζ_x from f and f^p. For the smoothed classifier g^p obtained from any of the approximate classifiers f^p it computes the certified radius by combining $\underline{p_A}$ and $\overline{p_B}$ with ζ_x.

DNN approximation. DNN weights need to be quantized to the appropriate datatype for deploying them on various edge devices. DNN approximations are used to compress the model size at the time of deployment, to allow inference speedup and energy savings without significant accuracy loss. While IRS can work with most of these approximations, for the evaluation, we focus on quantization and pruning as these are the most common ones (Zhou et al., 2017; Frankle and Carbin, 2019).

Algorithm 1 RS certification (Cohen et al., 2019)

Inputs: f: DNN, σ: standard deviation, x: input to the DNN, n_0: number of samples to predict the top class, n: number of samples for computing $\underline{p_A}$, α: confidence parameter

```
function CERTIFY($f, \sigma, x, n_0, n, \alpha$)
    $counts_0 \leftarrow$ SampleUnderNoise$(f, x, n_0, \sigma)$
    $\hat{c}_A \leftarrow$ top index in $counts_0$
    $counts \leftarrow$ SampleUnderNoise$(f, x, n, \sigma)$
    $\underline{p_A} \leftarrow$ LowerConfidenceBound$(counts[\hat{c}_A], n, 1 - \alpha)$
    if $\underline{p_A} > \frac{1}{2}$ then
        return prediction $\hat{c}_A$ and radius $\sigma \cdot \Phi^{-1}(\underline{p_A})$
    else
        return ABSTAIN
```

3 Incremental Randomized Smoothing

Figure 1 illustrates the high-level idea behind the workings of IRS. It takes as input the classifier f, the updated classifier f^p, and an input x. Let g and g^p denote the smoothed network obtained from f and f^p using RS respectively. IRS reuses the $\underline{p_A}$ and $\overline{p_B}$ estimates computed for g to compute the certified radius for g^p.

3.1 Motivation

Insight 1: Similarity in approximate networks We observe that for many practical approximations, f and f^p produce the same result on most inputs. In this experiment, we estimate the disparity between f and f^p on Gaussian corruptions of the input x. We compute a lower confidence bound ζ_x such that $\mathbb{P}_\epsilon(f(x+\epsilon) \neq f^p(x+\epsilon)) \leq \zeta_x$ for $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$.

Table 1: Average ζ_x with $n = 1000$ samples for various approximations.

	CIFAR10 ResNet-110	ImageNet ResNet-50
int8	0.009	0.006
prune10	0.010	0.008

Table 1 presents empirical average ζ_x for int8 quantization and pruning 10% lowest magnitude weights for some of the networks in our experiments computed over 500 inputs. We compute ζ_x value as the binomial confidence upper limit using (Clopper and Pearson, 1934) method with $n = 1000$ samples with $\sigma = 1$. The results show that the ζ_x value is quite small in all the cases.

Insight 2: Sample reduction through ζ_x estimation We demonstrate that ζ_x estimation for approximate networks is more efficient than running certification from scratch. Fig. 2 shows that for the fixed target error χ, confidence $(1 - \alpha)$ and estimation technique, the number of samples required for estimation peaks, when the actual parameter value is around 0.5 and is smallest around the boundaries. For example, when $\chi = 0.5\%$ and $\alpha = 0.01$ estimating the unknown binomial proportion will take $41,500$ samples if the actual parameter value is 0.05 while achieving the same target error and confidence takes $216,900$ samples (5.22x higher) if the actual parameter value is 0.5. As observed in the previous section, ζ_x's value for many practical approximations is close to 0.

Leveraging the observation shown in Fig. 2 and given actual value ζ_x is close to 0, estimating ζ_x with existing binomial proportion estimation techniques is efficient and requires a smaller number of samples. In Appendix A.7, we show the distribution of $\underline{p_A}$ and $\overline{p_B}$ for various cases. We see that $\underline{p_A}$ and $\overline{p_B}$ do not always lie close to 0 or 1 and have a more dispersed distribution. Thus, estimating those requires more samples. Prior work (Thulin, 2013) has theoretically shown that the expected length of the confidence interval for Clopper-Pearson follows a similar trend as in Fig. 2. This theoretical result supports our observation. We show in Appendix A.1 that this observation is not contingent on a specific estimation method and holds for other popular estimation techniques, e.g., (Wilson, 1927), (Agresti and Coull, 1998).

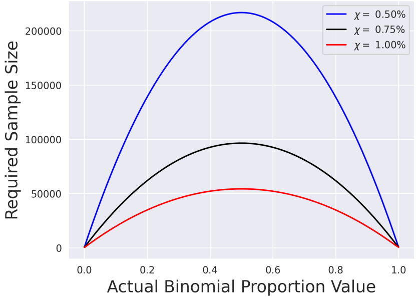


Figure 2: The number of samples for the Clopper-Pearson method to achieve a target error χ with confidence $(1-\alpha)$.

Insight 3: Computing the approximate network's certified radius using ζ_x For certification of the approximate network g^p, our main insight is that estimating ζ_x and using that value to compute the certified radius is more efficient than computing RS certified radius from scratch. The next theorem shows how to use estimated value of ζ_x to certify g^p (the proof is in Appendix A.2):

Theorem 2. *If a classifier f^p is such that for all $x \in \mathbb{R}^m, \mathbb{P}_\epsilon(f(x+\epsilon) \neq f^p(x+\epsilon)) \leq \zeta_x$, and classifier f satisfies $\mathbb{P}_\epsilon(f(x+\epsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}_\epsilon(f(x+\epsilon) = c)$ and $\underline{p_A} - \zeta_x \geq \overline{p_B} + \zeta_x$ then g^p satisfies $g^p(x+\delta) = c_A$ for all δ satisying $\|\delta\|_2 \leq \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A} - \zeta_x) - \Phi^{-1}(\overline{p_B} + \zeta_x))$*

Theorem 1 considers standard RS for a single network. Our Theorem 2 shows how to use the estimated value of ζ_x to transfer the certification guarantees across two networks f and f^p.

3.2 IRS Certification Algorithm

The Algorithm 2 presents the pseudocode for the IRS algorithm, which extends RS certification from Algorithm 1. The algorithm takes the modified classifier f^p and certifies the robustness of g^p around x. The input n_p denotes the number of Gaussian corruptions used by the algorithm.

The IRS algorithm utilizes a cache $\mathcal{C}_f$, which stores information obtained from the RS execution of the classifier f for each input x. The cached information is crucial for the operation of IRS. $\mathcal{C}_f$ stores the top predicted class index $\hat{c}_A$ and its lower confidence bound $\underline{p_A}$ for f on input x.

The standard RS algorithm takes a conservative value of $\overline{p_B}$ by letting $\overline{p_B} = 1 - \underline{p_A}$. This works reasonably well in practice and simplifies the computation of certified radius $\frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B}))$ to $\sigma\Phi^{-1}(\underline{p_A})$. We make a similar choice in IRS, which simplifies the certified radius calculation from $\frac{\sigma}{2}(\Phi^{-1}(\underline{p_A} - \zeta_x) - \Phi^{-1}(\overline{p_B} + \zeta_x))$ of Theorem 2 to $\sigma\Phi^{-1}(\underline{p_A} - \zeta_x)$ as we state in the next theorem (the proof is in Appendix A.2):

Algorithm 2 IRS algorithm: Certification with cache

Inputs: f^p: DNN obtained from approximating f, σ: standard deviation, x: input to the DNN, n_p: number of Gaussian samples used for certification, $\mathcal{C}_f$: stores the information to be reused from certification of f, α and α_ζ: confidence parameters, γ: threshold hyperparameter to switch between estimation methods

```
function CERTIFYIRS($f^p, \sigma, x, n_p, \mathcal{C}_f, \alpha, \alpha_\zeta, \gamma$)
    $\hat{c}_A \leftarrow$ top index in $\mathcal{C}_f[x]$
    $\underline{p_A} \leftarrow$ lower confidence of $f$ from $\mathcal{C}_f[x]$
    if $\underline{p_A} < \gamma$ then
        $\zeta_x \leftarrow$ EstimateZeta($f^p, \sigma, x, n_p, \mathcal{C}_f, \alpha_\zeta$)
        if $\underline{p_A} - \zeta_x > \frac{1}{2}$ then
            return prediction $\hat{c}_A$ and radius $\sigma\Phi^{-1}(\underline{p_A} - \zeta_x)$
    else
        $counts \leftarrow$ SampleUnderNoise($f^p, x, n_p, \sigma$)
        $p'_A \leftarrow$ LowerConfidenceBound(
                        $counts[\hat{c}_A], n_p, 1 - (\alpha + \alpha_\zeta)$)
        if $p'_A > \frac{1}{2}$ then
            return prediction $\hat{c}_A$ and radius $\sigma\Phi^{-1}(p'_A)$
    return ABSTAIN
```

Theorem 3. *If $\underline{p_A} - \zeta_x \geq \frac{1}{2}$, then $\sigma\Phi^{-1}(\underline{p_A} - \zeta_x) \leq \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A} - \zeta_x) - \Phi^{-1}(\overline{p_B} + \zeta_x))$*

As per our insight 2 (Section 3.1), binomial confidence interval estimation requires fewer samples for binomial with actual probability close to 0 or 1. IRS can take advtange of this when $\underline{p_A}$ is not close to 1. However, when $\underline{p_A}$ is close to 1 then there is no benefit of using ζ_x-based certified radius for g^p. Therefore, the algorithm uses a threshold hyperparameter γ close to 1 that is used to switch between certified radius from Theorem 2 and standard RS from Theorem 1.

If the $\underline{p_A}$ is less than the threshold γ, then an estimate of ζ_x for classifier f^p and the classifier f is computed using the EstimateZeta function. We discuss EstimateZeta procedure in the next section. If the $\underline{p_A} - \zeta_x$ is greater than $\frac{1}{2}$, then the top predicted class in the cache is returned as the prediction with the radius $\sigma\Phi^{-1}(\underline{p_A} - \zeta_x)$ as computed in Theorem 3.

In case, $\underline{p_A}$ is greater than the threshold γ, similar to standard RS, the IRS algorithm draws n^p samples of $f^p(x + \epsilon)$ by running n^p noise-corrupted copies of x through the classifier f^p. The function SampleUnderNoise(f^p, x, n_p, σ) in the pseudocode draws n_p samples of noise, $\epsilon_1 \ldots \epsilon_{n_p} \sim \mathcal{N}(0, \sigma^2 I)$, runs each $x + \epsilon_i$ through the classifier f^p, and returns a vector of class counts. If the lower confidence bound is greater than $\frac{1}{2}$, the top predicted class is returned as the prediction with a radius based on the lower confidence bound $\underline{p_A}$.

If the function does certify the input in both of the above cases, it returns ABSTAIN.

The hyperparameters α and α_ζ denote confidence of IRS results. The IRS algorithm result is correct with confidence at least $1 - (\alpha + \alpha_\zeta)$. For the case $\underline{p_A} \geq \gamma$, this holds since we follow the same steps as standard RS. The function LowerConfidenceBound($counts[\hat{c}_A], n_p, 1 - (\alpha + \alpha_\zeta)$) in the pseudocode returns a one-sided $1 - (\alpha + \alpha_\zeta)$ lower confidence interval for the Binomial parameter p given a sample $counts[\hat{c}_A] \sim Binomial(n_p, p)$. We next state the theorem that shows the confidence of IRS results in the other case when $\underline{p_A} < \gamma$ (the proof is in Appendix A.2):

Theorem 4. *If $\mathbb{P}_\epsilon(f(x + \epsilon) = f^p(x + \epsilon)) > 1 - \zeta_x$ with confidence at least $1 - \alpha_\zeta$. If classifier f satisfies $\mathbb{P}_\epsilon(f(x + \epsilon) = c_A) \geq \underline{p_A}$ with confidence at least $1 - \alpha$. Then for classifier f^p, $\mathbb{P}_\epsilon(f^p(x + \epsilon) = c_A) \geq \underline{p_A} - \zeta_x$ with confidence at least $1 - (\alpha + \alpha_\zeta)$*

3.3 Estimating the Upper Confidence Bound ζ_x

In this section, we present our method for estimating ζ_x such that $\mathbb{P}_\epsilon(f(x + \epsilon) \neq f^p(x + \epsilon)) \leq \zeta_x$ with high confidence (Algorithm 3). We use the Clopper-Pearson (Clopper and Pearson, 1934) method to estimate the upper confidence bound ζ_x.

Algorithm 3 Estimate ζ_x

Inputs: f^p: DNN obtained from approximating f, σ: standard deviation, x: input to the DNN, n_p: number of Gaussian samples used for estimating ζ_x, $\mathcal{C}_f$: stores the information to be reused from certification of f, α_ζ: confidence parameter
Output: Estimated value of ζ_x

```
function ESTIMATEZETA($f^p, \sigma, x, n_p, \mathcal{C}_f, \alpha_\zeta$)
    $n_\Delta \leftarrow 0$
    seeds $\leftarrow$ seeds for original samples from $\mathcal{C}_f[x]$
    predictions $\leftarrow$ $f$'s predictions on samples from $\mathcal{C}_f[x]$
    for i $\in \{1, \ldots n_p\}$ do
        $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ using seeds[$i$]
        $c_f \leftarrow$ predictions[$i$]
        $c_{f^p} \leftarrow f^p(x + \epsilon)$
        $n_\Delta \leftarrow n_\Delta + I(c_f \neq c_{f^p})$
    return UpperConfidenceBound($n_\Delta, n_p, 1 - \alpha_\zeta$)
```

We store the *seeds* used for randomly generating Gaussian samples while certifying the function f in the cache, and we reuse these seeds to generate the same Gaussian samples. *seeds*$[i]$ stores the seed used for generating i-th sample in the RS execution of f, and *predictions*$[i]$ stores the prediction of f on the corrsponding $x + \epsilon$. We evaluate f^p on each corruption ϵ generated from *seeds* and match them to predictions by f. c_f and c_{f^p} represent the top class prediction by f and f^p respectively. n_Δ is the count of the number of corruptions ϵ such that f and f^p do not match on $x + \epsilon$.

The function UpperConfidenceBound(n_Δ, $n_p, 1 - \alpha_\zeta$) in the pseudocode returns a one-sided $1 - \alpha_\zeta$ upper confidence interval for the Binomial parameter p given a sample $n_\Delta \sim Binomial(n_p, p)$. We compute this upper confidence bound using the Clopper-Pearson method. This is similar to how the lower confidence bound is computed in the standard RS Algorithm 1. It is sound since the Clopper-Pearson method is conservative.

Reusing the seeds for generating noisy samples does not change the certified radius and is 2x faster compared to naive Monte Carlo estimation of ζ_x with fresh Gaussian samples. Storing the seeds used in the cache results in a small memory overhead (less than 2MBs for our largest benchmark). We use the same Gaussian samples for estimations of $\underline{p_A}$ and ζ_x. This is equivalent to estimating two functions, $p(X)$ and $q(X)$, of a random variable X, where the same set of samples of X can be employed for their respective estimations. Theorem 4 makes no assumptions about the independence of estimating $\underline{p_A}$ and ζ_x, thus we can soundly reuse the same Gaussian samples for both estimations.

4 Experimental Methodology

Networks and Datasets. We evaluate IRS on CIFAR-10 (Krizhevsky et al.) and ImageNet (Deng et al., 2009). On each dataset, we use several classifiers, each with a different σ's. For an experiment that adds Gaussian corruptions with σ to the input, we use the network that is trained with Gaussian augmentation with variance σ^2. On CIFAR-10 we use the base classifier a 20-layer and 110-layer residual network. On ImageNet our base classifier is a ResNet-50.

Network Approximations. We evaluate IRS on multiple approximations. We consider float16 (fp16), bfloat16 (bf16), and int8 quantizations (Section 5.1). We show the effectiveness of IRS on pruning approximation in Section 5.3. For int8 quantization, we use dynamic per-channel quantization mode. from (Paszke et al., 2019) library. For float16 and bfloat16 quantization, we change the data type of the DNN weights from float32 to the respective types. We perform float32, float16, and bfloat16 inferences on the GPU and int8 inferences on CPU since Pytorch does not support int8 quantization for GPUs yet (PyTorch). For the pruning experiment, we perform the lowest weight magnitude (LWM) pruning. The top-1 accuracy of the networks used in the evaluation and the approximate networks is discussed in Appendix A.3.

Experimental Setup. We ran experiments on a 48-core Intel Xeon Silver 4214R CPU with 2 NVidia RTX A5000 GPUs. IRS is implemented in Python and uses PyTorch 2.0.1. (Paszke et al., 2019).

Hyperparameters. We use confidence parameters $\alpha = 0.001$ for the certification of g, and $\alpha_\zeta = 0.001$ for the estimation of ζ_x. To establish a fair comparison, we set the baseline confidence with $\alpha_b = \alpha + \alpha_\zeta = 0.002$. This choice ensures that both the baseline and IRS, provide certified radii with equal confidence. We use grid search to choose an effective value for γ. A detailed description of our hyperparameter search and its results are described in Section 5.4.

Average Certified Radius. We compute the certified radius r when the certification algorithm did not abstain and returned the correct class with radius r, for both IRS (Algorithm 2) and the baseline (Algorithm 1). In other cases, we say that the certified radius $r = 0$. We compute the *average certified radius* (ACR) by taking the mean of certified radii computed for inputs in the test set. Higher ACR indicates stronger robustness certification guarantees.

Speedup. IRS is applicable while certifying multiple similar networks, where it can reuse the certification of one of the networks for faster certification of all other similar networks. We demonstrate the effectiveness of IRS by comparing IRS's certification time for these other similar networks with the baseline certification from scratch. We do not include the certification time of the first network in the comparison as it adds the same time for both IRS and baseline.

5 Experimental Results

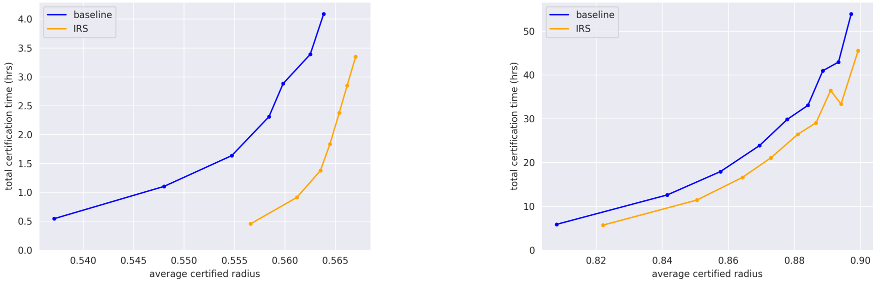
We now present our main evaluation results. We consider the float32 representation of the DNN as f and a particular approximation as f^p. However, IRS can be used with any similar f and f^ps, e.g., where f is an int8 quantized network and f^p is the float32 network. In all of our experiments, we follow a specific procedure:

1. We certify the smoothed classifier g using standard RS with a sample size of n.
2. We approximate the base classifier f with f^p.
3. Using the IRS, we certify smoothed classifier g^p by employing Algorithm 2 and utilizing the cached information $\mathcal{C}_f$ obtained from the certification of g.

We compare IRS to the baseline that uses standard non-incremental RS (Algorithm 1), to certify g^p. Our results compare ACR and certification time between IRS and the baseline for various n_p values.

5.1 Effectiveness of IRS

We compare the ACR and the certification time of the baseline and IRS for the common int8 quantization. We use $n = 10^5$ samples for certification of g. For certifying g^p, we consider n_p values from $\{5\%, \ldots 50\%\}$ of n and $\sigma = 1$. We perform experiments on 500 images and compute the total time for certifying g^p.



(a) ResNet-110 on CIFAR-10

(b) ResNet-50 on ImageNet

Figure 3: Total certification time versus ACR with $\sigma = 1.0$.

Figure 3 presents the comparison between IRS and RS for int8 quantization. The x-axis displays the ACR and the y-axis displays the certification time. The plot consists of 10 markers each for the IRS and the baseline representing a specific value of n_p. Expectedly, the higher the value of n_p, the higher the average time and ACR. The marker coordinate denotes the ACR and the time for an experiment. In all the cases, IRS consistently takes less certification time to obtain the same ACR.

Figure 3a, for ResNet-110 on CIFAR10, shows that IRS reduced the certification time from 2.91 hours (baseline) to 0.71 hours, resulting in time savings of 2.12 hours (4.1x faster). Moreover, we see that IRS achieves an ACR of more than 0.565, whereas the baseline does not reach this ACR for any of the n_p values in our experiments.

Figure 3b, for ResNet-50 on ImageNet, for certifying an ACR of 0.875, IRS substantially reduced certification time from 27.82 hours (baseline) to 22.45 hours, saving approximately 5.36 hours (1.24x faster). Additionally, IRS achieved an ACR of 0.90 and reduced the certification time from 53.93 hours (baseline) to 40.58 hours, resulting in substantial time savings of 13.35 hours (1.33x faster).

5.2 IRS speedups on different quantizations

Next, we study if IRS can handle other kinds of quantization. We perform experiments for 10 different values of n_p along with distinct approximations, and 3 values of σ. Since this would take months of experiment time with n and n_p values from Section 5.1, for the rest of the experiments we use smaller values for these parameters. In these experiments, we compute the relative speedup due to IRS in comparison to the baseline. We use $n = 10^4$ for samples for certification of g. For certifying g^p, we consider n_p values from $\{1\%, \ldots 10\%\}$ of n. For CIFAR10, we consider $\sigma \in \{0.25, 0.5, 1.0\}$, and for ImageNet, we consider $\sigma \in \{0.5, 1.0, 2.0\}$ as in the previous work(Cohen et al., 2019). We validated that the speedups for int8 quantization in this section for ResNet-50-ImageNet and ResNet-110-CIFAR10 are similar to those studied in Section 5.1.

Table 2: Average IRS speedup for combinations of quantizations and σ's.

Dataset	Architecture	σ	Quantization		
			fp16	bf16	int8
CIFAR10	ResNet-20	0.25	1.37x	1.29x	1.3x
		0.5	1.79x	1.7x	1.77x
		1.0	2.85x	2.41x	2.65x
CIFAR10	ResNet-110	0.25	1.42x	1.35x	1.29x
		0.5	1.97x	1.74x	1.77x
		1.0	3.02x	2.6x	2.6x
ImageNet	ResNet-50	0.5	1.2x	1.14x	1.19x
		1.0	1.43x	1.31x	1.43x
		2.0	2.04x	1.69x	1.80x

To quantify IRS's average speedup over the baseline, we employ an approximate area under the curve (AOC) analysis. Specifically, we plot the certification time against the ACR. In most cases, IRS certifies a larger ACR compared to the baseline, resulting in regions on the x-axis where IRS exists but the baseline does not. To ensure a conservative estimation, we calculate the speedup only within the range where both IRS and the baseline exist. We determine the speedup by computing the ratio of the AOC for IRS to the AOC for the baseline within this common range. Table 2 summarizes the average speedups for all quantization experiments.

We observe that IRS gets a larger speedup for smoothing with larger σ since on average the $\underline{p_A}$ values are smaller. Appendix A.7 presents a further justification for this observation. Appendix A.9 presents further experiments with all combinations of DNNs, σ, and quantizations.

5.3 IRS speedups on Pruned Models

In this experiment, we study IRS's ability to certify beyond quantized models. We employ l_1 unstructured pruning, which prunes the fraction of the lowest l_1 magnitude weights from the DNN. Table 3 presents the average IRS speedup for DNNs obtained by pruning $5\%, 10\%$ and 20% weights. The speedups range from 0.99x to 2.7x. As the DNN is pruned more aggressively, it's expected that IRS's speedup will be lower. This is due to higher values of ζ_x associated with aggressive pruning. In Appendix A.4, we provide average ζ_x values for all approximations. Compared to pruning, quantization typically yields smaller ζ_x values, making IRS more effective for quantization.

Table 3: Average IRS speedup for combinations of pruning ratio and σ's.

Dataset	Architecture	σ	Prune		
			5%	10%	20%
CIFAR10	ResNet-20	0.25	1.3x	1.25x	0.99x
		0.5	1.63x	1.39x	1.13x
		1.0	2.5x	2.09x	1.39x
CIFAR10	ResNet-110	0.25	1.35x	1.24x	1.04x
		0.5	1.83x	1.6x	1.23x
		1.0	2.7x	2.25x	1.63x
ImageNet	ResNet-50	0.5	1.19x	1.04x	0.87x
		1.0	1.36x	1.15x	0.87x
		2.0	1.87x	1.54x	1.01x

5.4 Ablation Studies

Next, we show the effect of γ on ACR. In Appendix A.5 we show IRS speedup on distinct values of n.

Sensitivity to threshold γ. For each DNN architecture, we chose the hyperparameter γ by running IRS to certify a small subset of the validation set images for certifying the int8 quantized DNN and comparing the ACR. The choice of γ has no effect on certification time, as we perform n_p inferences in both cases, $\underline{p_A} < \gamma$ and $\underline{p_A} > \gamma$. We use the same γ for each DNN irrespective of the approximation and σ. We use the grid search to choose the best value of gamma from the set $\{0.9, 0.95, 0.975, 0.99, 0.999\}$. Table 4 presents the ACR obtained for each γ. We chose γ as 0.99 for CIFAR10 networks and 0.995 for the ImageNet networks since they result in the highest ACR.

Table 4: ACR for each γ.

γ	CIFAR10	CIFAR10	ImageNet
	ResNet-20	ResNet-110	ResNet-50
0.9	0.438	0.436	0.458
0.95	0.442	0.439	0.464
0.975	0.445	0.441	0.465
0.99	**0.446**	**0.443**	0.466
0.995	0.445	0.442	**0.467**
0.999	0.444	0.442	0.464

6 Related Work

Incremental Program Verification. The scalability of traditional program verification has been significantly improved by incremental verification, which has been applied on an industrial scale (Johnson et al., 2013; O'Hearn, 2018; Stein et al., 2021). Incremental program analysis tasks achieve faster analysis of individual commits by reusing partial results (Yang et al., 2009), constraints (Visser et al., 2012), and precision information (Beyer et al., 2013) from previous runs.

Incremental DNN Certification. Several methods have been introduced in recent years to certify the properties of DNNs deterministically (Tjeng et al., 2017; Bunel et al., 2020; Katz et al., 2017; Wang et al., 2021b; Laurel et al., 2022) and probabilisticly (Cohen et al., 2019). Researchers used incremental certification speed up DNN certification (Fischer et al., 2022b; Ugare et al., 2022; Wei and Liu, 2021; Ugare et al., 2023) – these works apply complete and incomplete deterministic certification using formal logic cannot scale to e.g., ImageNet. In contrast, we propose incremental probabilistic certification with Randomized Smoothing, which enables much greater scalability.

Randomized Smoothing. Cohen et al. (2019) introduced the addition of Gaussian noise to achieve l_2-robustness results. Several extensions to this technique utilize different types of noise distributions and radius calculations to determine certificates for general l_p-balls. Yang et al. (2020) and Zhang et al. (2020) derived recipes for determining certificates for $p = 1, 2,$ and ∞. Lee et al. (2019), Wang et al. (2021a), and Schuchardt et al. (2021) presented extensions to discrete perturbations such as l_0-perturbations, while Bojchevski et al. (2023), Gao et al. (2020), Levine and Feizi (2020), and Liu et al. (2021) explored extensions to graphs, patches, and point cloud manipulations. Dvijotham et al. (2020) presented theoretical derivations for the application of both continuous and discrete smoothing measures, while Mohapatra et al. (2020) improved certificates by using gradient information. Horváth et al. (2022) used ensembles to improve the certificate.

Beyond norm-balls certificates, Fischer et al. (2020) and Li et al. (2021) presented how geometric operations such as rotation or translation can be certified via Randomized Smoothing. yeh Chiang et al. (2022) and Fischer et al. (2022a) demonstrated how the certificates can be extended from the setting of classification to regression (and object detection) and segmentation, respectively. For classification, Jia et al. (2020) extended certificates from just the top-1 class to the top-k classes, while Kumar et al. (2020) certified the confidence of the classifier, not just the top-class prediction. Rosenfeld et al. (2020) used Randomized Smoothing to defend against data poisoning attacks. These RS extensions (using different noise distributions, perturbations, and geometric operations) are orthogonal to the standard RS approach from Cohen et al. (2019). While these extensions have been shown to improve the overall bredth of RS, IRS is complementary to these extensions.

7 Limitations

We showed that IRS is effective at certifying the smoothed version of the approximated DNN. However, there are certain limitations to the effectiveness of IRS. First, the IRS algorithm requires a cache with the top predicted class index, its lower confidence bound, and the seeds for Gaussian corruptions obtained from the RS execution of the original classifier. However, storing this additional information is reasonable since it has negligible memory overhead and is a byproduct of certification (as trustworthy ML matures, we anticipate that this information will be shipped with pre-certified networks for reproducibility purposes).

The smoothing parameter σ used in IRS affects its efficiency, with larger values of σ generally leading to better results. As a consequence, we observed a smaller speedup when using a smaller value of σ (e.g., 0.25 on CIFAR10) compared to a larger value (e.g., 1 on CIFAR10). The value of σ offers a trade-off between robustness and accuracy. By choosing a larger σ, one can improve robustness but it may lead to a loss of accuracy in the model.

IRS targets fast certification while maintaining a sufficiently large radius. Therefore, we considered n_p smaller than 50% of n for our evaluation. However, IRS certified radius can be smaller than the non-incremental RS, provided the user has a larger sample budget. In our experiment in Appendix A.6 we test IRS on larger n_p and observe that IRS is better than baseline for n_p less than 70% of n. This is particularly advantageous when computational resources are limited.

8 Conclusion

We propose IRS, the first incremental approach for probabilistic DNN certification. IRS leverages the certification guarantees obtained from the smoothed model to certify a smoothed approximated model with very few samples. Reusing the computation of original guarantees significantly reduces the computational cost of certification while maintaining strong robustness guarantees. IRS speeds up certification up to 4.1x over the standard non-incremental RS baseline on state-of-the-art classification models. We anticipate that IRS can be particularly useful for approximate tuning when the users need to analyze the robustness of multiple similar networks. Further, one can easily ship the certification cache to allow other users to further modify these networks based on their specific device and application needs and recertify the new network. We believe that our approach paves the way for efficient and effective certification of DNNs in real-world applications.

REFERENCES

Alan Agresti and Brent A. Coull. Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician*, 52(2):119–126, 1998. doi: 10.1080/00031305.1998.10480550. URL https://doi.org/10.1080/00031305.1998.10480550.

Filippo Amato, Alberto López, Eladia María Peña-Méndez, Petr Vaňhara, Aleš Hampl, and Josef Havel. Artificial neural networks in medical diagnosis. *Journal of Applied Biomedicine*, 11(2): 47–58, 2013.

Wolfgang Balzer, Masanobu Takahashi, Jun Ohta, and Kazuo Kyuma. Weight quantization in boltzmann machines. *Neural Networks*, 4(3):405–409, 1991.

Dirk Beyer, Stefan Löwe, Evgeny Novikov, Andreas Stahlbauer, and Philipp Wendler. Precision reuse for efficient regression verification. In *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*, ESEC/FSE 2013, page 389–399, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450322379. doi: 10.1145/2491411.2491429. URL https://doi.org/10.1145/2491411.2491429.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Aleksandar Bojchevski, Johannes Gasteiger, and Stephan Günnemann. Efficient robustness certificates for discrete data: Sparsity-aware randomized smoothing for graphs, images and more, 2023.

Rudy Bunel, Jingyue Lu, Ilker Turkaslan, Pushmeet Kohli, P Torr, and P Mudigonda. Branch and bound for piecewise linear neural network verification. *Journal of Machine Learning Research*, 21 (2020), 2020.

Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Meghan Cowan, Haichen Shen, Leyuan Wang, Yuwei Hu, Luis Ceze, Carlos Guestrin, and Arvind Krishnamurthy. Tvm: An automated end-to-end optimizing compiler for deep learning. In *Proceedings of the 13th USENIX Conference on Operating Systems Design and Implementation*, OSDI'18, page 579–594, USA, 2018a. USENIX Association. ISBN 9781931971478.

Tianqi Chen, Lianmin Zheng, Eddie Yan, Ziheng Jiang, Thierry Moreau, Luis Ceze, Carlos Guestrin, and Arvind Krishnamurthy. Learning to optimize tensor programs. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 3393–3404, Red Hook, NY, USA, 2018b. Curran Associates Inc.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934. ISSN 00063444. URL http://www.jstor.org/stable/2331986.

Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR, 2019. URL http://proceedings.mlr.press/v97/cohen19c.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Krishnamurthy (Dj) Dvijotham, Jamie Hayes, Borja Balle, Zico Kolter, Chongli Qin, Andras Gyorgy, Kai Xiao, Sven Gowal, and Pushmeet Kohli. A framework for robustness certification of smoothed classifiers using f-divergences. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJlKrkSFPH.

Emile Fiesler, Amar Choudry, and H John Caulfield. Weight discretization paradigm for optical neural networks. In *Optical interconnections and networks*, volume 1281, pages 164–173. SPIE, 1990.

Marc Fischer, Maximilian Baader, and Martin Vechev. Certified defense to image transformations via randomized smoothing. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8404–8417. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/5fb37d5bbdbbae16dea2f3104d7f9439-Paper.pdf`.

Marc Fischer, Maximilian Baader, and Martin Vechev. Scalable certified segmentation via randomized smoothing, 2022a.

Marc Fischer, Christian Sprecher, Dimitar I. Dimitrov, Gagandeep Singh, and Martin T. Vechev. Shared certificates for neural network verification. In Sharon Shoham and Yakir Vizel, editors, *Computer Aided Verification - 34th International Conference, CAV 2022, Haifa, Israel, August 7-10, 2022, Proceedings, Part I*, volume 13371 of *Lecture Notes in Computer Science*, pages 127–148. Springer, 2022b. doi: 10.1007/978-3-031-13185-1_7. URL `https://doi.org/10.1007/978-3-031-13185-1_7`.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *Proc. International Conference on Learning Representations (ICLR)*, 2019.

Zhidong Gao, Rui Hu, and Yanmin Gong. Certified robustness of graph classification against topology attack with randomized smoothing. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–6, 2020. doi: 10.1109/GLOBECOM42002.2020.9322576.

Miklós Z. Horváth, Mark Niklas Mueller, Marc Fischer, and Martin Vechev. Boosting randomized smoothing with variance reduced classifiers. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=mHu2vIds_-b`.

ISO. Assessment of the robustness of neural networks. Standard, International Organization for Standardization, March 2021.

Steven A Janowsky. Pruning versus clipping in neural networks. *Physical Review A*, 39(12):6600, 1989.

Jinyuan Jia, Xiaoyu Cao, Binghui Wang, and Neil Zhenqiang Gong. Certified robustness for top-k predictions against adversarial perturbations via randomized smoothing. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=BkeWw6VFwr`.

Kenneth Johnson, Radu Calinescu, and Shinji Kikuchi. An incremental verification framework for component-based software systems. In *Proceedings of the 16th International ACM Sigsoft Symposium on Component-Based Software Engineering*, CBSE '13, page 33–42, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450321228. doi: 10.1145/2465449.2465456. URL `https://doi.org/10.1145/2465449.2465456`.

Kyle D. Julian, Mykel J. Kochenderfer, and Michael P. Owen. Deep neural network compression for aircraft collision avoidance systems. *CoRR*, abs/1810.04240, 2018.

Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In *Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I*, volume 10426 of *Lecture Notes in Computer Science*, 2017. doi: 10.1007/978-3-319-63387-9_5.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). URL `http://www.cs.toronto.edu/~kriz/cifar.html`.

Aounon Kumar, Alexander Levine, Soheil Feizi, and Tom Goldstein. Certifying confidence via randomized smoothing. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5165–5177. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/37aa5dfc44dddd0d19d4311e2c7a0240-Paper.pdf`.

Marta Kwiatkowska and Xiyue Zhang. When to trust ai: Advances and challenges for certification of neural networks. *arXiv preprint arXiv:2309.11196*, 2023.

Jacob Laurel, Rem Yang, Shubham Ugare, Robert Nagel, Gagandeep Singh, and Sasa Misailovic. A general construction for abstract interpretation of higher-order automatic differentiation. *Proc. ACM Program. Lang.*, 6(OOPSLA2), oct 2022. doi: 10.1145/3563324. URL https://doi.org/10.1145/3563324.

Guang-He Lee, Yang Yuan, Shiyu Chang, and Tommi Jaakkola. Tight certificates of adversarial robustness for randomly smoothed classifiers. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/fa2e8c4385712f9a1d24c363a2cbe5b8-Paper.pdf.

Alexander Levine and Soheil Feizi. (de)randomized smoothing for certifiable defense against patch attacks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/47ce0875420b2dbacfc5535f94e68433-Abstract.html.

Linyi Li, Maurice Weber, Xiaojun Xu, Luka Rimanic, Bhavya Kailkhura, Tao Xie, Ce Zhang, and Bo Li. Tss: Transformation-specific smoothing for robustness certification. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 535–557, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384544. doi: 10.1145/3460120.3485258. URL https://doi.org/10.1145/3460120.3485258.

Hongbin Liu, Jinyuan Jia, and Neil Zhenqiang Gong. Pointguard: Provably robust 3d point cloud classification, 2021.

Jeet Mohapatra, Ching-Yun Ko, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel. Higher-order certification for randomized smoothing, 2020.

Peter W. O'Hearn. Continuous reasoning: Scaling the impact of formal methods. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 13–25. ACM, 2018. doi: 10.1145/3209108.3209109. URL https://doi.org/10.1145/3209108.3209109.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

PyTorch. Torch quantization support. https://github.com/pytorch/pytorch/issues/87395.

Russell Reed. Pruning algorithms-a survey. *IEEE transactions on Neural Networks*, 4(5):740–747, 1993.

Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, and J. Zico Kolter. Certified robustness to label-flipping attacks via randomized smoothing, 2020.

Jan Schuchardt, Aleksandar Bojchevski, Johannes Gasteiger, and Stephan Günnemann. Collective robustness certificates: Exploiting interdependence in graph neural networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=ULQdiUTHe3y.

Hashim Sharif, Prakalp Srivastava, Muhammad Huzaifa, Maria Kotsifakou, Keyur Joshi, Yasmin Sarita, Nathan Zhao, Vikram S. Adve, Sasa Misailovic, and Sarita Adve. Approxhpvm: A portable compiler ir for accuracy-aware optimizations. *Proc. ACM Program. Lang.*, 3(OOPSLA), oct 2019. doi: 10.1145/3360612. URL https://doi.org/10.1145/3360612.

Benno Stein, Bor-Yuh Evan Chang, and Manu Sridharan. Demanded abstract interpretation. In Stephen N. Freund and Eran Yahav, editors, *PLDI '21: 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation, Virtual Event, Canada, June 20-25, 2021*, pages 282–295. ACM, 2021. doi: 10.1145/3453483.3454044. URL `https://doi.org/10.1145/3453483.3454044`.

Måns Thulin. The cost of using exact confidence intervals for a binomial proportion. *Electronic Journal of Statistics*, 8, 03 2013. doi: 10.1214/14-EJS909.

Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. *arXiv preprint arXiv:1711.07356*, 2017.

Shubham Ugare, Gagandeep Singh, and Sasa Misailovic. Proof transfer for fast certification of multiple approximate neural networks. *Proc. ACM Program. Lang.*, 6(OOPSLA):1–29, 2022. doi: 10.1145/3527319. URL `https://doi.org/10.1145/3527319`.

Shubham Ugare, Debangshu Banerjee, Sasa Misailovic, and Gagandeep Singh. Incremental verification of neural networks, 2023.

Willem Visser, Jaco Geldenhuys, and Matthew B. Dwyer. Green: Reducing, reusing and recycling constraints in program analysis. In *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, FSE '12, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450316149. doi: 10.1145/2393596.2393665. URL `https://doi.org/10.1145/2393596.2393665`.

Binghui Wang, Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Certified robustness of graph neural networks against adversarial structural perturbation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 1645–1653, New York, NY, USA, 2021a. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467295. URL `https://doi.org/10.1145/3447548.3467295`.

Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. Beta-crown: Efficient bound propagation with per-neuron split constraints for complete and incomplete neural network verification. *arXiv preprint arXiv:2103.06624*, 2021b.

Tianhao Wei and Changliu Liu. Online verification of deep neural networks under domain or weight shift. *CoRR*, abs/2106.12732, 2021. URL `https://arxiv.org/abs/2106.12732`.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. ISSN 01621459. URL `http://www.jstor.org/stable/2276774`.

Greg Yang, Tony Duan, J. Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes, 2020.

Guowei Yang, Matthew B. Dwyer, and Gregg Rothermel. Regression model checking. In *2009 IEEE International Conference on Software Maintenance*, pages 115–124, 2009. doi: 10.1109/ICSM.2009.5306334.

Ping yeh Chiang, Michael J. Curry, Ahmed Abdelkader, Aounon Kumar, John Dickerson, and Tom Goldstein. Detection as regression: Certified object detection by median smoothing, 2022.

Dinghuai Zhang, Mao Ye, Chengyue Gong, Zhanxing Zhu, and Qiang Liu. Black-box certification with randomized smoothing: A functional optimization based framework. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2316–2326. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1896a3bf730516dd643ba67b4c447d36-Paper.pdf`.

Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. Incremental network quantization: Towards lossless CNNs with low-precision weights. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=HyQJ-mclg`.