Efficient Global Robustness Certification of Neural Networks via Interleaving Twin-Network Encoding (Extended Abstract) *

Zhilu Wang¹ , Chao Huang² , Qi Zhu¹

¹Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL, USA
²Department of Computer Science, University of Liverpool, Liverpool, UK
zhilu.wang@u.northwestern.edu, chao.huang2@liverpool.ac.uk, qzhu@northwestern.edu

Abstract

The robustness of deep neural networks in safety-critical systems has received significant interest recently, which measures how sensitive the model output is under input perturbations. While most previous works focused on the local robustness property, the studies of the global robustness property, i.e., the robustness in the entire input space, are still lacking. In this work, we formulate the global robustness certification problem for ReLU neural networks and present an efficient approach to address it. Our approach includes a novel interleaving twin-network encoding scheme and an over-approximation algorithm leveraging relaxation and refinement techniques. Its timing efficiency and effectiveness are evaluated and compared with other state-of-the-art global robustness certification methods, and demonstrated via case studies on practical applications.

1 Introduction

Deep neural networks (DNNs) could be vulnerable to small adversarial perturbations on their inputs [Biggio et al., 2013]. The formally-defined robustness metric of a DNN tries to bound such uncertain behavior, measuring how much the network’s output may deviate when its input has a bounded perturbation. The local robustness problem has been extensively studied, with formal methods developed to bound the output range for a bounded disturbance around a given input [Katz et al., 2017; Singh et al., 2019; Huang et al., 2020b; Zhang et al., 2018; Wang et al., 2021a]. However, it is hard to apply these techniques in safety verification of a dynamic system (e.g., an autonomous vehicle) [Zhu et al., 2020; Wang et al., 2021d; Liu et al., 2022] as we will need to conduct local robustness analysis during runtime for each input sample that the system encounters or may encounter, and they are typically too computationally expensive for that.

This challenge motivates us to address the safety of DNN-enabled dynamic systems by considering the problem of global robustness, which measures the worst-case DNN output deviation against bounded perturbation for all possible input values. We can conduct such worst-case analysis offline, decoupling it from the safety verification [Huang et al., 2019; Fan et al., 2020; Huang et al., 2022] by applying the maximum deviation for all possible inputs. However, the previous MILP (mixed-integer linear programming) or SMT-based techniques for global robustness analysis [Katz et al., 2017; Chen et al., 2021] are still too complex for DNNs in practical systems, even in offline computation. Various approaches were proposed to tackle the complexity challenge, but they are either still computationally expensive, e.g., with region-based robustness analyses [Gopinath et al., 2018; Mangal et al., 2019], or lack the deterministic guarantees, e.g., with sampling-based techniques [Ruan et al., 2019; Bastani et al., 2016; Mangal et al., 2019].

In this work, we propose an efficient certification approach to over-approximate the global robustness. Our approach introduces a novel network encoding structure, namely interleaving twin-network encoding, to compare two copies of the neural network side-by-side under different inputs, with extra interleaving dependencies added between them to improve efficiency. Our approach also includes over-approximation techniques based on network decomposition and LP (linear programming) relaxation, to further reduce the computation complexity. To the best of our knowledge, our approach is the first global robustness over-approximation method that certifies the robustness among the entire input domain with sound and deterministic guarantee. Experiments show that our approach is much more efficient and scalable than the exact global robustness methods such as Reluplex [Katz et al., 2017], with tight over-approximation. A case study of close-loop control system safety verification with perception DNN component further demonstrates the potential of our approach in practical systems.

2 Global Robustness Certification

2.1 Problem Formulation

An n-layer neural network $F : \mathbb{R}^{m_0} \rightarrow \mathbb{R}^{m_n}$ maps input $x^{(i)} \in \mathbb{R}^m$ into output $y^{(n)} \in \mathbb{R}^{m_n}$. The output of layer $i$ is denoted as $x^{(i)} \in \mathbb{R}^m$. The mapping between two consecutive layers $x^{(i-1)}$ and $x^i$ is composed with a linear transformation $y^{(i)} = W^{(i)}x^{(i-1)} + b^{(i)}$ and (optionally) a ReLU

---

*This is an extended abstract of a paper that won the Best Paper Award at Design, Automation and Test in Europe Conference (DATE 2022).
activation function, where the linear transformation result is denoted as variable \( y^{(1)} \in \mathbb{R}^{m_1} \). An illustrating example of a 2-layer neural network is shown in Fig. 1. For simplicity, all bias terms \( b^{(j)} \) are 0.

As defined in [Katz et al., 2017; Wang et al., 2021], the neural network global robustness measures the worst-case output variation when there is a small input perturbation for any possible input sample in the entire input domain \( X \).

**Definition 1** (Global Robustness). The \( j \)-th output of a neural network \( F \) is \((\delta, \varepsilon)\)-globally robust in the input domain \( X \) iff
\[
\forall x^{(0)}, \hat{x}^{(0)} \in X, \|x^{(0)} - \hat{x}^{(0)}\|_\infty \leq \delta \implies |y^{(j)} - \hat{y}^{(j)}| \leq \varepsilon,
\]
where \( x^{(n)} = F(x^{(0)}) \) and \( \hat{x}^{(n)} = F(\hat{x}^{(0)}) \).

In this work, we tackle the problem of measuring how robust a neural network is, as formally defined in Problem 1:

**Problem 1.** For a neural network \( F \), given an input perturbation bound \( \delta \), determine the minimal output variation bound \( \varepsilon \) such that \( F \) is guaranteed to be \((\delta, \varepsilon)\)-globally robust.

[Katz et al., 2017] proposes to solve this problem by encoding two copies of the neural network side by side, as illustrated in the left part of Fig. 2. \( \hat{x}^{(0)} \) represents a perturbed input of \( x^{(0)} \), by the bounded perturbation \( \Delta x^{(0)} \). And \( \varepsilon \) will be the bound of output distance \( \Delta y^{(n)} \). Under this encoding, Problem 1 can be formulated as an optimization problem:

\[
\begin{align*}
\varepsilon & := \max |\hat{x}^{(n)} - x^{(n)}|, \\
\text{s.t.} \quad \hat{x}^{(n)} &= F(\hat{x}^{(0)}), x^{(n)} = F(x^{(0)}), \\
\Delta x^{(0)}, \hat{x}^{(0)} &\in X, \|\hat{x}^{(0)} - x^{(0)}\|_\infty < \delta.
\end{align*}
\]

Note that Eq. (1) can be solved with MILP by introducing a binary variable for each ReLU activation [Cheng et al., 2017], but the complexity is too high to be scalable. To overcome this, we present a new interleaving twin-network encoding (ITNE) scheme with two approximation techniques to efficiently find an over-approximated solution \( \varepsilon \geq \varepsilon \).

**2.2 Interleaving Twin-Network Encoding**

In this work, we design the interleaving twin-network encoding (ITNE) as shown in the right side of Fig. 2. Compared with the basic twin-network encoding (BTNE) [Katz et al., 2017] (the left side of Fig. 2), besides the connections between the input and output layers, interleaving connections are added for all hidden neurons between the two network copies. Specifically, for each neuron \( x = \text{relu}(y) \), two variables, \( \Delta y = \hat{y} - y \) and \( \Delta x = \hat{x} - x \), are added to encode the distance of \( y \) and \( x \) between the two copies. These distance variables reflect the hidden neuron variation caused by the perturbation. These changes enable the usage of the over-approximation techniques introduced below.

**2.3 Over-Approximation Techniques**

Leveraging ITNE, we design two over-approximation techniques, network decomposition (ND) and LP relaxation (LPR), to improve the global robustness certification efficiency. Inspired by the local robustness certification work in [Huang et al., 2020b], our ND and LPR techniques are specifically designed for global robustness.

**ITNE-Based Network Decomposition (ND)**

The main idea of ND is to divide a neural network into sub-networks and decompose the entire optimization problem into smaller problems to significantly reduce the optimization complexity. In the ITNE schema, instead of finding the output range of two copies of each sub-network, we look for the range of the original sub-network and the range of the output distance. For a decomposed network \( F_{w_i}(x^{(i)}) \) with input \( x^{(i)} \) and output \( y^{(i)} \), given the input bounds \( \underline{x}^{(i)} \) and \( \overline{x}^{(i)} \), optimization problem in Eq. (1) is formulated on \( F_{w_i}(x^{(i)}) \) to derive output ranges \( \underline{y}^{(i)} \) and \( \overline{y}^{(i)} \).

**ITNE-Based LP Relaxation (LPR)**

The idea of LPR is to relax the ReLU relation \( x = \max(0, y) \), \( y \in [\underline{y}, \overline{y}] \) into linear constraints. When \( y \leq 0 \leq \overline{y} \), ReLU relation can be relaxed by three linear inequations:

\[
\begin{align*}
x &\geq 0, \quad x \geq y, \quad (\overline{y} - y)x \leq \overline{y}(y - y).
\end{align*}
\]

In this work, we relax LPRs in the original network \( x^{(n)} = F(x^{(0)}) \) by Eq. (2), and relax the ReLU distance \( \Delta x = \text{relu}(y + \Delta y) - \text{relu}(y) \), as shown in Fig. 3. For \( \forall y \in \mathbb{R} \), the \( (\Delta x, \Delta y) \) mapping always falls in the shadowed area. Given \( \Delta y \in [\underline{\Delta y}, \overline{\Delta y}] \), the relation between \( \Delta x \) and \( \Delta y \) can be bounded by a linear lower and upper bound:

\[
\frac{l(u - \Delta y)}{u - l} \leq \Delta x \leq \frac{u(\Delta y - l)}{u - l},
\]

where \( l = \min(0, \Delta y) \) and \( u = \max(0, \Delta y) \).

---

1 We denote \( \overline{y} = [\underline{y}, \overline{y}] \) as the range of variable \( v \).
2.4 Illustrating Example

We consider the example neural network in Fig. 1, and set the input perturbation bound as \( \delta = 0.1 \) and the input domain as \( x^{(0)} \in [-1, 1]^2 \). The example neural network can be decomposed into three sub-networks:

\[
\begin{align*}
\mathbf{x}_1^{(1)} &= \text{relu}(y_1^{(1)}) = \text{relu}(x_1^{(0)} + 0.5x_2^{(0)}), \\
\mathbf{x}_2^{(1)} &= \text{relu}(y_2^{(1)}) = \text{relu}(-0.5x_1^{(0)} + x_2^{(0)}), \\
\mathbf{x}_2^{(2)} &= \text{relu}(y_2^{(2)}) = \text{relu}(x_1^{(1)} - x_2^{(1)}).
\end{align*}
\]

In Fig. 4, we demonstrate the global robustness certification processes of exact MILP, network decomposition (ND) and LPR relaxation (LPR) for each network encoding schema (BTNE and ITNE).

### Table 1: Neural network setting and experimental results.

<table>
<thead>
<tr>
<th>ID</th>
<th>Neurons</th>
<th>Layers</th>
<th>( t_{out} ) (s)</th>
<th>( \varepsilon )</th>
<th>( \overline{\varepsilon}_{out} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>FC:2</td>
<td>0.1s</td>
<td>0.0583</td>
<td>0.0657</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>Conv:2</td>
<td>0.2s</td>
<td>0.0527</td>
<td>0.0722</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>FC:2</td>
<td>0.8s</td>
<td>0.0496</td>
<td>0.0653</td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td>Conv:2</td>
<td>( \geq 0.0731 )</td>
<td>0.0481</td>
<td>0.0673</td>
</tr>
<tr>
<td>5</td>
<td>64</td>
<td>FC:3</td>
<td>50s</td>
<td>0.0452</td>
<td>0.0731</td>
</tr>
<tr>
<td>6</td>
<td>1416</td>
<td>Conv:1</td>
<td>4.8h</td>
<td>0.347</td>
<td>0.578</td>
</tr>
<tr>
<td>7</td>
<td>3872</td>
<td>FC:2</td>
<td>3.3h</td>
<td>0.453</td>
<td>0.874</td>
</tr>
<tr>
<td>8</td>
<td>5824</td>
<td>FC:2</td>
<td>3.5h</td>
<td>0.519</td>
<td>1.521</td>
</tr>
</tbody>
</table>

2.5 Efficient Over-Approximation Algorithm

Finally, our global robustness certification algorithm is designed by combining the ITNE-based ND and LPR techniques with Selective Refinement (SR). For layer \( i \), ND constructs sub-networks \( F_k(x_i^{(j)}) \), \( \forall j \). MILP problems are built by LPR and SR to get the output variation bounds of layer \( i \). The over-approximated output variation bound \( \overline{\varepsilon} \) is derived by iteratively evaluating the output bounds (both the \( \Delta y_i \) and \( \Delta x_i \)) of each hidden layer.

### 3 Evaluation and Applications

We first evaluate our algorithm on various DNNs and compare its results with exact global robustness (when available) and an under-approximated global robustness. Then, we demonstrate the application of our approach in a case study of safety verification for a vision-based robotic control system, and show the importance of efficient global robustness certification for safety-critical systems that involve neural nets. Finally, we discuss its comparison with adversarial training.

### 3.1 Performance Evaluation

We compare our approach with other methods on a set of DNNs, as shown in Table 1. DNNs 1 to 5 are 3-layer fully-connected (FC) networks trained on the Auto MPG dataset [Quinlan, 1993]. DNNs 6 to 8 are convolutional networks trained on the MNIST dataset [Lecun et al., 1998]. The input perturbation bound \( \delta = 0.001 \) for DNNs 1 to 5 and \( \delta = 2/255 \) for DNNs 6 to 8.

We compare our over-approximated output variation bound \( \overline{\varepsilon}_{out} \) with the exact bound \( \varepsilon \) solved by Reluplex [Katz et al., 2017; Katz et al., 2019] and the MILP encoding in Eq. (1).

--

While LPR can remove all integer variables in the MILP formulation to reduce the complexity, such extreme over-approximation may be too inaccurate. Thus, we try to selectively refine a limited number of neurons, by not relaxing their ReLU relations. This is similar to the layer-level refinement idea in [Huang et al., 2020b], but with a focus on global robustness.

Due to limited space, we only present 2 outputs of MNIST (out of 10) in Table 1. The rest show similar trends.

More detailed experiment settings can be found in the full paper [Wang et al., 2022b].
The runtime of Reluplex $t_R$ and MILP $t_M$ quickly increases with respect to neural network size. None of them can address 64-neuron DNN-5 within 24 hours. From DNNs 1 to 4, whose exact bounds $\varepsilon$ are available, our algorithm can finish in seconds with only about 13% to 40% over-approximation. Starting from DNN 5, there is no other work in the literature that can derive a sound and deterministic global robustness in a reasonable time. To assess our over-approximated results for larger networks, we leverage adversarial examples from the Projected Gradient Descent (PGD) [Madry et al., 2018] among the entire dataset to derive an under-approximated output variation bound $\tilde{\varepsilon}$, inspired by [Ruan et al., 2019]. The experiments of DNNs 6 to 8 demonstrate that our method can provide meaningful over-approximation (less than 3x of the under-approximation) for DNNs with more than 5000 hidden neurons within 5 hours.

In our recent work [Wang et al., 2022c], we further improved the efficiency and tightness of our approach, where we leverage novel symbolic propagation technique inspired by $\beta$-CROWN [Wang et al., 2021a] to replace the MILP solver in this work. The symbolic propagation technique takes advantage of GPU acceleration and can certify DNNs 6 to 8 within 3 hours and reduce the $(\varepsilon_{out} - \varepsilon)$ gap by 9% - 60%.

### 3.2 Case Study on Control Safety Verification

For control systems that use neural networks for perception, a critical and yet challenging question is whether the system can remain safe under perturbation of network inputs. In [Wang et al., 2021d], we formulate this as a design-time safety assurance problem based on global robustness. Here, leveraging our global robustness certification technique, we demonstrate a solution for this safety assurance problem.

In particular, we consider an advanced cruise control (ACC) case study, where an ego vehicle, equipped with a camera, is following a reference vehicle. The captured images may be slightly perturbed. The distance from the reference vehicle is inferred from the images by a DNN. A feedback controller controls the ego vehicle based on the estimated distance. We model this example in the tool Webots [Michel, 2004] (Fig. 5). The ego vehicle is safe if distance $d \in [0.5, 1.9]$ and speed $v_e \in [0.1, 0.7]$. The reference vehicle speed $v_r$ is randomly adjusted within $[0.2, 0.6]$. The camera takes RGB images with resolution $24 \times 48$. A 5-layer convolutional network is trained with 100k pre-captured images. We model the entire system as an LTI system with external disturbance terms.\footnote{External disturbances are caused by the randomness of reference vehicle speed and the inaccuracy of the linear model.} [Wang et al., 2022b], where the system states include distance and vehicle speed. The control input follows the feedback control law $u = K \hat{x}$, where $\hat{x}$ is the estimated system state. According to the invariant set based verification [Huang et al., 2020a; Wang et al., 2020; Wang et al., 2021c], the vehicle control safety can be verified if the distance estimation error $\Delta d$ is within $[-0.14, 0.14]$.

The distance estimation error $\Delta d = \Delta d_1 + \Delta d_2$ contains the DNN model inaccuracy $\Delta d_1$ and the output variation $\Delta d_2$ caused by input perturbation. While $|\Delta d_1| \leq 0.0730$ is the worst-case model inaccuracy among the dataset, $\Delta d_2$ is the focus of this study and can be bounded by our global robustness certification algorithm. Assuming the input perturbation is bounded by $\delta = 2/255$, the certified output variation bound becomes $|\Delta d_2| \leq \tau = 0.0568$. Combined with $\Delta d_1$, we have $|\Delta d| \leq 0.1292$. Therefore, under the assumed perturbation bound, we can assert that the DNN in this ACC system is safe.

This is validated in Webots simulations, where adversarial perturbations are added by the Fast Gradient Sign Method (FGSM) [Goodfellow et al., 2015]. During simulations, when the perturbation bound $\delta = 2/255$, we always have $|\Delta d| \leq 0.14$ and a safe system. If we further increase input perturbation, $|\Delta d| > 0.14$ is observed when $\delta = 5/255$ and unsafe states are observed when $\delta = 10/255$. This shows the impact of input perturbation on system safety and the importance of our global robustness analysis.

### 3.3 Comparison with Adversarial Training

Neural network adversarial attack techniques and adversarial training algorithms are often jointly developed. Adversarial training performance is usually measured under the existing attack techniques, which may be less effective for more advanced attacks in the future. Instead, network robustness is a metric independent of attack techniques and can provide a deterministic guarantee. Compare to local robustness, where the guarantee is only on a finite set of data, global robustness can provide a universal guarantee for all possible inputs. Besides being a reliable metric, global robustness can also provide guidance to improve network robustness [Wang et al., 2022c; Fu et al., 2022] under arbitrary adversarial attacks.

### 4 Conclusion and Future Work

We present an efficient certification algorithm to provide sound and deterministic global robustness analysis for ReLU neural networks. Experiments demonstrate that our approach is much more efficient and scalable than the exact certification approaches while providing tight over-approximation, and a case study further demonstrates its potential for practical systems. We believe that the approach has the potential to be applied in a variety of domains such as autonomous driving [Jiao et al., 2021; Zhu et al., 2021; Wang et al., 2021e] and smart building control [Xu et al., 2021], where disturbances and noises to sensor inputs are common. We will explore these applications in future work. We also plan to develop methods that can improve the global robustness of neural networks during their design and training, and investigate methods for joint design and verification, inspired by our recent works in this area [Wang et al., 2021b; Wang et al., 2022a; Wang et al., 2023a; Wang et al., 2023b].

![Figure 5: (a) Simulated ACC in Webots: The ego car (left) follows the reference car (right); (b) An example camera image.](image-url)
Ethical Statement

There are no ethical issues with the proposed technique.

Acknowledgments

We gratefully acknowledge funding support by National Science Foundation (NSF) grants 1834701, 1724341, 2038853, Department of Energy (DOE) award DE-EE0009150, and Office of Naval Research grant N00014-19-1-2496.

References


6502


6503