Enhancing Certifiable Semantic Robustness via Robust Pruning of Deep Neural Networks

Hanjiang Hu* Bowei Li* Ziwei Wang* Tianhao Wei Casidhe Hutchison Eric Sample Changliu Liu

***Abstract*— Deep neural networks have been widely adopted in many vision and robotics applications with visual inputs. It is essential to verify its robustness against semantic transformation perturbations, such as brightness and contrast. However, current certified training and robustness certification methods face the challenge of over-parameterization, which hinders the tightness and scalability due to the over-complicated neural networks. To this end, we first analyze stability and variance of layers and neurons against input perturbation, showing that certifiable robustness can be indicated by a fundamental Unbiased and Smooth Neuron metric (USN). Based on USN, we introduce a novel neural network pruning method that removes neurons with low USN and retains those with high USN, thereby preserving model expressiveness without over-parameterization. To further enhance this pruning process, we propose a new Wasserstein distance loss to ensure that pruned neurons are more concentrated across layers. We validate our approach through extensive experiments on the challenging robust keypoint detection task, which involves realistic brightness and contrast perturbations, demonstrating that our method achieves superior robustness certification performance and efficiency compared to baselines.**

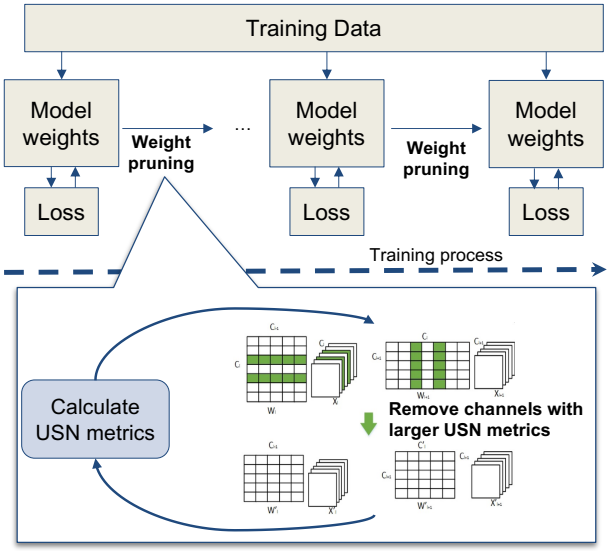


Fig. 1: The overview of model training with the progressive robust pruning pipeline.

I. INTRODUCTION

Deep neural networks (DNNs) have emerged as fundamental components in numerous computer vision and robotics applications, from image classification to pose estimation [7], [17], [26]. In safety-critical scenarios such as autonomous driving and human-robot interaction, these DNN-based systems must maintain reliable performance under various environmental conditions, such as seasonal and daylight changes [12], [13], image corruptions and degradations [15], [16], and sensor placement and perturbations [9], [8]. However, ensuring the robustness of DNNs against such semantic perturbations remains a significant challenge, particularly when formal guarantees are required [20], [11], [10].

Current approaches to neural network robustness have primarily focused on adversarial training [21] and empirical evaluation methods. While these techniques can improve practical robustness, they fall short of providing the formal verification guarantees essential for safety-critical applications. Certified robustness methods [4], [30] address this limitation by providing mathematical guarantees on model behavior within specified perturbation bounds. However, existing robust training and certification techniques face fundamental scalability challenges due to the over-parameterization of modern deep networks, which leads to loose bounds and computational intractability for larger DNNs [18].

The core issue underlying these challenges is that current robust training and certification methods treat all neurons equally, without considering their individual contributions to model robustness. Neurons that exhibit high variance or instability under input perturbations contribute more to the over-approximation errors that fail existing verification algorithms [30], [29]. This observation suggests that strategic removal of problematic neurons through structured pruning could simultaneously improve both certification tightness and computational efficiency.

Neural network pruning has been extensively studied mainly for model compression and acceleration [6], [19], [3], but naive structured pruning may remove important features needed for model expressiveness and robust performance [5]. Although some existing pruning methods demonstrate a correlation with adversarial robustness [25], [14], [32], they do not account for the impact on formal neural network verification based on neuron statistics. The challenge lies in theoretically identifying unstable neurons that are truly detrimental to robustness certification while preserving the model's expressive capacity for the underlying task.

To this end, we propose a novel approach that inte-

*Equal contribution

All authors are with the Robotics Institute, Carnegie Mellon University. Ziwei Wang is also with School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. The work was done while Ziwei Wang was a postdoc fellow at Carnegie Mellon University. hanjianghu@cmu.edu

grates robustness-aware pruning with formal certification guarantees through the introduction of Unbiased and Smooth Neuron (USN) metrics, which quantify the bias and variance characteristics of individual neurons under semantic perturbations. We highlight that the proposed USN metric is a generalized form of the signal-to-noise ratio metrics [29], which additionally bridges the metrics with statistical principles without normalization, focusing on complex semantic perturbation rather than simple ℓ_∞-bounded perturbations [29]. This provides a principled criterion for identifying neurons that contribute most to certification looseness. We develop a progressive training and pruning pipeline that simultaneously optimizes task performance and USN metrics, as shown in Figure 1. Wasserstein distance regularization is further adopted to encourage concentrated pruning patterns while preserving essential representational capacity. As one of the representative regression tasks, we evaluate the proposed method on the challenging keypoint detection task under image-based realistic semantic perturbations, demonstrating that USN-guided pruning consistently outperforms both non-pruned models and random pruning baselines across multiple base architectures and perturbation magnitudes. The main contributions of this work are as follows:

- We establish a theoretical connection between neuron-level statistics and probabilistic robustness certification bounds, providing the foundation for robustness-aware pruning.
- We introduce the Unbiased and Smooth Neuron (USN) metrics that quantify individual neuron contributions to certification tightness under semantic perturbations.
- We propose a progressive training pipeline that integrates USN-guided pruning with Wasserstein distance regularization to achieve concentrated, structure-preserving pruning patterns.
- We demonstrate superior certification performance on keypoint detection tasks against brightness and contrast perturbations, showing that the proposed pruning can simultaneously improve robustness and computational efficiency.

II. PROBLEM FORMULATION

In this section, we first formally formulate the key point detection neural networks as feedforward layers and nonlinear ReLU activation layers. Then, we define the certifiable robustness within the local semantic perturbation set.

A. Neural Networks with Neuron Characterization

Modern certifiable image-based deep neural network models are typically based on ResNet [7], [17], [26], [20], including convolutional neural networks (CNNs) and linear layers with nonlinear activation functions. Since convolution can be seen as sparse matrix multiplication with shared weights of convolutional kernels, the fundamental components of these neural networks for regression tasks (e.g. keypoint detection) are linear layers with nonlinear activation functions, as defined below.

Definition 2.1 (Deep Neural Networks for Regression): Let $f^L : \mathbb{R}^{d_0} \to \mathbb{R}^{d_L}$ be an L-layer feedforward layers with nonlinear activation layers defined as:

$$f^L(x) = g^L \circ \sigma^{L-1} \circ g^{L-1} \circ \sigma^{L-2} \circ \cdots \circ \sigma^1 \circ g^1(x), \quad (1)$$

where each linear layer $g^i : \mathbb{R}^{d_{i-1}} \to \mathbb{R}^{d_i}$ is given by $g^i(z^{i-1}) = W^i z^{i-1} + b^i$ with weight matrix $W^i \in \mathbb{R}^{d_i \times d_{i-1}}$ and bias vector $b^i \in \mathbb{R}^{d_i}$, and activation layer is defined as $z^i = \sigma^i(g^i(z^{i-1}))$ with nonlinear activation function $\sigma^i : \mathbb{R}^{d_i} \to \mathbb{R}^{d_i}$, $i = 1, 2, \ldots, L$ and $z^0 = x$.

Furthermore, we define the output of each neuron on the layer before each activation layer as follows, as the nonlinearity of the neural network in Equation (1) highly depends on the output of pre-activation neurons.

Definition 2.2 (Pre-activation Neuron and Layer Output): Denote the j-th neuron output after the linear layer f^i as $f^i_j(x) = w^i_j \cdot z^{i-1} + b^i_j \in \mathbb{R}$, $j = 0, 1, \ldots, d_i - 1$ where w^i_j is the j-th row of W^i, $z^{i-1} = \sigma(f^{i-1}(x))$ and the layer output $f^i(x)$ is the aggregated vector of $f^i_j(x)$ as $f^i(x) = [f^i_0(x), f^i_1(x), \ldots, f^i_{d_i-1}(x)]^T \in \mathbb{R}^{d_i}$.

B. Semantic Perturbation and Certifiable Robustness

Based on the neural networks in Definition 2.1 with each neuron output in Definition 2.2, we can define the certifiable robustness against semantic perturbation. We first define the semantic perturbation set as follows.

Definition 2.3 (Semantic Perturbation Set): Given an input $x_0 = h(s_0) \in \mathbb{R}^{d_0}$ with a continuous semantic transformation function $h(\cdot) : \mathbb{R}^s \to \mathbb{R}^{d_0}$, define the bounded perturbation set as $B^h_p(x_0, \epsilon) = \{h(s) \in \mathbb{R}^{d_0} : \|s - s_0\|_p \leq \epsilon\}$, where the norm $\|\cdot\|_p$ is defined in semantic perturbation space $\mathbb{R}^s$ with perturbation radius $\epsilon > 0$.

Remark 2.1: Since we are dealing with keypoint detection as a regression task using neural networks under semantic perturbation on the input image, e.g., brightness and contrast, the perturbation space $\mathbb{R}^s$ is usually the 1D real space $s = 1$, and the assumption holds that semantic transformation function is continuous.

Based on the semantic perturbation over the input of the neural network, we then define the robustness certification problem as follows.

Definition 2.4 (Robustness Certification): Given the certification criteria with radius δ under $\|\cdot\|_q$ norm in the output space $\mathbb{R}^{d_L}$ of neural networks f^L in Definition 2.1, the robustness certification problem is to verify whether the following condition holds for semantic perturbation set $B^h_p(x_0, \epsilon)$ in Definition 2.3,

$$\forall x \in B^h_p(x_0, \epsilon), \|f^L(x) - f^L(x_0)\|_q \leq \delta. \quad (2)$$

Remark 2.2: In the case of the keypoint detection task with deep neural networks under semantic perturbation on the input image, the original output is the heat map. However, with the addition of an extra layer of differentiable spatial-to-numerical transformation [22], the entire neural network aligns well with Definition 2.1 for the keypoint regression task. The certification criteria is defined as the maximal pixel deviation of all keypoints away from those of $f^L(x_0)$, i.e., δ in pixels under ℓ_∞ norm for d_L keypoints.

III. METHODOLOGY

Equipped with the definitions of neural networks and pre-activation neuron outputs and the goal of robustness certification under semantic perturbation, in this section, we first analyze the mean and variance of neuron output distribution given the semantic perturbation. Then we introduce the unbiased and smooth neuron metric as an empirical estimate from Monte Carlo sampling of the semantic perturbation set, based on which we introduce a neural network pruning training pipeline to gain more stable and low-variance neurons. Furthermore, a regularization term is proposed based on the Wasserstein distance for concentrated pruning.

A. Neuron Stability and Variance Analysis

Given input perturbation of neural networks, the robustness certification in Definition 2.3 is determined by the nonlinearity and Lipschitz continuity of neural networks [27], [24], [29]. We present the following Lemma to quantify the Lipschitz bound propagation from intermediate layers to neural network output.

Lemma 3.1 (Layer-to-Output Lipschitz Bound): For a neural network f^L as defined in Definition 2.1 and original input x_0, the output deviation of perturbed input $x \in B_p^h(x_0, \epsilon)$ under norm $\|\cdot\|_q, q \geq 2$, can be bounded in terms of any intermediate layer $i = 1, \ldots, L-1$ as,

$$\|f^L(x) - f^L(x_0)\|_q \leq C_i \|f^i(x) - f^i(x_0)\|_2, \tag{3}$$

where the constant $C_i = \|W^i\|_2 \prod_{k=i+1}^{L} \|W^k\|_2 \cdot L_{\sigma^{k-1}}\ L_{\sigma^k}$ and the ℓ_2 Lipschitz constant of activation function σ^k.

Proof: By the induced norm of i-th layer weight W^i and Lipschitz constant of activation function σ^i, we have

$$\|f^L(x) - f^L(x_0)\|_q \leq \|f^L(x) - f^L(x_0)\|_2 \tag{4}$$

$$\leq \|W^L\|_2 \cdot L_{\sigma^{L-1}} \cdot \|f^{L-1}(x) - f^{L-1}(x_0)\|_2 \leq \ldots \tag{5}$$

$$\leq \left(\|W^i\|_2 \prod_{k=i+1}^{L} \|W^k\|_2 L_{\sigma^{k-1}} \right) \|f^i(x) - f^i(x_0)\|_2. \tag{6}$$

Therefore, the bound holds for any layer $i, i \leq L-1$. ∎

Lemma 3.1 bridges the certification goal of Equation (2) with layer-wise deviation given input perturbation, which are highly related to the stability and variance of the intermediate pre-activation neuron outputs in Definition 2.2. Unlike [29], we investigate the stability of neurons based on the statistical distribution of neuron outputs at each layer from semantic perturbation sampling.

Definition 3.1 (Neuron Output Distribution): Given input sample $x_0 \sim p(x_0)$, suppose the perturbed input $p(x \mid x_0)$ is uniformly sampled from $B_p^h(x_0, \epsilon)$, i.e., $x \mid x_0 \sim \mathcal{U}(B_p^h(x_0, \epsilon))$, the j-th pre-activation neuron output on layer i $f_j^i(x)$ has the mean of $\mathbb{E}_{x \sim p(x|x_0)} f_j^i(x)$ and variance of $\text{Var}_{x \sim p(x|x_0)} f_j^i(x)$.

Based on the neuron output distribution under perturbation sampling, we have the following Theorem showing how the mean and variance of the neuron output distribution affect the robustness certification goal in Equation (2).

Theorem 3.1 (Probabilistic Robustness Certification): For any sample $x_0 \sim p(x_0)$ and the uniformly perturbed sample $x \mid x_0 \sim \mathcal{U}(B_p^h(x_0, \epsilon))$, the robustness certification goal $\|f^L(x) - f^L(x_0)\|_q \leq \delta$ in Equation (2) holds with confidence of $1 - \alpha$ if the following inequalities hold for each neuron $j = 1, 2, \ldots, d_i$ on layer $i = 1, 2, \ldots, L-1$

$$\left| \mathbb{E}_{x \sim p(x|x_0)} f_j^i(x) - f_j^i(x_0) \right| \leq \frac{\delta}{2C_i\sqrt{d_i}}, \tag{7}$$

$$\text{Var}_{x \sim p(x|x_0)} f_j^i(x) \leq \frac{\alpha \delta^2}{4C_i^2 d_i^2 (L-i)}, \tag{8}$$

where C_i is Lipschitz bound from Lemma 3.1.

Proof: For the random variable of neuron output $f_j^i(x)$ from Definition 3.1, first apply Chebyshev's inequality and variance bound in Equation (8) as follows,

$$P(|f_j^i(x) - \mathbb{E} f_j^i(x)| \geq \frac{\delta}{2C_i\sqrt{d_i}}) \leq \frac{\text{Var} f_j^i(x)}{(\frac{\delta}{2C_i\sqrt{d_i}})^2} \leq \frac{\alpha}{d_i(L-i)}.$$

Therefore, with a probability of at least $1 - \frac{\alpha}{d_i(L-i)}$, $|f_j^i(x) - \mathbb{E}_{x \sim p(x|x_0)} f_j^i(x)| \leq \frac{\delta}{2C_i\sqrt{d_i}}$ holds. Then by triangle inequality and the bias bound in Equation (7), we have the following hold with probability of at least $1 - \frac{\alpha}{d_i(L-i)}$,

$$|f_j^i(x) - f_j^i(x_0)| \leq |f_j^i(x) - \mathbb{E}_{x \sim p(x|x_0)} f_j^i(x)| + |\mathbb{E}_{x \sim p(x|x_0)} f_j^i(x) - f_j^i(x_0)| \leq \frac{\delta}{C_i\sqrt{d_i}}. \tag{9}$$

By union bound of Equation (9) along each neuron j along layer i, the ℓ_2 norm of layer deviation $f^i(x) - f^i(x_0)$ can be upper bounded below with probability of at least $1 - \frac{\alpha}{(L-i)}$,

$$\|f^i(x) - f^i(x_0)\|_2 = \sqrt{\sum_{j=1}^{d_i} |f_j^i(x) - f_j^i(x_0)|^2} \leq \frac{\delta}{C_i}. \tag{10}$$

Again, based on Lemma 3.1, by union bound of Equation (10) along layers from i to $L-1$, we have $\|f^L(x) - f^L(x_0)\|_q \leq C_i \|f^i(x) - f^i(x_0)\|_2 \leq \delta$ hold with probability of at least $1 - \alpha$, which concludes the proof. ∎

Remark 3.1: We remark that the probabilistic certification will naturally become the deterministic robustness certification in Definition 2.4 by letting $\alpha \to 0$. In practice, we usually want the strongest robustness certification with $\delta = 0$. Therefore, we need to make the upper bounds of Equation (7) and Equation (8) close to 0 during model training.

B. Unbiased and Smooth Neuron (USN) Metrics

During empirical model training, we need to estimate the statistics of the neuron output distribution in Definition 3.1 through finite Monte Carlo sampling. Therefore, in this section, we define the unbiased and smooth neuron metrics to empirically estimate the statistics in Equation (7) and Equation (8), respectively.

Definition 3.2 (Unbiased and Smooth Neuron Metrics): Given the input x_0, the unbiased and smooth metrics for

layer i are defined as:

$$\mathcal{L}^i_{\text{unbiased}} := \frac{1}{m}\sum_{k=1}^{m} \|f^i(x_k) - f^i(x_0)\|_1, \tag{11}$$

$$\mathcal{L}^i_{\text{smooth}} := \frac{1}{m}\sum_{k=1}^{m} \|f^i(x_k) - f^i(x_0)\|_2^2, \tag{12}$$

where m perturbed input are uniformly sampled from $x_k \in B_p^h(x_0, \epsilon), k = 1, 2, \ldots, m$.

Even though Equation (11) and Equation (12) are focusing on ℓ_1 and ℓ_2 norm of the difference between original layer output $f^i(x_0)$ and perturbed layer output $f^i(x_k)$, they have fundamental nuance in the stability and variance of the neuron output in the sense of Equation (7) and Equation (8), respectively. We present the following lemma to formally characterize these relationships.

Lemma 3.2 (USN Metrics and Robustness Bounds): For the unbiased and smooth neuron metrics defined in Equation (11) and Equation (12) with sufficient samples, the following relationships with Equation (7) and Equation (8) hold:

$$\mathcal{L}^i_{\text{unbiased}} = \sum_{j=1}^{d_i} |\mathbb{E}_{x\sim p(x|x_0)} f^i_j(x) - f^i_j(x_0)|, \tag{13}$$

$$\mathcal{L}^i_{\text{smooth}} = \sum_{j=1}^{d_i} [\text{Var}_x f^i_j(x) + |\mathbb{E}_x f^i_j(x) - f^i_j(x_0)|^2]. \tag{14}$$

Proof: For the first relation in Equation (13), by definition of the ℓ_1 norm and law of large number: $\mathcal{L}^i_{\text{unbiased}} = \frac{1}{m}\sum_{k=1}^{m}\sum_{j=1}^{d_i} |f^i_j(x_k) - f^i_j(x_0)| = \sum_{j=1}^{d_i} \mathbb{E}_{x\sim p(x|x_0)} |f^i_j(x) - f^i_j(x_0)|$. Since x_0 is deterministic given the conditioning, we have $|f^i_j(x) - f^i_j(x_0)| = |\mathbb{E}_{x\sim p(x|x_0)} f^i_j(x) - f^i_j(x_0)|$ in expectation, establishing Equation (13). Similarly, for the second relation in Equation (14), based on bias-variance decomposition, we have

$$\begin{aligned}\mathcal{L}^i_{\text{smooth}} &= \sum_{j=1}^{d_i} \mathbb{E}_x [f^i_j(x) - \mathbb{E}(f^i_j(x)) + \mathbb{E}(f^i_j(x)) - f^i_j(x_0)]^2 \\ &= \sum_{j=1}^{d_i} [\text{Var}_x f^i_j(x) + (\mathbb{E}_x f^i_j(x) - f^i_j(x_0))^2],\end{aligned}$$

which concludes the proof. ∎

Combining Lemma 3.2 and Theorem 3.1, we present the following robustness certification theorem with the USN metric conditions.

Corollary 3.1 (USN Necessary Bound Conditions): If Equation (7) and Equation (8) hold for any neuron i on layer j in Theorem 3.1, we have the following upper bounds for $\mathcal{L}^i_{\text{unbiased}}$ and $\mathcal{L}^i_{\text{smooth}}$,

$$\mathcal{L}^i_{\text{unbiased}} \le \frac{\delta\sqrt{d_i}}{2C_i},\ \mathcal{L}^i_{\text{smooth}} \le \frac{\delta^2}{4C_i^2}\left(\frac{\alpha}{d_i(L-i)} + 1\right). \tag{15}$$

Proof: The upper bounds can be obtained by applying Equation (7) and Equation (8) to Equation (13) and Equation (14). ∎

Remark 3.2: Even though Equation (15) is not a sufficient condition for robustness certification for general $\delta > 0, \alpha > 0$, but it is aligned with the robustness certification goal in Equation (2) when $\delta \to 0, \alpha \to 0$, showing that minimizing $\mathcal{L}^i_{\text{unbiased}}$ and $\mathcal{L}^i_{\text{smooth}}$ will lead to certification goal. We remark that the signal-to-noise ratio (SNR) losses in [29] are special cases of unbiased and smooth neuron metrics, where they are normalized by $\|f^i(x_0)\|$. When $\|f^i(x_0)\|$ significantly increases during model training, even if SNR losses can be greatly reduced, the upper bounds of Equations (7) and (8) do not necessarily hold, and therefore the certification goal would fail when $\delta \to 0, \alpha \to 0$. Besides, ours can handle more general semantic perturbations while SNR losses mainly focus on ℓ_∞-bounded perturbations.

C. Wasserstein Distance for USN Regularization

Since smaller USN metrics of $\mathcal{L}^i_{\text{smooth}}$ and $\mathcal{L}^i_{\text{smooth}}$ in Definition 3.2 can inherently ensure the robustness of the neural network, we adopt structured pruning [32], [2] of the neurons with larger USN metrics. To ensure concentrated and coherent pruning patterns across network layers, we introduce a Wasserstein distance regularization that promotes structured sparsity. The Wasserstein distance [23], also known as the Earth Mover's Distance, measures the minimum cost to transform one probability distribution into another.

Definition 3.3 (Wasserstein Distance): For two discrete probability distributions $\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$ and $\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$ with $\sum_i a_i = \sum_j b_j = 1$, the 2-Wasserstein distance is defined as:

$$\mathcal{W}_2(\mu, \nu) = \min_{\pi\in\Pi(\mu,\nu)} \left(\sum_{i,j} \pi_{ij} \|x_i - y_j\|_2^2\right)^{1/2}, \tag{16}$$

where $\Pi(\mu, \nu)$ represents the set of all joint distributions with marginals μ and ν, and π_{ij} denotes the transport plan indicating how much probability mass moves from x_i to y_j.

In the context of neural network pruning, we apply the Wasserstein distance to align the importance distributions of neurons across layers. Drawing from the USN relations in Lemma 3.2, we define the importance score for each neuron based on its contribution to both the unbiased and smooth metrics. For each layer i, the neuron-wise contribution to the USN metrics can be decomposed as:

$$\mathcal{L}^i_{\text{unbiased},j} = |\mathbb{E}_{x\sim p(x|x_0)} f^i_j(x) - f^i_j(x_0)|, \tag{17}$$

$$\mathcal{L}^i_{\text{smooth},j} = \text{Var}_x f^i_j(x) + |\mathbb{E}_x f^i_j(x) - f^i_j(x_0)|^2. \tag{18}$$

Therefore, the importance score $\mathcal{A}^i_j$ for neuron j in layer i is then defined as the following dimensionless ratio between the smooth metric (with dimension of squared ℓ_2 norm in Equation (12)) and the square of the unbiased metric (with dimension of ℓ_1 norm in Equation (11)),

$$\mathcal{A}^i_j = \frac{\mathcal{L}^i_{\text{smooth},j}}{(\mathcal{L}^i_{\text{unbiased},j}{}^2 + \epsilon_{usn}) \cdot d_i}, \tag{19}$$

where where $\mathcal{L}^i_{\text{smooth},j}$ and $\mathcal{L}^i_{\text{unbiased},j}$ are the neuron-wise contributions in Equation (17) and Equation (18), ϵ_{usn} is a small regularization constant to prevent division by zero, and

d_i provides layer-wise normalization to ensure dimensional consistency and comparability across layers of different widths. Note that in practice, we conduct structured pruning by calculating the importance score $\mathcal{A}$ for each coarser-grained channel of neurons [5], [32], [3], which identifies the channels with the most unstable neurons for pruning.

To encourage concentrated pruning, we define the target distribution $\mathcal{E}^i$ using percentile-based thresholding:

$$\mathcal{E}_j^i = \begin{cases} \frac{1}{d_i}, & \text{if } \mathcal{A}_j^i > \text{percentile}(\mathcal{A}^i, (1-\rho)\times 100) \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

where ρ is the target pruning ratio. Based on Definition 3.3, the Wasserstein regularization loss for layer i is then:

$$\mathcal{L}_{\text{W}}^i = \mathcal{W}_2(\mathcal{M}^i, \mathcal{E}^i). \tag{21}$$

This regularization encourages the network to develop clear distinctions between important and unimportant neurons based on their USN characteristics, facilitating more effective structured pruning by promoting concentrated removal of the most unstable neurons.

D. Integrated Training and Pruning Pipeline

We finally propose a progressive training pipeline that integrates robust training with dynamic pruning based on USN metrics. The training process consists of multiple phases where the pruning ratio gradually increases, allowing the network to adapt to the reduced capacity.

The total loss function combines multiple components:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \sum_{i\in\mathcal{I}_{\text{prune}}} \left(\lambda_u \mathcal{L}_{\text{unbiased}}^i + \lambda_s \mathcal{L}_{\text{smooth}}^i + \lambda_W \mathcal{L}_{\text{W}}^i\right),$$

where task loss $\mathcal{L}_{\text{task}}$ comes from the task-specific domain (e.g. keypoint detection literature [20]) and $\mathcal{I}_{\text{prune}}$ is the set of layers to be pruned. The progressive pruning schedule is defined as:

$$\rho(t) = \begin{cases} 0, & \text{if } t < t_{\text{start}} \\ \frac{\rho}{N_{\text{steps}}} \left\lfloor \frac{t-t_{\text{start}}}{t_{\text{interval}}} \right\rfloor, & \text{if } t_{\text{start}} \le t \le t_{\text{end}} \\ \rho, & \text{if } t > t_{\text{end}} \end{cases} \tag{22}$$

where t is the current epoch, ρ is the final target pruning ratio, N_{steps} is the number of pruning steps, and t_{interval} is the interval between pruning operations.

This progressive approach allows the network to gradually adapt to reduced capacity while maintaining performance. The USN metrics guide the pruning process by identifying neurons that exhibit high variance or bias under semantic perturbations, ensuring that the most stable and reliable neurons are preserved. The Wasserstein regularization promotes coherent pruning patterns that maintain the network's structural integrity and representational power.

Algorithm 1 Progressive USN-Guided Training and Pruning

Require: Neural network f_θ^L, training data $\mathcal{D}$, semantic perturbation set $B_p^h(\cdot,\epsilon)$, final pruning ratio ρ, pruning steps N_{steps}, learning rate α, pruning layer set $\mathcal{I}_{\text{prune}}$
Ensure: Trained and progressively pruned network

Initialize: $\rho_{\text{current}} \leftarrow 0$, importance $\{\mathcal{A}^i \leftarrow \mathbf{0}\}_{i=1}^{L-1}$
for epoch $t = 1$ to T **do**
  Update pruning ratio: $\rho_{\text{current}} \leftarrow \rho(t)$ according to schedule in Equation (22)
  for each batch $\{x_b, y_b\}$ in $\mathcal{D}$ **do**
    Sample semantic perturbations $x \leftarrow B_p^h(x_b, \epsilon)$
    Compute task loss: $\mathcal{L}_{\text{task}}$
    for each layer $i \in \mathcal{I}_{\text{prune}}$ **do**
      Compute USN metrics $\mathcal{L}_{\text{unbiased}}^i, \mathcal{L}_{\text{smooth}}^i$ based on Equation (11) and Equation (12)
      Compute neuron importance $\mathcal{A}_j^i$ based on Equation (19)
      Compute Wasserstein loss $\mathcal{L}_{\text{W}}^i$ based on Equation (21)
    end for
    Total loss: $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{task}} + \sum_{i\in\mathcal{I}_{\text{prune}}}(\lambda_u \mathcal{L}_{\text{unbiased}}^i + \lambda_s \mathcal{L}_{\text{smooth}}^i + \lambda_W \mathcal{L}_{\text{W}}^i)$
    Backpropagate and update parameters: $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\text{total}}$
  end for
  if t is pruning epoch **then**
    for each layer $i \in \mathcal{I}_{\text{prune}}$ **do**
      Compute pruning threshold: $\tau^i \leftarrow \text{percentile}(\mathcal{A}^i, (1-\rho_{\text{current}})\times 100)$
      Generate pruning mask: $\mathcal{P}_j^i \leftarrow \mathbb{1}[\mathcal{A}_j^i \ge \tau^i]$
      Apply structured pruning by keeping masked neurons: $\theta \leftarrow \theta(\mathbb{1}[\mathcal{P}_j^i = 1])$
    end for
    Reset importance: $\mathcal{A}^i \leftarrow \mathbf{0}_{i=1}^{L-1}$
  end if
end for
return Progressively pruned network θ

IV. EXPERIMENTS

In this section, we evaluate our USN-guided pruning approach by answering two key research questions: 1) *Does USN-guided pruning improve robustness certification performance compared to random pruning and unpruned baselines under realistic semantic perturbations?* 2) *What are the effects of pruning ratio and Wasserstein regularization on the trade-off between certification accuracy, model expressiveness, and verification efficiency?* The answer to the first question will be found in Section IV-B through comparisons on CNN7 and ResNet18 architectures, while the second question is addressed in Section IV-C through systematic ablation studies. Prior to those, we first introduce the experimental setup with datasets, training procedures, and evaluation metrics in Section IV-A.

A. Experimental Setup

a) Training and Pruning Details: Following the literature of keypoint detection [20], we use the same aircraft dataset with 24 keypoints per image and split them into train/val/test with fixed seeds. We adopt backbones of different architectures: a seven-layer CNN (CNN7) from [1], [31]

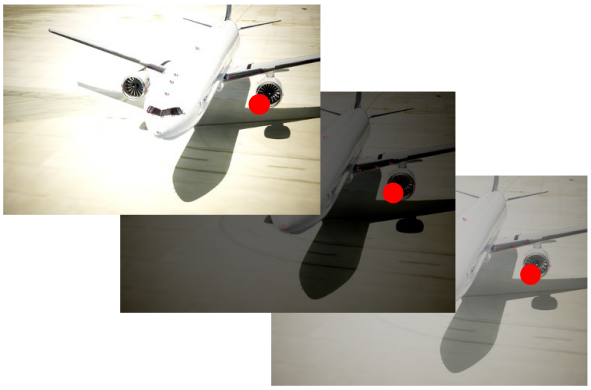


Fig. 2: Samples of keypoint detection under changing semantic perturbation of images.

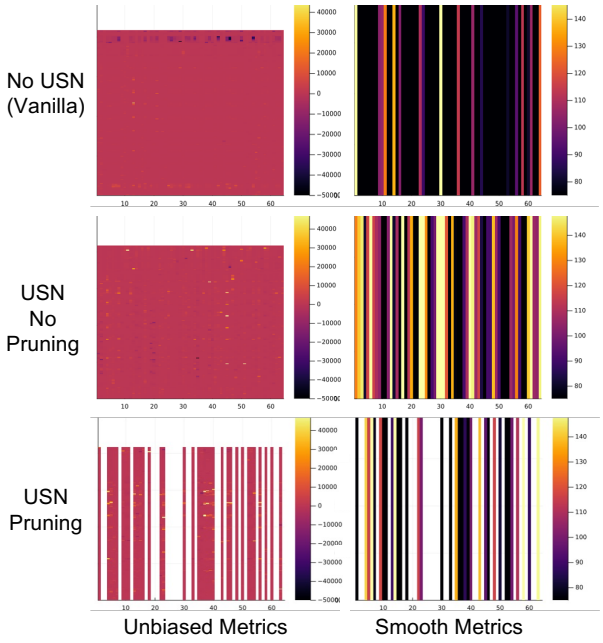


Fig. 3: Visualization of USN metrics (color-coded) w.r.t. flattened channels of neurons (each column represents one channel and the horizontal axes show different channels) after vanilla training without USN, robust training with USN but without pruning, and robust training and pruning with USN (ours). We can see that our robust pruning can significantly reduce the neurons with high unbiased and smooth metrics compared to robust training with USN without pruning.

and ResNet18 [7]. Models are trained for 200 epochs with Adam (learning rate $\alpha = 0.01$), batch size of 64, and task losses from [20]. We inject small photometric perturbations (brightness $\pm\frac{1}{255}$, contrast ± 0.01), apply USN regularization on the convolution layers, and perform progressive channel-level pruning to $\rho = 0.1, 0.2$ over $N_{\text{steps}} = 200$ steps in Algorithm 1. Note that channel-level pruning uses the same technique as neuron-wise pruning but with different granularity, and is much more efficient in the literature of structured pruning [5], [32], [3]. We also conduct an ablation of the Wasserstein regularization with $\lambda_W \in \{0, 10\}$. During the model training, we monitor the keypoint task loss on clean data and uniformly perturbed ones and the best checkpoint is chosen by the lowest task loss on the validation set. Experiments run on a workstation with four NVIDIA RTX A6000 GPUs.

b) Certification Metrics and Baselines: We certify all the models over brightness shifts $\{\pm 2, \pm 5\}$ pixels ($\epsilon \in \{2/255, 5/255\}$ for brightness transformation h in Definition 2.3) and contrast scalings $\{\pm 0.01, \pm 0.02, \pm 0.05\}$ ($\epsilon \in \{0.01, 0.02, 0.05\}$ for contrast transformation h in Definition 2.3). For each test image, we crop and resize to 64×64 with the output of the 24 keypoints of this frame, and check whether all predicted keypoints stay within the pixel error bound of 1 px. That being said, the robustness certification problem is with $d_L = 24, \delta = 1, q = \infty$ in Definition 2.4. We adopt the solver `ModelVerification.jl` [28] to return *Holds*, *Violated*, or *Unknown* for the certification goal in Equation (2). We adopt the metric of verification accuracy, which is computed as the proportion of test images with certification goal achieved. We also compare the verification time for efficiency and the number of correctly predicted and verified keypoints for fidelity as fine-grained metrics. The baselines are the ones without pruning and with random pruning under multiple pruning rates.

B. Results Comparison

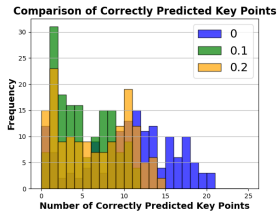
a) CNN7: Table I shows comparison of *USN-guided pruning* with *random pruning* at matched rates under brightness (± 2, ± 5) and contrast (± 0.01, ± 0.02, ± 0.05) perturbations. USN guidance consistently matches or exceeds random pruning across all magnitudes, with the largest gains appearing at the highest contrast; under stronger brightness shifts, USN also improves over the random pruning baseline. Overall, USN pruning at a rate $\rho = 0.2$ provides the most consistent accuracy across perturbations, indicating that channels selected by USN is more robust than random pruning due to the removal of unstable and high-variance neurons.

TABLE I: CNN7: USN-guided vs. random pruning under different pruning ratio ρ against brightness/contrast perturbations

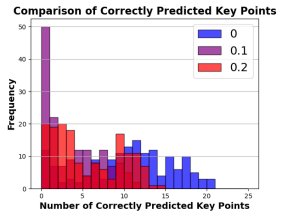
ρ	Pruning Rule	Brightness		Contrast		
		± 2	± 5	± 0.01	± 0.02	± 0.05
0.1	Random	0.9545	0.9025	**0.9805**	0.9350	0.8051
	USN-guided	**0.9740**	**0.9285**	**0.9805**	**0.9545**	**0.8961**
0.2	Random	0.9805	0.9155	0.9805	0.9610	0.9285
	USN-guided	**0.9870**	**0.9480**	**0.9935**	**0.9740**	**0.9675**
0.3	Random	0.9545	0.9220	**0.9675**	0.9545	0.9350
	USN-guided	**0.9805**	**0.9350**	**0.9675**	**0.9610**	**0.9415**

b) ResNet18: We compare pruning strategies in terms of fidelity (correct keypoints) and efficiency (verification time) in Figure 4. The model without pruning ($\rho = 0$) results in more correct keypoints, but the verification time

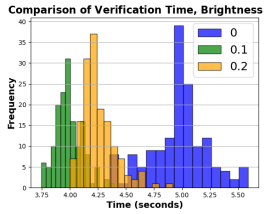
is significantly larger than that of the models with pruning. For pruned models under the same Wasserstein regularization $\lambda_W = 10$, compared to *random pruning* baselines, the *USN-guided pruning* ones produce keypoint histograms that shift to the right in Figure 4 (a,b), indicating more correctly predicted keypoints; in Figure 4 (c,d), both achieve a comparable left-shift in runtime, indicating similar verification speedups. In short, USN-guided pruning attains better efficiency with substantially higher fidelity than the random pruning baseline. Among USN settings, the intermediate configuration offers the best efficiency-fidelity trade-off; within random pruning, the higher rate is less damaging than the lower rate yet remains inferior to USN-guided pruning.



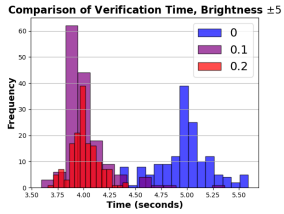
(a) Correctly predicted keypoints (USN-guided pruning rates $\rho \in \{0, 0.1, 0.2\}$)



(b) Correctly predicted keypoints (Random pruning rates $\rho \in \{0, 0.1, 0.2\}$)



(c) Average verification time (USN-guided pruning $\rho \in \{0, 0.1, 0.2\}$)



(d) Average verification time (Random pruning $\rho \in \{0, 0.1, 0.2\}$)

Fig. 4: Comparison of the number of correctly predicted keypoints and verification time under different pruning strategies for ResNet18 under Wasserstein regularization $\lambda_W = 10$.

c) Visualization of model weight after training: In Figure 3, we visualize the fourth layer of the CNN7 model after three different training paradigms: vanilla training with only task losses without USN metrics, robust training with USN metrics but without pruning, and the proposed robust pruning with USN metrics. It can be seen that the unbiased and smooth metrics for neurons after robust training with USN metrics but without pruning (`USN No Pruning`) are the highest due to over-parameterization by robust training [29]. However, ours (`USN Pruning`) can significantly reduce the number of neurons with higher values of unbiased and smooth metrics and retain most of the stable neurons, where our unbiased and smooth metrics are comparable to those obtained after vanilla training. In summary, ours is easier to verify compared to the robust training of `USN No Pruning` and more robust than the model of vanilla training without USN.

C. Ablation Study

a) Effect of pruning rate in Table II: Sweeping $\rho \in \{0.1, 0.2, 0.3\}$ indicates that $\rho = 0.2$ offers the most balanced trade-off between model verifiability and expressiveness under input perturbations compared to the non-pruning baseline $\rho = 0.0$. Due to the progressive pruning, the higher the pruning ratio is, the less expressive the pruned model will be. In contrast, a stronger pruning effect will make the pruned model easier to verify. At $\rho = 0.1$, insufficient pruning fails to eliminate enough unstable neurons that contribute to over-approximation during verification, resulting in suboptimal certified robustness despite maintaining high model capacity (which is still lower than $\rho = 0.0$ through). However, aggressive pruning $\rho = 0.3$ removes too many essential neurons needed for robust feature representation, explaining the degradation at higher pruning ratios.

TABLE II: CNN7 verification accuracy under different pruning ratio under $\lambda_W = 10$.

Pruning Ratio	Brightness		Contrast		
	±2	±5	±0.01	±0.02	±0.05
$\rho = 0.0$	0.9805	**0.9480**	0.9870	**0.9870**	0.9545
$\rho = 0.1$	0.9740	0.9285	0.9805	0.9545	0.8961
$\rho = 0.2$	**0.9870**	**0.9480**	**0.9935**	0.9740	**0.9675**
$\rho = 0.3$	0.9805	0.9350	0.9675	0.9610	0.9415

b) Effect of Wasserstein Regularization in Table III: Enabling the Wasserstein term ($\lambda_W = 10$) generally improves certification at $\rho \in \{0.2, 0.3\}$, with observable gains across all tested perturbations at the moderate rate and the largest improvements at higher contrast and stronger brightness at the stronger rate. At $\rho = 0.1$, however, the Wasserstein regularizer can over-concentrate pruning and degrade robustness under strong perturbations, indicating that the regularization term is most beneficial under mild perturbations with satisfactory accuracy.

TABLE III: CNN7 verification accuracy with or without Wasserstein regularization ($\lambda_W \in \{10, 0\}$) under different pruning ratio ρ.

ρ	Wasserstein weight	Brightness		Contrast		
		±2	±5	±0.01	±0.02	±0.05
0.1	$\lambda_W = 0$	0.9610	**0.9480**	0.9740	**0.9610**	**0.9415**
	$\lambda_W = 10$	**0.9740**	0.9285	**0.9805**	0.9545	0.8961
0.2	$\lambda_W = 0$	0.9675	0.9350	0.9610	0.9675	0.9350
	$\lambda_W = 10$	**0.9870**	**0.9480**	**0.9935**	**0.9740**	**0.9675**
0.3	$\lambda_W = 0$	0.9545	0.9090	**0.9740**	0.9415	0.8311
	$\lambda_W = 10$	**0.9805**	**0.9350**	0.9675	**0.9610**	**0.9415**

V. CONCLUSION

This paper addresses over-parameterization challenges in robustness certification by introducing Unbiased and Smooth Neuron (USN) metrics that identify neurons contributing to certification looseness under semantic perturbations. Our progressive training pipeline integrates USN-guided pruning with Wasserstein distance regularization to achieve structured

sparsity while preserving model expressiveness. Experiments on keypoint detection under brightness and contrast variations demonstrate that strategic removal of high-variance neurons consistently improves both certification accuracy and verification efficiency compared to unpruned and randomly pruned baselines. This work establishes a principled foundation for robustness-aware model compression, enabling more tractable formal guarantees for safety-critical applications while maintaining task performance.

ACKNOWLEDGMENT

This material is based upon work supported by The Boeing Company. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of The Boeing Company.

REFERENCES

[1] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[2] T. Chen, H. Zhang, Z. Zhang, S. Chang, S. Liu, P.-Y. Chen, and Z. Wang. Linearity grafting: Relaxed neuron pruning helps certifiable robustness. In *International conference on machine learning*, pages 3760–3772. PMLR, 2022.

[3] H. Cheng, M. Zhang, and J. Q. Shi. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[4] J. Cohen, E. Rosenfeld, and Z. Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR, 2019.

[5] J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2018.

[6] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations*, 2016.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] H. Hu, C. Liu, and D. Zhao. Robustness verification for perception models against camera motion perturbations. In *International conference on machine learning (ICML) Workshop on Formal Verification of Machine Learning (WFVML)*, 2023.

[9] H. Hu, Z. Liu, S. Chitlangia, A. Agnihotri, and D. Zhao. Investigating the impact of multi-lidar placement on object detection for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2550–2559, 2022.

[10] H. Hu, Z. Liu, L. Li, J. Zhu, and D. Zhao. Robustness certification of visual perception models via camera motion smoothing. In *Conference on Robot Learning*, pages 1309–1320. PMLR, 2022.

[11] H. Hu, Z. Liu, L. Li, J. Zhu, and D. Zhao. Pixel-wise smoothing for certified robustness against camera motion perturbations. In *International Conference on Artificial Intelligence and Statistics*, pages 217–225. PMLR, 2024.

[12] H. Hu, H. Wang, Z. Liu, C. Yang, W. Chen, and L. Xie. Retrieval-based localization based on domain-invariant feature learning under changing environments. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 3684–3689. IEEE, 2019.

[13] H. Hu, B. Yang, Z. Qiao, S. Liu, J. Zhu, Z. Liu, W. Ding, D. Zhao, and H. Wang. Seasondepth: Cross-season monocular depth prediction dataset and benchmark under multiple environments. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11384–11389. IEEE, 2023.

[14] A. Jordao and H. Pedrini. On the effect of pruning on adversarial robustness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–11, 2021.

[15] L. Kong, S. Xie, H. Hu, L. X. Ng, B. Cottereau, and W. T. Ooi. Robodepth: Robust out-of-distribution depth estimation under corruptions. *Advances in Neural Information Processing Systems*, 36:21298–21342, 2023.

[16] L. Kong, S. Xie, H. Hu, Y. Niu, W. T. Ooi, B. R. Cottereau, L. X. Ng, Y. Ma, W. Zhang, L. Pan, et al. The robodrive challenge: Drive anytime anywhere in any condition. *arXiv preprint arXiv:2405.08816*, 2024.

[17] P. Kouvaros, F. Leofante, B. Edwards, C. Chung, D. Margineantu, and A. Lomuscio. Verification of semantic key point detection for aircraft pose estimation. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 19, pages 757–762, 2023.

[18] C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett, M. J. Kochenderfer, et al. Algorithms for verifying deep neural networks. *Foundations and Trends® in Optimization*, 4(3-4):244–404, 2021.

[19] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell. Rethinking the value of network pruning. In *International Conference on Learning Representations*, 2018.

[20] X. Luo, T. Wei, S. Liu, Z. Wang, L. Mattei-Mendez, T. Loper, J. Neighbor, C. Hutchison, and C. Liu. Certifying robustness of learning-based keypoint detection and pose estimation methods. *ACM Transactions on Cyber-Physical Systems*, 9(2):1–26, 2025.

[21] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[22] A. Nibali, Z. He, S. Morgan, and L. Prendergast. Numerical coordinate regression with convolutional neural networks. *arXiv preprint arXiv:1801.07372*, 2018.

[23] V. M. Panaretos and Y. Zemel. Statistical aspects of wasserstein distances. *Annual review of statistics and its application*, 6(1):405–431, 2019.

[24] P. Pauli, A. Koch, J. Berberich, P. Kohler, and F. Allgöwer. Training robust neural networks using lipschitz bounds. *IEEE Control Systems Letters*, 6:121–126, 2021.

[25] V. Sehwag, S. Wang, P. Mittal, and S. Jana. Hydra: Pruning adversarially robust neural networks. *Advances in Neural Information Processing Systems*, 33:19655–19666, 2020.

[26] R. Talak, L. R. Peng, and L. Carlone. Certifiable object pose estimation: Foundations, learning models, and self-training. *IEEE Transactions on Robotics*, 39(4):2805–2824, 2023.

[27] A. Virmaux and K. Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

[28] T. Wei, H. Hu, L. Marzari, K. S. Yun, P. Niu, X. Luo, and C. Liu. Modelverification. jl: a comprehensive toolbox for formally verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 395–408. Springer, 2025.

[29] T. Wei, Z. Wang, P. Niu, A. Abuduweili, W. Zhao, C. Hutchison, E. Sample, and C. Liu. Improve certified training with signal-to-noise ratio loss to decrease neuron variance and increase neuron stability. *Transactions on Machine Learning Research*, 2024.

[30] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel. Efficient neural network robustness certification with general activation functions. *Advances in neural information processing systems*, 31, 2018.

[31] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019.

[32] L. Zhangheng, T. Chen, L. Li, B. Li, and Z. Wang. Can pruning improve certified robustness of neural networks? *Transactions on Machine Learning Research*, 2022.