Comparative Analysis of Interval Reachability for Robust Implicit and Feedforward Neural Networks

Alexander Davydov[1,*], Saber Jafarpour[2,*], Matthew Abate[2], Francesco Bullo[1], Samuel Coogan[2]

***Abstract*— Implicit neural networks (INNs) are a class of learning models that use implicit algebraic equations as layers and have been shown to exhibit several notable benefits over traditional feedforward neural networks (FFNNs). In this paper, we use interval reachability analysis to study robustness of INNs and compare them with FFNNs. We first introduce the notion of tight inclusion function and use it to provide the tightest rectangular over-approximation of the neural network's input-output map. We also show that tight inclusion functions lead to sharper robustness guarantees than the well-studied robustness measures of Lipschitz constants. Like exact Lipschitz constants, tight inclusions functions are computationally challenging to obtain, and thus we develop a framework based upon mixed monotonicity and contraction theory to estimate the tight inclusion functions for INNs. We show that our approach performs at least as well as, and generally better than, state-of-the-art interval-bound propagation methods for INNs. Finally, we design a novel optimization problem for training robust INNs and we provide empirical evidence that suitably-trained INNs can be more robust than comparably-trained FFNNs.**

I. Introduction

Implicit neural networks (INNs) are a class of implicit learning models where the hidden layers are replaced with implicit equations [1], [2], [3]. Compared to their explicit counterparts, INNs are known to have advantages including (i) being more suitable for some problems such as constrained optimization problems [1] (ii) being more memory efficient while maintaining comparable accuracy [2] (iii) allowing for new architecture possibilities [3] (vi) showing improved training due to fewer vanishing and exploding gradients [4]. Despite their benefits, INNs can suffer from well-posedness issues and convergence instabilities. Additionally, their input-output behavior may suffer from robustness issues and adversarial perturbations; indeed, such robustness vulnerabilities are a well-studied and major issue in traditional deep neural networks as well [5].

Problem statement and motivations: Obtaining provable robustness guarantees for learning algorithms has been a major goal in the machine learning literature [6], [7]. For feedforward neural networks (FFNNs), four well-established methods for producing robustness guarantees include (i) Lipschitz bound approaches, (ii) interval-bound propagation (IBP) methods, (iii) convex-relaxation approaches, and (iv) approaches based on Satisfiability Modulo Theories (SMT). Lipschitz constants of neural networks are coarse but rigorous measures for their input-output robustness. While it has been shown that computing the exact Lipschitz constant of a neural network is NP-hard [8], several efficient methods for providing sharp estimates of Lipschitz bounds are proposed in the literature [9]. IBP methods use interval analysis to provide box over-approximations of the reachable set of neural networks. These methods have been successfully used to perform formal verification [10] and to train robust FFNNs [11], [12]. Convex-relaxation approaches are based on relaxations of nonlinear activation functions using either linear [13] or quadratic constraints [14]. Finally, the SMT-based methods generalize the existing formal verification techniques for robustness analysis of neural networks [15].

For robustness guarantees of INNs, several works provide estimates of their Lipschitz constants [16], [17], [18]. However, global Lipschitz bounds do not provide information about the local sensitivity of the networks. The paper [13] proposes an iterative IBP approach for reachability analysis and training of INNs. However, convergence of this iteration requires strong conditions which limit the expressivity of the resulting implicit models. In [19], a method based on semidefinite programming is proposed for robustness analysis of INNs. Unfortunately, this approach is computationally intensive and cannot be implemented in training.

Most works on INNs focus on comparing their representation power [3] or their memory efficiency [2] with traditional deep neural networks. While recent empirical evidence indicate that appropriately-trained INNs can be significantly more robust than deep feedforward models [16], there are very few works in the literature that rigorously compare robustness of INNs with FFNNs. In this paper, we aim to address two main challenges regarding robustness of neural networks: (i) designing algorithms to train provably robust INNs, and (ii) appropriately comparing the robustness of INNs with the robustness of their explicit counterparts.

Contributions: In this paper, we use interval reachability analysis to study robustness of INNs. We first introduce the notion of tight inclusion function associated to the INN that gives the tightest rectangular over-approximation for the input-output behavior of the INN. We show that tight inclusion functions are sharper than any robustness guarantees based on local Lipschitz bounds. Similar to Lipschitz constants, computing the tight inclusion function is computationally challenging. Instead, using mixed monotone systems theory and contraction theory, we provide computa-

* These authors contributed equally and the order is alphabetical.

[1]Alexander Davydov and Francesco Bullo are with the Center for Control, Dynamical Systems, and Computation, University of California, Santa Barbara, {davydov, bullo}@ucsb.edu

[2]Saber Jafarpour, Matthew Abate, and Samuel Coogan are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, {saber, matt.abate, sam.coogan}@gatech.edu

This work is partly funded by the Air Force Office of Scientific Research under grant FA9550-22-1-0059 and National Science Foundation under awards #1749357 and #1931980.

tionally efficient estimates of the tight inclusion functions of INNs. We use two different interpretations of implicit neural networks to compare our approach with the IBP approach for FFNNs. We show that our mixed monotone contracting approach is the natural extension of IBP methods to INNs and performs at least as well as, and generally better than, IBP methods naively applied to INNs. Lastly, we provide an algorithm to efficiently implement our mixed monotone contracting approach in the training optimization problem to design robust INNs. In numerical experiments, we compare the performance of INNs and FFNNs with a comparable number of parameters and demonstrate that suitably-trained INNs have improved certified and empirical robustness compared to their feedforward counterparts, even when trained with IBP. In the conference paper [20], we focus on verifying robustness of INNs. In contrast, in this paper, we provide an efficient means for training robust INNs, compare theoretical robustness guarantees for INNs to FFNNs, extend [20, Theorem 1] and provide the proof for it, and present an empirical study of training for robustness.

II. Mathematical Preliminary

For $x, y, z \in \mathbb{R}^n$, we write $x \leq y$ if $x_i \leq y_i$ for all $i \in \{1, \dots, n\}$ and $z \in [x, y]$ if $x \leq z \leq y$. For $\eta \in \mathbb{R}^n_{>0}$, we define the diagonal matrix $[\eta] \in \mathbb{R}^{n \times n}$ by $[\eta]_{ii} = \eta_i$, for every $i \in \{1, \dots, n\}$ and the diagonally weighted ℓ_∞-norm by $\|x\|_{\infty,[\eta]^{-1}} = \max_i |x_i|/\eta_i$, the diagonally weighted ℓ_∞-matrix measure is defined by $\mu_{\infty,[\eta]^{-1}}(A) = \max_{i \in \{1,\dots,n\}} A_{ii} + \sum_{j \neq i} \frac{\eta_j}{\eta_i} |A_{ij}|$. For any matrix $A \in \mathbb{R}^{n \times n}$, the spectral radius of A is denoted by $\rho(A)$ and the elementwise absolute value of A is denoted by $|A| \in \mathbb{R}^{n \times n}_{\geq 0}$. For two matrices A, B, let $A \otimes B$ denote their Kronecker product. Given a matrix $B \in \mathbb{R}^{n \times m}$, we denote the non-negative part of B by $[B]^+ := \max(B, 0)$ and the nonpositive part of B by $[B]^- := \min(B, 0)$. The Metzler and non-Metzler parts of a square matrix $A \in \mathbb{R}^{n \times n}$ are denoted by $\lceil A \rceil^{\mathrm{Mzl}}$ and $\lfloor A \rfloor^{\mathrm{Mzl}}$, respectively, where

$$(\lceil A \rceil^{\mathrm{Mzl}})_{ij} := \begin{cases} A_{ij} & \text{if } A_{ij} \geq 0 \text{ or } i = j \\ 0 & \text{otherwise,} \end{cases}$$

and $\lfloor A \rfloor^{\mathrm{Mzl}} := A - \lceil A \rceil^{\mathrm{Mzl}}$. The subset $\mathcal{T}^n \subset \mathbb{R}^{2n}$ is defined by $\mathcal{T}^n := \{(x, \widehat{x}) \in \mathbb{R}^{2n} \mid x \leq \widehat{x}\}$. Given a map $f : \mathbb{R}^n \to \mathbb{R}^m$, a set $\mathcal{U} \subset \mathbb{R}^n$, and $p \in [1, \infty]$, the ℓ_p-Lipschitz constant of f on $\mathcal{U}$ is the smallest $\mathrm{Lip}_p^{\mathcal{U}}(f) \in \mathbb{R}_{\geq 0}$ such that $\|f(x) - f(y)\|_p \leq \mathrm{Lip}_p^{\mathcal{U}}(f)\|x - y\|_p$, for all $x, y \in \mathcal{U}$. Given a map $f : \mathbb{R}^n \to \mathbb{R}^n$ and $\alpha \in [0, 1]$, the α-average map is defined by $f_\alpha(x) = (1 - \alpha)x + \alpha f(x)$, for every $x \in \mathbb{R}^n$.

III. Inclusion Functions

Given a mapping f, an ℓ_∞-box over-approximation of the image of f is attainable via an inclusion function.

Definition 3.1 (Inclusion function): Let $f : \mathbb{R}^r \to \mathbb{R}^q$ be a mapping. Then $\mathsf{F} = \begin{bmatrix} \underline{\mathsf{F}} \\ \overline{\mathsf{F}} \end{bmatrix} : \mathcal{T}^r \to \mathbb{R}^{2q}$ is an inclusion function for f, if, for every $x \leq y \leq \widehat{x}$,

(i) $\underline{\mathsf{F}}(y, y) \geq \underline{\mathsf{F}}(x, \widehat{x})$ and $\overline{\mathsf{F}}(y, y) \leq \overline{\mathsf{F}}(x, \widehat{x})$;
(ii) $\underline{\mathsf{F}}(x, x) = \overline{\mathsf{F}}(x, x) = f(x)$.

Moreover, the inclusion function F for f is called *tight*, if

(iii) for every inclusion function $\mathsf{G} = \begin{bmatrix} \underline{\mathsf{G}} \\ \overline{\mathsf{G}} \end{bmatrix} : \mathcal{T}^r \to \mathbb{R}^{2q}$ of f, we have

$$\underline{\mathsf{G}}(x, \widehat{x}) \leq \underline{\mathsf{F}}(x, \widehat{x}), \quad \overline{\mathsf{F}}(x, \widehat{x}) \geq \overline{\mathsf{G}}(x, \widehat{x}), \quad \text{for all } x \leq \widehat{x}$$

If F is an inclusion function for f, then it is easy to see that,

$$f([x, \widehat{x}]) \subseteq [\underline{\mathsf{F}}(x, \widehat{x}), \overline{\mathsf{F}}(x, \widehat{x})], \quad \text{for all } x \leq \widehat{x}. \tag{1}$$

Given a map $f : \mathbb{R}^r \to \mathbb{R}^q$, one can use [21, Theorem 1] to compute the tight inclusion function, component-wise. Indeed, for every $i \in \{1, \dots, n\}$, one can show that:

$$\underline{\mathsf{F}}_i(x, \widehat{x}) = \min_{z \in [x, \widehat{x}]} f_i(z), \quad \overline{\mathsf{F}}_i(x, \widehat{x}) = \max_{z \in [x, \widehat{x}]} f_i(z) \tag{2}$$

The next Theorem studies the connection between the local Lipschitz constants and the tight inclusion functions.

Theorem 3.2 (Inclusion function vs. Lipschitz constant): Let $f : \mathbb{R}^r \to \mathbb{R}^q$ be a continuous mapping and $\mathsf{F} = \begin{bmatrix} \underline{\mathsf{F}} \\ \overline{\mathsf{F}} \end{bmatrix} : \mathcal{T}^r \to \mathbb{R}^{2q}$ be the tight inclusion function for f. Then, for every $\underline{x} \leq \overline{x}$, we have

$$\|\overline{\mathsf{F}}(\underline{x}, \overline{x}) - \underline{\mathsf{F}}(\underline{x}, \overline{x})\|_\infty \leq \mathrm{Lip}_\infty^{[\underline{x}, \overline{x}]}(f)\|\underline{x} - \overline{x}\|_\infty.$$

Proof: Let $i \in \{1, \dots, k\}$ be such that $\|\overline{\mathsf{F}}(\underline{x}, \overline{x}) - \underline{\mathsf{F}}(\underline{x}, \overline{x})\|_\infty = \left|\overline{\mathsf{F}}_i(\underline{x}, \overline{x}) - \underline{\mathsf{F}}_i(\underline{x}, \overline{x})\right|$. Note that since f is continuous and the box $[\underline{x}, \overline{x}]$ is compact, there exist $\eta^*, \xi^* \in [\underline{x}, \overline{x}]$ such that $\max_{y \in [\underline{x}, \overline{x}]} f_i(y) = f_i(\eta^*), \min_{y \in [\underline{x}, \overline{x}]} f_i(y) = f_i(\xi^*)$. This implies that $\|\overline{\mathsf{F}}(\underline{x}, \overline{x}) - \underline{\mathsf{F}}(\underline{x}, \overline{x})\|_\infty = |f_i(\eta^*) - f_i(\xi^*)| \leq \|f(\xi^*) - f(\eta^*)\|_\infty \leq \mathrm{Lip}_\infty^{[\underline{x}, \overline{x}]} \|\xi^* - \eta^*\|_\infty \leq \mathrm{Lip}_\infty^{[\underline{x}, \overline{x}]}(f)\|\underline{x} - \overline{x}\|_\infty$. ∎

Remark 3.3 (Tight inclusion functions): Theorem 3.2 shows that a tight inclusion function for $y = f(x)$ will provide a tighter over-approximation of the image of f than is attainable from local Lipschitz constants of f. In general, finding tight inclusion functions using (2) is not computationally tractable. This motivates developing efficient methods for estimating the tight inclusion function.

IV. Implicit Neural Networks

We consider the implicit neural network

$$\begin{aligned} z &= \sigma(Wz + Ux + b) := \mathsf{N}(z, x), \\ y &= Cz + c, \end{aligned} \tag{3}$$

where $z \in \mathbb{R}^n$ is the hidden variable, $x \in \mathbb{R}^r$ is the input, $y \in \mathbb{R}^q$ is the output, $W \in \mathbb{R}^{n \times n}$, $U \in \mathbb{R}^{n \times r}$, and $C \in \mathbb{R}^{q \times n}$ are the weight matrices and $b \in \mathbb{R}^n$ and $c \in \mathbb{R}^q$ are bias vectors. Moreover, σ is a diagonal activation function (e.g., ReLU) defined by $\sigma(z_1, \dots, z_n) = (\sigma_1(z_1), \dots, \sigma_n(z_n))^{\mathsf{T}}$, where for every $i \in \{1, \dots, n\}$, the activation function $\sigma_i : \mathbb{R} \to \mathbb{R}$ satisfies $0 \leq \frac{\sigma_i(x) - \sigma_i(y)}{x - y} \leq 1$ for all $x \neq y \in \mathbb{R}$. Compared to traditional neural networks, INNs replace the layers with a fixed-point equation. This change in the structure is known to allow for new architecture possibilities and provide alternative approaches to deep modeling.

Generalized architecture: Notably, FFNNs can be considered as special cases of INNs [3]. Consider the FFNN

$$z^i = \sigma(W_i z^{i-1} + b_i) =: \mathsf{FN}_i(z^{i-1}), \qquad i \in \{1, \dots, k\},$$
$$y = Cz^k + c \tag{4}$$

where $z_0 = x \in \mathbb{R}^r$ is the input. For every $i \in \{1, \dots, k\}$, $z^i \in \mathbb{R}^{n_i}$, $W_i \in \mathbb{R}^{n_{i-1} \times n_i}$, and $b_i \in \mathbb{R}^{n_{i-1}}$ are the weights, the biases, and the hidden variables in the i-th layer of the network, respectively. Finally, $y \in \mathbb{R}^q$ is the output, C and c are the output layer's weight matrix and bias vector. The FFNN (4) is equivalent to the following INN

$$z = \sigma(W^{\mathsf{FN}} z + U^{\mathsf{FN}} x + b) =: \mathsf{IFN}(z, x),$$
$$y = C^{\mathsf{N}} z + c \tag{5}$$

where $z = [z_k, \dots, z_1]^\mathsf{T}$, $b = [b_k, \dots, b_1]^\mathsf{T}$, and W^{FN}, U^{FN}, and C^{FN} are defined as follows:

$$W^{\mathsf{FN}} = \begin{bmatrix} \mathbb{0} & W_k & \mathbb{0} & \cdots & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & W_{k-1} & \cdots & \mathbb{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \cdots & W_1 \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \cdots & \mathbb{0} \end{bmatrix}, \quad U^{\mathsf{FN}} = \begin{bmatrix} \mathbb{0} \\ \mathbb{0} \\ \vdots \\ \mathbb{0} \\ W_0 \end{bmatrix}$$
$$C^{\mathsf{FN}} = \begin{bmatrix} C & \mathbb{0} & \mathbb{0} & \cdots & \mathbb{0} \end{bmatrix}.$$

Using this perspective, implicit neural networks generalize FFNNs by allowing arbitrary interconnections between layers leading to full weight matrices W, U, and C.

Alternative deep modeling: By replacing the notion of layer with an algebraic equation, implicit neural networks provide a novel perspective toward deep modeling. Consider the class of FFNN where weights and biases are equal for each layer (i.e., the network is weight-tied) and the input is injected to each layer given by

$$z^i = \sigma(Wz^{i-1} + Ux + b), \quad i \in \{1, \dots, k\},$$
$$y = Cz_k + c \tag{6}$$

where $z^0 = x$. While weight-tying may appear restrictive, it is usually considered as a form of regularization that stabilizes training and significantly reduces the model size [2]. If the depth of the network increases, i.e., $k \to \infty$ and the iteration (6) converges, then the weight-tied input-injected neural network (6) is equivalent to the implicit neural network (3). Using this perspective, INNs provide a depth-independent alternative to deep FFNNs.

Suppose that, for every input $x \in \mathbb{R}^r$, the implicit neural network (3) has a unique fixed point $z^*(x) \in \mathbb{R}^n$. Then, the *input-output map* $f : \mathbb{R}^r \to \mathbb{R}^q$ is given by

$$f(x) := y = Cz^*(x) + c. \tag{7}$$

In the next section, our goal is to provide estimates for the tight inclusion function of the input-output map f.

V. Reachability Analysis of Implicit Neural Networks

In this section, we use mixed monotone system theory to present a framework for estimating the tight inclusion function for the input-output map of INNs. Given an implicit neural network (3) and input bounds $\underline{x} \le \overline{x} \in \mathbb{R}^r$, we first introduce the embedding map $\mathsf{N}^{\mathrm{E}} : \mathbb{R}^{2n} \times \mathbb{R}^{2r} \to \mathbb{R}^n$ by

$$\mathsf{N}^{\mathrm{E}}(\underline{z}, \overline{z}, \underline{x}, \overline{x}) = \sigma(\lceil W \rceil^{\mathrm{Mzl}} \underline{z} + \lfloor W \rfloor^{\mathrm{Mzl}} \overline{z} + [U]^+ \underline{x} + [U]^- \overline{x} + b).$$

Using the embedding map N^{E}, we define the *embedded implicit neural network* associated to (3) by

$$\begin{bmatrix} \underline{z} \\ \overline{z} \end{bmatrix} = \begin{bmatrix} \mathsf{N}^{\mathrm{E}}(\underline{z}, \overline{z}, \underline{x}, \overline{x}) \\ \mathsf{N}^{\mathrm{E}}(\overline{z}, \underline{z}, \overline{x}, \underline{x}) \end{bmatrix}, \quad \begin{bmatrix} \underline{y} \\ \overline{y} \end{bmatrix} = \begin{bmatrix} [C]^+ & [C]^- \\ [C]^- & [C]^+ \end{bmatrix} \begin{bmatrix} \underline{z} \\ \overline{z} \end{bmatrix} + \begin{bmatrix} c \\ c \end{bmatrix}. \tag{8}$$

The embedded INN (8) can be considered as an INN with the box input $[\underline{x}, \overline{x}]$ and the box output $[\underline{y}, \overline{y}]$. In the next theorem, we use the embedded system (8) to obtain an inclusion function for the input-output map (7).

Theorem 5.1 (Inclusion function via embedded network): Consider the implicit neural network (3) and its associated embedded implicit neural network (8). Suppose that there exists $\eta \in \mathbb{R}^n_{>0}$ such that $\mu_{\infty,[\eta]^{-1}}(W) < 1$. For every $\underline{x} \le x \le \overline{x}$, and every $\alpha \in (0, \alpha^* := [1 - \min_{i \in \{1,\dots,n\}}(W_{ii})_-]^{-1}]$, the following statements hold:

(i) the iterations $\begin{bmatrix} \underline{z}^{k+1} \\ \overline{z}^{k+1} \end{bmatrix} = \begin{bmatrix} \mathsf{N}^{\mathrm{E}}_\alpha(\underline{z}^k, \overline{z}^k, \underline{x}, \overline{x}) \\ \mathsf{N}^{\mathrm{E}}_\alpha(\overline{z}^k, \underline{z}^k, \overline{x}, \underline{x}) \end{bmatrix}$ are contracting with respect to the norm $\|\cdot\|_{\infty, I_2 \otimes [\eta]^{-1}}$ and converge to the unique fixed-point $\begin{bmatrix} \underline{z}^* \\ \overline{z}^* \end{bmatrix}$ of the embedded INN (8);

(ii) the iterations $z^{k+1} = \mathsf{N}_\alpha(z^k, x)$ are contracting with respect to the norm $\|\cdot\|_{\infty,[\eta]^{-1}}$ and converges to the unique fixed point $z^* \in [\underline{z}^*, \overline{z}^*]$ of the INN (3);

(iii) the map $\mathsf{F}^{\mathsf{N}} : \mathcal{T}^r \to \mathbb{R}^{2q}$ defined by

$$\underline{\mathsf{F}}^{\mathsf{N}}(\underline{x}, \overline{x}) = [C]^+ \underline{z}^* + [C]^- \overline{z}^* + c$$
$$\overline{\mathsf{F}}^{\mathsf{N}}(\underline{x}, \overline{x}) = [C]^+ \overline{z}^* + [C]^- \underline{z}^* + c \tag{9}$$

is an inclusion function for f defined in (7).

Proof: Regarding part (i), we define $\tilde{\sigma} = I_2 \otimes \sigma$, the map $\mathsf{G} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ by $\mathsf{G}(\underline{z}, \overline{z}) = \begin{bmatrix} \lceil W \rceil^{\mathrm{Mzl}} \underline{z} + \lfloor W \rfloor^{\mathrm{Mzl}} \overline{z} \\ \lfloor W \rfloor^{\mathrm{Mzl}} \underline{z} + \lceil W \rceil^{\mathrm{Mzl}} \overline{z} \end{bmatrix}$ and the matrices $D = \begin{bmatrix} [U]^+ & [U]^- \\ [U]^- & [U]^+ \end{bmatrix}$ and $w = [\underline{x}, \overline{x}]^\mathsf{T}$. Then define $\tilde{\sigma}^{\mathsf{G}} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ as follows

$$\tilde{\sigma}^{\mathsf{G}}(\underline{z}, \overline{z}, w) := \begin{bmatrix} \mathsf{N}^{\mathrm{E}}(\underline{z}, \overline{z}, \underline{x}, \overline{x}) \\ \mathsf{N}^{\mathrm{E}}(\overline{z}, \underline{z}, \overline{x}, \underline{x}) \end{bmatrix} = \tilde{\sigma}(\mathsf{G}(\underline{z}, \overline{z}) + Dw + I_2 \otimes b).$$

The assumptions on each scalar activation function imply that (i) $\tilde{\sigma} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ is non-expansive with respect to $\|\cdot\| := \|\cdot\|_{\infty, I_2 \otimes [\eta]^{-1}}$ and (ii) for every $p, q \in \mathbb{R}$, there exists $\theta_i \in [0, 1]$ such that $\sigma_i(p) - \sigma_i(q) = \theta_i(p - q)$ or in the matrix form $\tilde{\sigma}(\mathbf{p}) - \tilde{\sigma}(\mathbf{q}) = \Theta(\mathbf{p} - \mathbf{q})$ where $\Theta \in \mathbb{R}^{2n \times 2n}$

is a diagonal matrix with diagonal elements $\theta_i \in [0,1]$ and $\mathbf{p}, \mathbf{q} \in \mathbb{R}^{2n}$. As a result, for every $y_1, y_2 \in \mathbb{R}^{2n}$, we have

$$\begin{aligned}\|\tilde{\sigma}^{\mathsf{G}}_\alpha(y_1,w) - \tilde{\sigma}^{\mathsf{G}}_\alpha(y_2,w)\| &= \|(1-\alpha)(y_1-y_2) + \alpha\Theta(\mathsf{G}(y_1)-\mathsf{G}(y_2))\| \\ &\leq \sup_{y\in\mathbb{R}^{2n}} \|(1-\alpha)I_{2n} + \alpha\Theta D\mathsf{G}(y)\|\|y_1-y_2\|\end{aligned}$$

where the inequality holds by the mean value theorem. Then, for every $\alpha \in (0, [1-\min_i \inf_{y\in\mathbb{R}^{2n}}(\Theta D\mathsf{G}(y))_{ii}]^{-1}]$,

$$\begin{aligned}\|I_{2n}+\alpha(-I_{2n}+\Theta D\mathsf{G}(y))\| &= 1+\alpha\mu_{\infty,I_2\otimes[\eta]^{-1}}\big(-I_{2n}+\Theta D\mathsf{G}(y)\big) \\ &= 1+\alpha\big(-1+\mu_{\infty,I_2\otimes[\eta]^{-1}}(\Theta D\mathsf{G}(y))\big) \\ &\leq 1+\alpha\big(-1+\mu_{\infty,I_2\otimes[\eta]^{-1}}(D\mathsf{G}(y))^+\big) \\ &\leq 1-\alpha(1-\mu_{\infty,[\eta]^{-1}}(W)^+) < 1,\end{aligned}$$

where the first equality holds by [17, Lemma 7(i)], the second equality holds by translation property of matrix measures, the third inequality holds by [17, Lemma 8(i)], and the fourth inequality holds by the definition of matrix measure. Moreover, since $\theta_i \in [0,1]$, we have $\theta_i(D\mathsf{G})_{ii} \geq (D\mathsf{G})^-_{ii}$, for every $i \in \{1,\dots,2n\}$. This means that

$$\inf_{y\in\mathbb{R}^{2n}}(\Theta D\mathsf{G}(y))_{ii} \geq \inf_{y\in\mathbb{R}^{2n}}(D\mathsf{G}_{ii}(y))^- = (W_{ii})^-.$$

This implies that, for every $\alpha \in (0,\alpha^*]$,

$$\begin{aligned}\|\tilde{\sigma}^{\mathsf{G}}_\alpha(x_1,u) - \tilde{\sigma}^{\mathsf{G}}_\alpha(x_2,u)\| &\leq (1-\alpha(1-\mu_{\infty,[\eta]^{-1}}(W)^+))\|x_1-x_2\|\end{aligned}$$

Since $1-\alpha(1-\mu_{\infty,[\eta]^{-1}}(W)^+) < 1$, $\tilde{\sigma}^{\mathsf{G}}_\alpha(\cdot,w)$ is a contraction mapping with respect to $\|\cdot\|_{\infty,I_2\otimes[\eta]^{-1}}$ for every $\alpha \in (0,\alpha^*]$. It is easy to see that $\tilde{\sigma}^{\mathsf{G}}_\alpha$ and $\tilde{\sigma}^{\mathsf{G}}$ have the identical fixed-points, for every $\alpha \in [0,1]$. Therefore the iterations in part (i) converge to the unique fixed point of the embedded INN (8). Regarding part (ii), the proof follows by applying the same argument as in the proof of part (i) and using $\sigma(Wz+Ux+b)$ instead of $\tilde{\sigma}^{\mathsf{G}}(\underline{z},\overline{z},\underline{x},\overline{x})$. Now, we show that $\underline{z}^* \leq z^* \leq \overline{z}^*$. We choose the initial condition $\begin{bmatrix}\underline{z}^0\\ \overline{z}^0\end{bmatrix}$ for the iterations in part (i) and choose an initial condition $z^0 \in \mathbb{R}^n$ satisfying $\underline{z}^0 \leq z^0 \leq \overline{z}^0$ for the iterations in part (ii). We prove by induction that, for every $k \in \mathbb{Z}_{\geq 0}$, we have $\underline{z}^k \leq z^k \leq \overline{z}^k$. Suppose that this claim is true for $k \in \{1,\dots,m\}$ and we show that this claim is true for $k = m+1$. We first define $p = \lceil W\rceil^{\mathrm{Mzl}}\underline{z}^m + \lfloor W\rfloor^{\mathrm{Mzl}}\overline{z}^m + [U]^+\underline{x} + [U]^-\overline{x} + b$ and $q = Wz^m + Ux + b$. Then we have

$$\begin{aligned}\underline{z}^{m+1} - z^{m+1} &= (1-\alpha^*)(\underline{z}^m - z^m) + \alpha^*(\sigma(p)-\sigma(q)) \\ &= \big((1-\alpha^*)I_n + \alpha^*\Theta\lceil W\rceil^{\mathrm{Mzl}}\big)(\underline{z}^m - z^m) \\ &\quad + \alpha^*\Theta\lfloor W\rfloor^{\mathrm{Mzl}}(\overline{z}^m - z^m) \\ &\quad + \alpha^*\Theta[U]^+(\underline{x}-x) + \alpha^*\Theta[U]^-(\overline{x}-x),\end{aligned}$$

where the non-negative diagonal matrix $\Theta = \mathrm{diag}(\theta_i) \in \mathbb{R}^n$ is defined as follows: for every $i \in \{1,\dots,n\}$, $\theta_i \in [0,1]$ is such that $\sigma_i(p_i) - \sigma_i(q_i) = \theta_i(p_i - q_i)$. Moreover, we know that $\Theta\lfloor W\rfloor^{\mathrm{Mzl}} \leq \mathbb{0}_{n\times n}$ and, for every $i \in \{1,\dots,n\}$,

$$(1-\alpha^*) + \alpha^*\theta_i W_{ii} \geq (1-\alpha^*) + \alpha^* W^-_{ii} \geq 0.$$

This implies that $(1-\alpha^*)I_n + \alpha^*\Theta\lceil W\rceil^{\mathrm{Mzl}} \geq \mathbb{0}_{n\times n}$. Additionally, we have $\Theta[U]^+ \geq \mathbb{0}_{n\times r}$ and $\Theta[U]^- \leq \mathbb{0}_{n\times r}$. Therefore, using the induction assumption, we get $\underline{z}^{m+1} - z^{m+1} \leq \mathbb{0}_n$. Similarly, one can show that $z^{m+1} - \overline{z}^{m+1} \leq \mathbb{0}_n$. As a consequence, $\underline{z}^* = \lim_{k\to\infty}\underline{z}^k \leq \lim_{k\to\infty} z^k = z^* \leq \lim_{k\to\infty}\overline{z}^k = \overline{z}^*$. This proves part (ii). The proof of part (iii) follows easily from parts (i) and (ii) and by checking the properties of inclusion functions from Definition 3.1. ■

Remark 5.2 (Mixed monotone contracting approach): Theorem 5.1 can be interpreted as a dynamical system approach to study robustness of INNs. Indeed, the α-average iteration in part (ii) is the forward Euler discretization of the dynamical system $\frac{dz}{dt} = -z + \mathsf{N}(z,x)$. The convergence of the iterations is due to the contraction property of the dynamical system and the estimate for the inclusion function is due to the mixed monotonicity of the dynamical system associated with the embedded INN [20].

VI. Feedforward vs. Implicit Neural Networks

In this section, we compare the robust training framework developed in Sections (V) and (VII) with the IBP approach developed in [11]. Consider a k-layer FFNN with input-output map $f(x) =: y$ defined by (4). For every $i \in \{1,\dots,k\}$, following [11], [12], we can obtain layer-wise upper and lower bounds for the hidden variables as follows:

$$\begin{aligned}\underline{z}^{i+1} &:= \mathsf{FN}^{\mathrm{E}}_i(\underline{z}^i,\overline{z}^i) = \sigma([W_i]^+\underline{z}^i + [W_i]^-\overline{z}^i + b_i), \\ \overline{z}^{i+1} &:= \mathsf{FN}^{\mathrm{E}}_i(\overline{z}^i,\underline{z}^i) = \sigma([W_i]^+\overline{z}^i + [W_i]^-\underline{z}^i + b_i).\end{aligned}$$

By applying this bounding technique recursively, one can obtain the upper bound $\overline{y}$ and lower bound $\underline{y}$ for the output of the FFNN. The IBP inclusion function $\mathsf{F}^{\mathsf{FN}} = \begin{bmatrix}\underline{\mathsf{F}}^{\mathsf{FN}}\\ \overline{\mathsf{F}}^{\mathsf{FN}}\end{bmatrix} : \mathcal{T}^r \to \mathbb{R}^{2q}$ for the input-output map f is then defined by:

$$\underline{\mathsf{F}}^{\mathsf{FN}}(\underline{x},\overline{x}) = \underline{y}, \qquad \overline{\mathsf{F}}^{\mathsf{FN}}(\underline{x},\overline{x}) = \overline{y}, \tag{10}$$

In the next two subsections, we use the two perspective *generalized architecture* and *alternative deep modeling* toward INNs, to establish connections between the IBP approach in [11] and our mixed monotone contracting approach.

A. Generalized architecture

By considering finite-depth FFNNs as a special case of implicit neural networks, one can show that our mixed monotone contracting approach is a generalization of the IBP approach in [11] to INNs.

Theorem 6.1 (Embedded feedforward neural networks): Consider the FFNN (4) with the associated implicit neural network (5). The following statements hold:

(i) for every $i \in \{1,\dots,k\}$, the function $(\underline{z},\overline{z}) \mapsto \begin{bmatrix}\mathsf{FN}^{\mathrm{E}}_i(\underline{z},\overline{z})\\ \mathsf{FN}^{\mathrm{E}}_i(\overline{z},\underline{z})\end{bmatrix}$ is a tight inclusion function for the ith layer evaluation map $\mathsf{FN}_i(z) := \sigma(W_i z + b_i)$;

(ii) there exists $\eta \in \mathbb{R}^n_{>0}$ such that $\mu_{\infty,[\eta]^{-1}}(W^{\mathsf{FN}}) < 1$.

If $\mathsf{F}^{\mathsf{IFN}}$ is the inclusion function obtained from (5) using mixed monotone contracting approach and F^{FN} is the inclusion function (10) obtained using IBP approach, then

(iii) $\mathsf{F}^{\mathsf{IFN}}(\underline{x},\overline{x}) = \mathsf{F}^{\mathsf{FN}}(\underline{x},\overline{x})$, for every $\underline{x} \leq \overline{x} \in \mathbb{R}^r$.

The proof of Theorem 6.1(ii) and (iii) is based on choosing $[\eta]^{-1} = \mathrm{diag}(\delta^{-1}, \dots, \delta^{-n})$ for $\delta > 0$ sufficiently small and then applying Theorem 5.1.

B. Alternative deep modeling

By separating the notion of depth from the layer-wise evaluation, our mixed monotone contracting approach can be used to estimate the reachable sets of deep weight-tied FFNNs. For the weight-tied FFNN (6), we define

$$\mathsf{WFN}_i^{\mathrm{E}}(\underline{z}, \overline{z}, \underline{x}, \overline{x}) = \sigma([W]^+\underline{z} + [W]^-\overline{z} + [U]^+\underline{x} + [U]^-\overline{x} + b).$$

By replacing $\mathsf{FN}_i^{\mathrm{E}}$ with $\mathsf{WFN}_i^{\mathrm{E}}$ in (10), we the IBP inclusion function $\mathsf{F}^{\mathsf{WFN}} : \mathcal{T}^r \to \mathbb{R}^{2q}$.

Theorem 6.2 (Weight-tied infinite-layer neural networks): Suppose that $\rho(|W|) < 1$ and let $\eta \in \mathbb{R}^n_{>0}$ be the right Perron eignvector of $|W|$. Then for the weight-tied FFNN (6),

(i) $\lim_{i\to\infty} \begin{bmatrix} \underline{z}^i \\ \overline{z}^i \end{bmatrix} = \begin{bmatrix} \underline{w}^* \\ \overline{w}^* \end{bmatrix}$ for some $\underline{w}^* \le \overline{w}^* \in \mathbb{R}^n$;

(ii) $\lim_{i\to\infty} z^i = w^*$ for some $w^* \in [\underline{w}^*, \overline{w}^*]$;

Moreover, for the implicit neural network (3),

(iii) $\mu_{\infty,[\eta]^{-1}}(W) < 1$;

If $\mathsf{F}^{\mathsf{WFN}}$ is the inclusion function by the IBP approach as $k \to \infty$ and F^{N} is the inclusion function from (3), then

(iv) for every $\underline{x} \le \overline{x} \in \mathbb{R}^r$, $\underline{\mathsf{F}}^{\mathsf{N}}(\underline{x}, \overline{x}) \ge \underline{\mathsf{F}}^{\mathsf{WFN}}(\underline{x}, \overline{x})$, and $\overline{\mathsf{F}}^{\mathsf{N}}(\underline{x}, \overline{x}) \le \overline{\mathsf{F}}^{\mathsf{WFN}}(\underline{x}, \overline{x})$

The proof of Theorem 6.2(iii) is based on the inequality $\mu_{\infty,[\eta]^{-1}}(W) \le \rho(|W|) < 1$ and the proof of Theorem 6.2(iv) follows from $[W]^+ \le \lceil W \rceil^{\mathrm{Mzl}}$.

VII. Training Robust Implicit Neural Networks

A. Certified adversarial robustness for classification tasks

We say an INN is certifiably robust for input x if its prediction at x is verifiably constant within a given ℓ_∞ ball around x. We refer to [20] for a rigorous definition of certified adversarial robustness. We use the embedded INN (8) to obtain a sufficient condition for certified robustness. Given a robustness radius $\epsilon > 0$, for every input $x \in \mathbb{R}^r$, we define $\underline{x} = x - \epsilon\mathbb{1}_r, \overline{x} = x + \epsilon\mathbb{1}_r$. Following [12, Eq. 3] and [20], for each input $x' \in [\underline{x}, \overline{x}]$, we define the *relative classifier variable*, $m^x(x') \in \mathbb{R}^q$ by

$$m^x(x') = f(x')_i \mathbb{1}_q - f(x'), \tag{11}$$

where i is the correct label of x. Note that $m^x(x')_j > 0$ for all $j \neq i$ if and only if x' is labeled the same as x by the neural network. Therefore, we write $m^x(x') = T^x f(x') = T^x C z^*(x') + T^x c$, for suitable specification matrix $T^x \in \{-1, 0, 1\}^{q\times q}$ defined via the linear transformation (11). Moreover, if there exists $\eta \in \mathbb{R}^n_{>0}$ so that $\mu_{\infty,[\eta]^{-1}}(W) < 1$, then we can use Theorem 5.1 to define

$$\underline{m}^x(\underline{x}, \overline{x}) = [T^x C]^+ \underline{z}^*(\underline{x}, \overline{x}) + [T^x C]^- \overline{z}^*(\underline{x}, \overline{x}) + T^x c.$$

Moreover, $\min_{j\neq i}(\underline{m}^x(\underline{x}, \overline{x})_j > 0$ is a sufficient condition for certified adversarial robustness of the INN [20].

B. Training optimization problem

We aim to design optimization problems to train a neural network which is robust to input perturbations with ℓ_∞-norm smaller than some ϵ. Let $\mathcal{L}$ be the cross-entropy loss function and assume that $\{(\widehat{x}^l, \widehat{y}^l)\}_{l=1}^N$ is a set of N labeled data points used for training. For every $l \in \{1, \dots, N\}$, we define the following upper and the lower bounds on the input $\widehat{x}^l$ by $\underline{x}^l = \widehat{x}^l - \epsilon\mathbb{1}_r$ and $\overline{x}^l = \widehat{x}^l + \epsilon\mathbb{1}_r$. We use the robust optimization framework [7] for designing robust neural networks. Our objective is to minimize the robust loss function $\sum_{l=1}^N \max_{x\in[\underline{x}^l, \overline{x}^l]} \mathcal{L}(f(\widehat{x}^l), \widehat{y}^l)$ on the training data. Using [6, Theorem 2], for the cross-entropy loss, and for $\underline{m}^l := \underline{m}^{\widehat{x}^l}(\underline{x}^l, \overline{x}^l)$ and every $l \in \{1, \dots, N\}$,

$$\mathcal{L}(f(\widehat{x}^l), \widehat{y}^l) \le \mathcal{L}(-\underline{m}^l, \widehat{y}^l), \quad \text{for all } x \in [\underline{x}^l, \overline{x}^l].$$

As pointed out in [11] for FFNNs, using the loss function $\mathcal{L}(-\underline{m}^l, \widehat{y}^l)$ in the training can lead to convergence instability and difficulty in training. To improve the stability of the training, following [11], we instead use a convex combination of the empirical risk loss and the robust loss. Therefore, for $T^l := T^{\widehat{x}^l}$ we get the following training problem:

$$\begin{aligned} \min_{W,U,C,b,c,\eta} \quad & \sum_{l=1}^N (1-\kappa)\mathcal{L}(y^l, \widehat{y}^l) + \kappa\mathcal{L}(-\underline{m}^l, \widehat{y}^l), \\ & \begin{bmatrix} \underline{z}^l \\ \overline{z}^l \end{bmatrix} = \begin{bmatrix} \mathsf{N}^{\mathrm{E}}(\underline{z}^l, \overline{z}^l, \underline{x}^l, \overline{x}^l) \\ \mathsf{N}^{\mathrm{E}}(\overline{z}^l, \underline{z}^l, \overline{x}^l, \underline{x}^l) \end{bmatrix}, \\ & \underline{m}^l = [T^l C]^+ \underline{z}^l + [T^l C]^- \overline{z}^l + T^l c, \quad z^l = \mathsf{N}(z^l, \widehat{x}^l), \\ & y^l = Cz^l + c, \quad \mu_{\infty,[\eta]^{-1}}(W) \le \gamma. \end{aligned} \tag{12}$$

where $\kappa \in [0, 1]$ and $\gamma \in (-\infty, 1)$ are hyperparameters.

VIII. Numerical Experiments

In this section we provide an experimental comparison between the robustness of FFNNs and INNs trained with and without IBP and mixed monotonicity, respectively[1].

Experimental setup: We consider the MNIST dataset, which contains 70000 28×28 pixel images of handwritten digits. For training, pixels are normalized into the range $[0, 1]$. All INNs have $n = 100$ neurons with ReLU activation, while we consider five-layer FFNNs ($784 \to 100 \to 75 \to 50 \to 40 \to 25 \to 10$) with ReLU activation.

Each model was trained for 40 epochs using the Adam optimizer. INNs that were trained using mixed monotonicity and FFNNs that were trained using IBP have $\epsilon_{\text{test}} = 0.1$ and $\kappa_{\text{nom}} = 0.75$. From epochs 1 to 10, κ and ϵ are set to 0 so the models undergo regular (nonrobust) training. From epochs 11 to 20, ϵ and κ are linearly increased such that at epoch 20, $\epsilon = \epsilon_{\text{test}}$ and $\kappa = \kappa_{\text{nom}}$. Regarding training INNs, we follow the non-Euclidean monotone operator framework described in [17]. We impose $\mu_{\infty,[\eta]^{-1}}(W) \le 0$ for some $\eta \in \mathbb{R}^n_{>0}$.

10 FFNNs and 10 INNs were trained; 5 of each were trained using IBP or mixed monotonicity (for feedforwad and implicit, respectively) and 5 of each were trained without

[1]Code to reproduce the experiments is available at https://github.com/davydovalexander/robust-inn-mm.

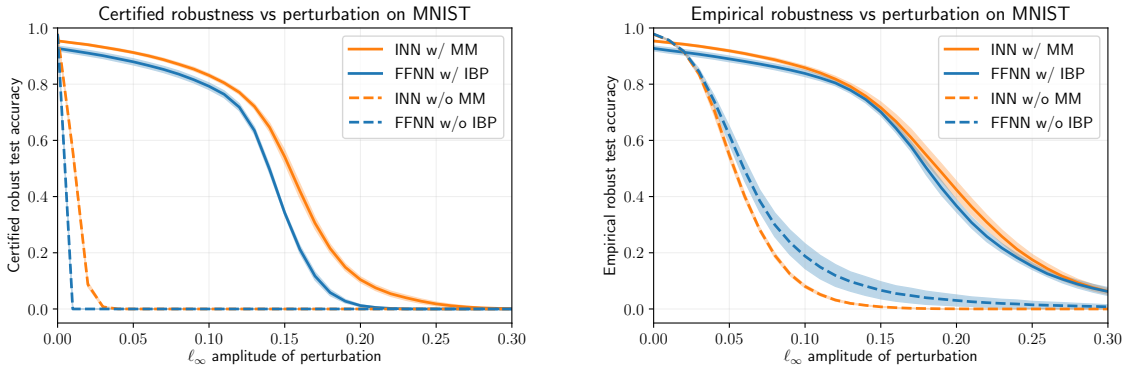

Fig. 1: Performance comparison on the MNIST test data between INNs trained with and without mixed monotonicity (MM) and 5-layer FFNNs trained with and without IBP. The INNs have 89710 trainable parameters and FFNNs have 93200 trainable parameters. The left plot shows the certified robust accuracy of the models computed using either MM or IBP while the right plot shows the empirical robustness of the models against a PGD attack. In each plot, dark lines correspond to the mean accuracy across 5 neural networks while light envelopes around the dark lines correspond to one standard deviation.

any robust optimization (i.e., $\epsilon_{\text{test}} = 0$). Figure 1 provides plots of certified adversarial robustness via the corresponding interval reachability technique and the empirically-observed robustness against a projected gradient descent (PGD) attack.

Evaluation summary: Regarding certified robustness, at an ℓ_∞ perturbation radius of 0.1, we observe that INNs trained using mixed monotonicity had, on average, an accuracy of 83.13%, while FFNNs trained using IBP had, on average an accuracy of 79.26%. We additionally observe that at the cost of a few percentage points in clean accuracy, both INNs and FFNNs trained robustly vastly outperform non-robustly trained models in both certified and empirical robustness. For example, at an ℓ_∞ perturbation radius of 0.1, INNs trained without mixed monotonicity have an empirical accuracy of 8.04%, while INNs trained with mixed monotonicity have an accuracy of 85.84%, indicating an order of magnitude improvement in empirical robustness.

IX. Conclusion

We develop a computationally efficient algorithm for training robust INNs. Moreover, we provide theoretical and empirical evidence in support of the following claims: (i) robustly-trained INNs are more robust than comparably-trained FFNNs, (ii) inclusion functions provide tighter estimates than Lipschitz constants, (iii) robustly-trained networks enjoy much stronger robustness properties than their non-robustly trained counterparts.

References

[1] B. Amos and J. Z. Kolter, "OptNet: Differentiable optimization as a layer in neural networks," in *International Conference on Machine Learning*, 2017.

[2] S. Bai, J. Z. Kolter, and V. Koltun, "Deep equilibrium models," in *Advances in Neural Information Processing Systems*, 2019.

[3] L. El Ghaoui, F. Gu, B. Travacca, A. Askari, and A. Tsai, "Implicit deep learning," *SIAM Journal on Mathematics of Data Science*, vol. 3, no. 3, pp. 930–958, 2021.

[4] A. Kag, Z. Zhang, and V. Saligrama, "RNNs incrementally evolving on an equilibrium manifold: A panacea for vanishing and exploding gradients?" in *International Conference on Learning Representations*, 2020.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[6] E. Wong and J. Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International Conference on Machine Learning*, 2018, pp. 5286–5295.

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Machine Learning*, 2018.

[8] A. Virmaux and K. Scaman, "Lipschitz regularity of deep neural networks: analysis and efficient estimation," in *Advances in Neural Information Processing Systems*, vol. 31, 2018, p. 3839–3848.

[9] P. Pauli, A. Koch, J. Berberich, P. Kohler, and F. Allgower, "Training robust neural networks using Lipschitz bounds," *IEEE Control Systems Letters*, vol. 6, pp. 121–126, 2022.

[10] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, "Formal security analysis of neural networks using symbolic intervals," in *USENIX Conference on Security Symposium*, 2018, p. 1599–1614.

[11] S. Gowal, K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. A. Mann, and P. Kohli, "Scalable verified training for provably robust image classification," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 4841–4850.

[12] H. Zhang, H. Chen, C. Xiao, S. Gowal, R. Stanforth, B. Li, D. Boning, and C.-J. Hsieh, "Towards stable and efficient training of verifiably robust neural networks," in *International Conference on Learning Representations*, 2020.

[13] C. Wei and J. Z. Kolter, "Certified robustness for deep equilibrium models via interval bound propagation," in *International Conference on Learning Representations*, 2022.

[14] M. Fazlyab, M. Morari, and G. J. Pappas, "Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming," *IEEE Transactions on Automatic Control*, vol. 67, no. 1, pp. 1–15, 2022.

[15] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient smt solver for verifying deep neural networks," in *Computer Aided Verification*. Springer International Publishing, 2017, pp. 97–117.

[16] C. Pabbaraju, E. Winston, and J. Z. Kolter, "Estimating Lipschitz constants of monotone deep equilibrium models," in *International Conference on Learning Representations*, 2021.

[17] S. Jafarpour, A. Davydov, A. V. Proskurnikov, and F. Bullo, "Robust implicit networks via non-Euclidean contractions," in *Advances in Neural Information Processing Systems*, 2021.

[18] M. Revay, R. Wang, and I. R. Manchester, "Lipschitz bounded equilibrium networks," *arXiv preprint arXiv:2010.01732*, 2020.

[19] T. Chen, J. B. Lasserre, V. Magron, and E. Pauwels, "Semialgebraic representation of monotone deep equilibrium models and applications to certification," in *Advances in Neural Information Processing Systems*, 2021.

[20] S. Jafarpour, M. Abate, A. Davydov, F. Bullo, and S. Coogan, "Robustness certificates for implicit neural networks: A mixed monotone contractive approach," in *Learning for Dynamics & Control Conference*, June 2022.

[21] L. Yang and N. Ozay, "Tight decomposition functions for mixed monotonicity," in *IEEE Conference on Decision and Control (CDC)*, 2019, pp. 5318–5322.