LUND UNIVERSITY

Formal Local Implication Between Two Neural Networks

Baninajjar, Anahita; Rezine, Ahmed; Aminifar, Amir

Published in:
European Conference on Artificial Intelligence (ECAI)

2025

Link to publication

Citation for published version (APA):
Baninajjar, A., Rezine, A., & Aminifar, A. (in press). Formal Local Implication Between Two Neural Networks. In *European Conference on Artificial Intelligence (ECAI)*

Total number of authors:
3

General rights
Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: https://creativecommons.org/licenses/

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Download date: 07. Nov. 2025

Formal Local Implication Between Two Neural Networks

Anahita Baninajjar[a,*], **Ahmed Rezine**[b] **and Amir Aminifar**[a]

[a]Department of Electrical and Information Technology, Lund University, Sweden
[b]Department of Computer and Information Science, Linköping University, Sweden

Abstract. Given two neural network classifiers with the same input and output domains, our goal is to compare the two networks in relation to each other over an entire input region (e.g., within a vicinity of an input sample). To this end, we establish the foundation of *formal local implication* between two networks, i.e., $\mathcal{N}_2 \overset{\boldsymbol{D}}{\Longrightarrow} \mathcal{N}_1$, in an *entire input region* $\boldsymbol{D}$. That is, network $\mathcal{N}_1$ consistently makes a correct decision every time network $\mathcal{N}_2$ does, and it does so in an entire input region $\boldsymbol{D}$. We further propose a *sound* formulation for establishing such formally-verified (provably correct) local implications. The proposed formulation is relevant in the context of several application domains, e.g., for comparing a trained network and its corresponding compact (e.g., pruned, quantized, distilled) networks. We evaluate our formulation based on the MNIST, CIFAR10, and two real-world medical datasets, to show its relevance.

1 Introduction

Quantitative comparison of neural networks, e.g., in terms of performance, is a fundamental concept in the Machine Learning (ML) domain. One common example is when a network is pruned, quantized, or distilled to run the compact networks on edge devices or smart sensors. In the medical domain, for instance, neural networks can enable implantable and wearable devices to detect myocardial infarction [26] or epileptic seizures [1] in real time. However, due to their limited computing resources, such devices often adopt the compact networks corresponding to the original medical-grade networks. It is vital for the compact network to reliably detect cardiac abnormalities/seizures, as lack of reliable decisions can jeopardize patients' lives. Therefore, reasoning about the decisions made by the compact network and their relation to the decisions made by the original/reference network is vital for the safe deployment of the compact networks.

In this work, we focus on compatible neural networks, i.e., two neural networks trained for the same learning/classification task, with the same input and output domains, but not the same weights and/or architectures. We define an input region as the region in a vicinity of a given input sample, e.g., captured by the absolute-value/Euclidean/maximum norm centered around the input sample. Given the two networks and an input region, we investigate whether it is possible to prove that, *in the entire input region*, one network ($\mathcal{N}_1$) consistently makes a correct decision every time the other network ($\mathcal{N}_2$) does. That is, $\mathcal{N}_2 \overset{\boldsymbol{D}}{\Longrightarrow} \mathcal{N}_1$, in an entire input region $\boldsymbol{D}$. This is the definition of *implication* that is valid in an entire input region $\boldsymbol{D}$ (i.e., local), hence referred to as *local implication*.

* Corresponding Author. Email: anahita.baninajjar@eit.lth.se.

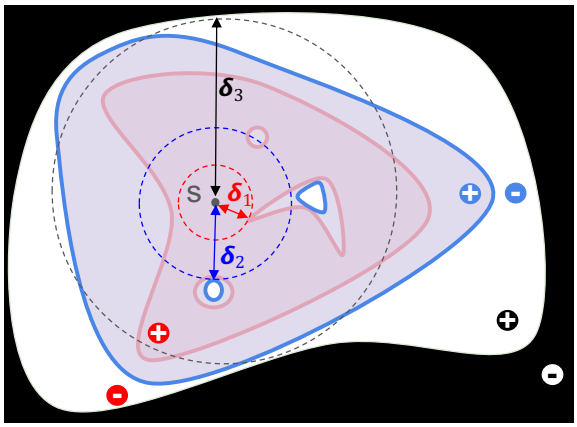


Figure 1: Two binary classifiers (red and blue) learn to capture the black/white classes $\oplus$ and $\ominus$. Formal local implication captures, despite robustness violations of both red and blue classifiers, that the blue classifier makes the right decision each time the red one does within δ_3 perturbations of sample s.

Formal local implication can capture, as illustrated in Figure 1, that despite possible violations of local robustness, one network (the blue classifier in Figure 1) makes the correct decision (with respect to ground truth as captured by the black/white $\oplus$ and $\ominus$ in the figure) each time the other network does (the red classifier in the figure). Tracking the decisions of the two networks separately cannot capture local implication. Instead, outputs of both networks need to be compared for each single input sample throughout the whole considered input region. To this end, we define the notion of Local Relative Output Margin (LROM). LROM not only enables us to reason about the given input samples, but also to reason about the entire region in the vicinity of the samples.

In this paper, we propose a formulation to establish safe (provably-correct) bounds on LROM and formal verification guarantees on the decisions made by the two networks in the entire input region. LROM enables us to formally prove that a network consistently makes a correct decision every time the other network does, and it does so in the entire input region. We evaluate our proposed formulation extensively on several datasets to show its relevance, including two real-world medical applications for detection of cardiac arrhythmia or epileptic seizures. Our main contributions are summarized below:

- We establish the foundation of *formal local implication* between two networks $\mathcal{N}_1$ and $\mathcal{N}_2$, i.e., $\mathcal{N}_2 \overset{\boldsymbol{D}}{\Longrightarrow} \mathcal{N}_1$.
- We propose a *sound* formulation for establishing such formally-verified (provably correct) local implications.

- We conduct extensive experiments to compare the decisions made by pre-trained classifiers and their corresponding pruned, quantized, knowledge-distilled, or verification-friendly counterparts, on the MNIST dataset [18], CIFAR10 dataset [17], CHB-MIT epilepsy dataset [24], and MIT-BIH arrhythmia dataset [10].

2 Formal Definitions and Notations

In this section, we formally describe Deep Neural Network (DNN) classifiers. Moreover, we introduce and formalize the notion of Relative Output Margin (ROM) and its extension to an entire input region, i.e., LROM. Finally, we connect the notion of LROM to formal local implications.

2.1 Deep Neural Networks (DNNs)

In this work, we mainly consider DNN classifiers. A DNN classifier is a nonlinear function $\mathcal{N} : \mathbb{R}^{n_0^{\mathcal{N}}} \rightarrow \mathbb{R}^{n_N^{\mathcal{N}}}$ consisting of a sequence of N layers followed by a softmax layer. Each layer is a linear transformation followed by a nonlinear activation function. Here, $n_k^{\mathcal{N}}$ is the number of neurons in the k^{th} layer of network $\mathcal{N}$. Let $f_k^{\mathcal{N}}(\cdot) : \mathbb{R}^{n_{k-1}^{\mathcal{N}}} \rightarrow \mathbb{R}^{n_k^{\mathcal{N}}}$ be the function that derives values of the k^{th} layer from the output of its preceding layer. The values of the k^{th} layer, denoted by $\boldsymbol{x}^{(k)}$, are given by:

$$\boldsymbol{x}^{(k)} = f_k^{\mathcal{N}}(\boldsymbol{x}^{(k-1)}) = act_k^{\mathcal{N}}(\boldsymbol{W}^{(k)}\boldsymbol{x}^{(k-1)} + \boldsymbol{b}^{(k)}),$$

where $\boldsymbol{W}^{(k)}$ and $\boldsymbol{b}^{(k)}$ capture weights and biases of the k^{th} layer, and $act_k^{\mathcal{N}}$ represents an activation function. The last layer uses softmax as the activation function. For each class c_i in the last layer $N+1$, the softmax function value is: $\boldsymbol{x}_{c_i}^{(N+1)} = \sigma_{c_i}(\boldsymbol{x}^{(N)})$.

2.2 Local Relative Output Margins (LROMs)

We consider two DNNs $\mathcal{N}_1$ with N_1+1 layers with values $\boldsymbol{x}^{(0)}, \ldots \boldsymbol{x}^{(N_1+1)}$ and $\mathcal{N}_2$ with N_2+1 layers with values $\boldsymbol{y}^{(0)}, \ldots \boldsymbol{y}^{(N_2+1)}$. Suppose $n_0^{\mathcal{N}_1} = n_0^{\mathcal{N}_2}$ and $n_{N_1}^{\mathcal{N}_1} = n_{N_2}^{\mathcal{N}_2}$. Such networks are said to be compatible as their inputs and outputs have the same dimensions.

Let us now introduce the notions of Output Margin (OM) and of Relative Output Margin (ROM).

Definition 1. *Output Margin (OM) $\pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_1}(c_i, c_j)$ of classes (c_i, c_j) for DNN $\mathcal{N}_1$ and input $\boldsymbol{x}^{(0)}$ is the ratio $\pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_1}(c_i, c_j) = \frac{\sigma_{c_i}(\boldsymbol{x}^{(N_1)})}{\sigma_{c_j}(\boldsymbol{x}^{(N_1)})}$ of the outcome being c_i by the one of being c_j.*

Recall classifiers decide on the class with a maximum softmax value. Let us consider binary classification for simplicity. Assuming the predicted class is c_i, we know $\sigma_{c_i}(\boldsymbol{x}^{(N_1)}) \geq \sigma_{c_j}(\boldsymbol{x}^{(N_1)})$ and, in turn, $\pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_1}(c_i, c_j) = \frac{\sigma_{c_i}(\boldsymbol{x}^{(N_1)})}{\sigma_{c_j}(\boldsymbol{x}^{(N_1)})} \geq 1$.

Definition 2. *Relative Output Margin (ROM) $\Pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_1|\mathcal{N}_2}(c_i, c_j)$ of class pair (c_i, c_j) for $\mathcal{N}_1$ w.r.t. compatible $\mathcal{N}_2$, and for common input $\boldsymbol{x}^{(0)} = \boldsymbol{y}^{(0)}$, is the quotient of the Output Margin (OM) in $\mathcal{N}_1$ by the one in $\mathcal{N}_2$:*

$$\Pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_1|\mathcal{N}_2}(c_i, c_j) = \frac{\pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_1}(c_i, c_j)}{\pi_{\boldsymbol{y}^{(0)}}^{\mathcal{N}_2}(c_i, c_j)} = \frac{\sigma_{c_i}(\boldsymbol{x}^{(N_1)}) \cdot \sigma_{c_j}(\boldsymbol{y}^{(N_2)})}{\sigma_{c_j}(\boldsymbol{x}^{(N_1)}) \cdot \sigma_{c_i}(\boldsymbol{y}^{(N_2)})}.$$

We use $\Pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_1|\mathcal{N}_2}(c_i, c_j)$ to compare output margins between c_i and c_j in DNNs $\mathcal{N}_1$ and $\mathcal{N}_2$ for the same input $\boldsymbol{x}^{(0)}$.

To explore formal local implication, we establish in this paper bounds on ROM values in entire input regions, e.g., in the vicinity of an input $\tilde{\boldsymbol{x}}^{(0)}$ or in a δ-neighborhood of an input $\tilde{\boldsymbol{x}}^{(0)}$, defined as $\boldsymbol{D}_{\tilde{\boldsymbol{x}}^{(0)}}^{\delta} = \left\{ \boldsymbol{x}^{(0)} \text{ s.t. } \|\boldsymbol{x}^{(0)} - \tilde{\boldsymbol{x}}^{(0)}\|_\infty \leq \delta \right\}$.

Definition 3. Local Relative Output Margin (LROM) *of classes (c_i, c_j) for $\mathcal{N}_1$ w.r.t. its compatible network $\mathcal{N}_2$ in $\boldsymbol{D}_{\tilde{\boldsymbol{x}}^{(0)}}^{\delta}$ is the set $\left\{ \Pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_1|\mathcal{N}_2}(c_i, c_j) \mid \boldsymbol{x}^{(0)} \in \boldsymbol{D}_{\tilde{\boldsymbol{x}}^{(0)}}^{\delta} \right\}$.*

Remark 1. *If $min\left\{ \Pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_1|\mathcal{N}_2}(c_i, c_j) \mid \boldsymbol{x}^{(0)} \in \boldsymbol{D}_{\tilde{\boldsymbol{x}}^{(0)}}^{\delta} \right\} \geq 1$, then $\pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_1}(c_i, c_j) \geq \pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_2}(c_i, c_j)$, for all $\boldsymbol{x}^{(0)}$ in the entire input region $\boldsymbol{D}_{\tilde{\boldsymbol{x}}^{(0)}}^{\delta}$. If $\mathcal{N}_2$ makes a correct decision, i.e., $\pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_2}(c_i, c_j) = \frac{\sigma_{c_i}(\boldsymbol{x}^{(N_2)})}{\sigma_{c_j}(\boldsymbol{x}^{(N_2)})} \geq 1$, then $\pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_1}(c_i, c_j) = \frac{\sigma_{c_i}(\boldsymbol{x}^{(N_1)})}{\sigma_{c_j}(\boldsymbol{x}^{(N_1)})} \geq 1$, assuming the predicted class is c_i. This, in turn, means that, in the entire input region $\boldsymbol{D}_{\tilde{\boldsymbol{x}}^{(0)}}^{\delta}$, $\mathcal{N}_1$ will make a correct decision every time $\mathcal{N}_2$ does, i.e., $\mathcal{N}_2 \overset{\boldsymbol{D}_{\tilde{\boldsymbol{x}}^{(0)}}^{\delta}}{\Longrightarrow} \mathcal{N}_1$. We say that $\mathcal{N}_2$ implies $\mathcal{N}_1$ on $\boldsymbol{D}_{\tilde{\boldsymbol{x}}^{(0)}}^{\delta}$.*

3 Method

In this section, we introduce an optimization problem to bound LROMs for two compatible DNNs and establish formal local implication between two networks. We also describe how we introduce and handle an over-approximation of the two networks in order to soundly solve the optimization problem and derive a (provably-correct) verified bound.

Assume two compatible DNNs $\mathcal{N}_1$ and $\mathcal{N}_2$ with respectively N_1+1 and N_2+1 layers, a common input $\tilde{\boldsymbol{x}}^{(0)}$ in the domain $\boldsymbol{D}$ of $\mathcal{N}_1$ and $\mathcal{N}_2$, and a perturbation bound δ. Our goal is to find, for any class pair (c_i, c_j), a tight lower bound for $min\left\{ \Pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_1|\mathcal{N}_2}(c_i, c_j) \mid \boldsymbol{x}^{(0)} \in \boldsymbol{D}_{\tilde{\boldsymbol{x}}^{(0)}}^{\delta} \right\}$ and a tight upper bound for $max\left\{ \Pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_1|\mathcal{N}_2}(c_i, c_j) \mid \boldsymbol{x}^{(0)} \in \boldsymbol{D}_{\tilde{\boldsymbol{x}}^{(0)}}^{\delta} \right\}$.

The above optimization problem involves the softmax function. Therefore, to solve this optimization problem, we look into $\ln\left(\Pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_1|\mathcal{N}_2}(c_i, c_j)\right)$ and observe it coincides with $(\boldsymbol{x}_{c_i}^{(N_1)} - \boldsymbol{x}_{c_j}^{(N_1)}) - (\boldsymbol{y}_{c_i}^{(N_2)} - \boldsymbol{y}_{c_j}^{(N_2)})$. Hence, we can characterize LROM bounds by reasoning on inputs to the softmax layers (i.e., networks' logits). As a result, our optimization objective is simplified to:

$$\ln\left(min\left\{ \Pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_1|\mathcal{N}_2}(c_i, c_j) \mid \boldsymbol{x}^{(0)} \in \boldsymbol{D}_{\tilde{\boldsymbol{x}}^{(0)}}^{\delta} \right\}\right) =$$
$$\min_{\boldsymbol{x}^{(0)} = \boldsymbol{y}^{(0)} \in \boldsymbol{D}_{\tilde{\boldsymbol{x}}^{(0)}}^{\delta}} \left((\boldsymbol{x}_{c_i}^{(N_1)} - \boldsymbol{x}_{c_j}^{(N_1)}) - (\boldsymbol{y}_{c_i}^{(N_2)} - \boldsymbol{y}_{c_j}^{(N_2)}) \right),$$
$$\ln\left(max\left\{ \Pi_{\boldsymbol{x}^{(0)}}^{\mathcal{N}_1|\mathcal{N}_2}(c_i, c_j) \mid \boldsymbol{x}^{(0)} \in \boldsymbol{D}_{\tilde{\boldsymbol{x}}^{(0)}}^{\delta} \right\}\right) =$$
$$\max_{\boldsymbol{x}^{(0)} = \boldsymbol{y}^{(0)} \in \boldsymbol{D}_{\tilde{\boldsymbol{x}}^{(0)}}^{\delta}} \left((\boldsymbol{x}_{c_i}^{(N_1)} - \boldsymbol{x}_{c_j}^{(N_1)}) - (\boldsymbol{y}_{c_i}^{(N_2)} - \boldsymbol{y}_{c_j}^{(N_2)}) \right).$$

Let us now formulate the optimization problem based on the above objective function and the constraints imposed by the neural networks

and input region, as follows:

$$\min_{\boldsymbol{x}^{(0)}} \ (\boldsymbol{x}_{c_i}^{(N_1)} - \boldsymbol{x}_{c_j}^{(N_1)}) - (\boldsymbol{y}_{c_i}^{(N_2)} - \boldsymbol{y}_{c_j}^{(N_2)}), \tag{1}$$

$$\text{s.t.} \quad \boldsymbol{y}^{(0)} = \boldsymbol{x}^{(0)}, \quad \tilde{\boldsymbol{x}}^{(0)} \in \boldsymbol{D}, \tag{2}$$

$$\|\boldsymbol{x}^{(0)} - \tilde{\boldsymbol{x}}^{(0)}\|_\infty = \|\boldsymbol{y}^{(0)} - \tilde{\boldsymbol{x}}^{(0)}\|_\infty \leq \delta, \tag{3}$$

$$\boldsymbol{x}^{(k)} = f_k^{\mathcal{N}_1}(\boldsymbol{x}^{(k-1)}), \ \forall k \in \{1, \ldots, N_1\}, \tag{4}$$

$$\boldsymbol{y}^{(l)} = f_l^{\mathcal{N}_2}(\boldsymbol{y}^{(l-1)}), \ \forall l \in \{1, \ldots, N_2\}. \tag{5}$$

Let $\mathcal{M}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j)$ be the exact value obtained as solution to this problem. Equation (1) introduces the objective function used to capture the (logarithm of the) minimum ROM of the class pair (c_i, c_j) for $\mathcal{N}_1$ w.r.t. $\mathcal{N}_2$ in the input region $\boldsymbol{D}^\delta_{\tilde{\boldsymbol{x}}^{(0)}}$.[1] Note that $\boldsymbol{x}_{c_i}^{(N_1)} - \boldsymbol{x}_{c_j}^{(N_1)}$ captures the difference between the logit values associated to classes c_i and c_j in network $\mathcal{N}_1$. Similarly, $\boldsymbol{y}_{c_i}^{(N_2)} - \boldsymbol{y}_{c_j}^{(N_2)}$ captures the difference between the logit values associated to the same classes in $\mathcal{N}_2$. The objective function is then to minimize the difference between these two quantities.

Let us consider Equations (2)–(5). Equation (2) enforces both that $\boldsymbol{y}^{(0)}$ (the perturbed input to network $\mathcal{N}_2$) equals $\boldsymbol{x}^{(0)}$ (the perturbed input to network $\mathcal{N}_1$), and that the original input $\tilde{\boldsymbol{x}}^{(0)}$ belongs to the dataset $\boldsymbol{D}$ of the two networks. Equation (3) enforces that the perturbed inputs $\boldsymbol{x}^{(0)}$ and $\boldsymbol{y}^{(0)}$ are in the δ-neighborhood of $\tilde{\boldsymbol{x}}^{(0)}$. Equation (4) characterizes values of the first N_1 layers of network $\mathcal{N}_1$ as it relates the values of the k^{th} layer (for k in $\{1, \ldots, N_1\}$) to those of its preceding layer, using the nonlinear function $f_k^{\mathcal{N}_1} : \mathbb{R}^{n_{k-1}^{\mathcal{N}_1}} \rightarrow \mathbb{R}^{n_k^{\mathcal{N}_1}}$. The same is applied to network $\mathcal{N}_2$ using the nonlinear functions $f_l^{\mathcal{N}_2} : \mathbb{R}^{n_{l-1}^{\mathcal{N}_2}} \rightarrow \mathbb{R}^{n_l^{\mathcal{N}_2}}$ for each layer l as captured in Equation (5).

Equations (4) and (5) involve activation functions, hence result in nonlinear constraints. As such, finding the exact global optimal solution is intractable. To address this, we consider a sound over-approximation of nonlinearities. In particular, we consider Rectified Linear Unit (ReLU) activation function and adopt existing relaxations for it [8, 25, 2], to over-approximate the values computed at each layer using linear inequalities (see the appendix of [5]).

These over-approximations result in a relaxed optimization problem that can be solved using Linear Programming (LP). The solution of the relaxed optimization problem is denoted by $\mathcal{R}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j)$. Because the relaxed optimization over-approximates the exact one in Equations (1)–(5) and that we are able to find the optimal solution to the LP relaxed formulation, any lower bound obtained for the relaxed problem is guaranteed to be smaller than a solution for the original minimization problem, i.e., $\mathcal{R}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j) \leq \mathcal{M}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j)$.

Theorem 1. *Let (c_i, c_j) be a pair of classes of compatible DNNs $\mathcal{N}_1$ and $\mathcal{N}_2$. Assume a neighborhood $\boldsymbol{D}^\delta_{\tilde{\boldsymbol{x}}^{(0)}}$ and let $\mathcal{R}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j)$ (resp. $\mathcal{R}^{\mathcal{N}_2|\mathcal{N}_1}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j)$) be a solution to the relaxed minimization problem corresponding to LROM of $\mathcal{N}_1$ w.r.t. $\mathcal{N}_2$ (resp. $\mathcal{N}_2$ w.r.t. $\mathcal{N}_1$). Then, we have:*

$$\mathcal{R}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j) \leq \mathcal{M}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j) =$$
$$\ln\left(min\left\{\Pi^{\mathcal{N}_1|\mathcal{N}_2}_{\boldsymbol{x}^{(0)}}(c_i, c_j) \mid \boldsymbol{x}^{(0)} \in \boldsymbol{D}^\delta_{\tilde{\boldsymbol{x}}^{(0)}}\right\}\right) \leq$$
$$\ln\left(max\left\{\Pi^{\mathcal{N}_1|\mathcal{N}_2}_{\boldsymbol{x}^{(0)}}(c_i, c_j) \mid \boldsymbol{x}^{(0)} \in \boldsymbol{D}^\delta_{\tilde{\boldsymbol{x}}^{(0)}}\right\}\right) =$$
$$-\ln\left(min\left\{\Pi^{\mathcal{N}_2|\mathcal{N}_1}_{\boldsymbol{x}^{(0)}}(c_i, c_j) \mid \boldsymbol{x}^{(0)} \in \boldsymbol{D}^\delta_{\tilde{\boldsymbol{x}}^{(0)}}\right\}\right) =$$
$$-\mathcal{M}^{\mathcal{N}_2|\mathcal{N}_1}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j) \leq -\mathcal{R}^{\mathcal{N}_2|\mathcal{N}_1}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j).$$

Based on Theorem 1, the solutions to relaxed optimization problems provide safe lower/upper bounds for LROMs, i.e., $\mathcal{R}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j) \leq \mathcal{M}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j) \leq -\mathcal{R}^{\mathcal{N}_2|\mathcal{N}_1}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j)$.

Corollary 1. *Let (c_i, c_j) be a pair of classes of compatible DNNs $\mathcal{N}_1$ and $\mathcal{N}_2$. Assume a neighborhood $\boldsymbol{D}^\delta_{\tilde{\boldsymbol{x}}^{(0)}}$ and let $\mathcal{R}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j)$ be a solution to the relaxed minimization problem corresponding to LROM of $\mathcal{N}_1$ w.r.t. $\mathcal{N}_2$. If $\mathcal{R}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j) > 0$, then for all perturbed inputs $\boldsymbol{x}^{(0)} \in \boldsymbol{D}^\delta_{\tilde{\boldsymbol{x}}^{(0)}}$ for which $\mathcal{N}_2$ correctly classifies $\boldsymbol{x}^{(0)}$, then $\mathcal{N}_1$ also correctly classifies $\boldsymbol{x}^{(0)}$. That is, $\mathcal{N}_2$ implies $\mathcal{N}_1$ on $\boldsymbol{D}^\delta_{\tilde{\boldsymbol{x}}^{(0)}}$, i.e., $\mathcal{N}_2 \overset{\boldsymbol{D}^\delta_{\tilde{\boldsymbol{x}}^{(0)}}}{\Longrightarrow} \mathcal{N}_1$.*

Joint vs Independent Analysis Our original optimization problem and its linear relaxation compute ROMs bounds for a common input across both networks, ranging over the considered neighborhood. An alternative approach is to reason based on independently obtained ranges of Output Margins (OMs) for each network. However, while independently computing and combining the ranges of OMs for each network leads to sound approximations of ROMs, it results in a significant loss of precision, as it does not consider a common input to both networks. This is formalized by the theorem below, and is witness by our experiments where we evaluate the corresponding loss in precision.

Theorem 2. *Let $\mathcal{M}^{\mathcal{N}_1|\mathbf{0}}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j)$ denote the value of the objective function in Equation (1), i.e., $(\boldsymbol{x}_{c_i}^{(N_1)} - \boldsymbol{x}_{c_j}^{(N_1)}) - (\boldsymbol{y}_{c_i}^{(N_2)} - \boldsymbol{y}_{c_j}^{(N_2)})$, under the choice of a constant second network $\mathcal{N}_2$ that assigns uniform probabilities to all outcomes. In this case, the second term becomes zero, and the expression simplifies to $\boldsymbol{x}_{c_i}^{(N_1)} - \boldsymbol{x}_{c_j}^{(N_1)}$, corresponding to computing minimum OMs for $\mathcal{N}_1$ on its own. Similarly, $\mathcal{M}^{\mathcal{N}_2|\mathbf{0}}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j)$ equals $\boldsymbol{y}_{c_i}^{(N_2)} - \boldsymbol{y}_{c_j}^{(N_2)}$, and $\mathcal{M}^{\mathbf{0}|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j)$ corresponds to its negation. This leads to the following inequality, expressing that the sum of the independently obtained OMs is less than or equal to the ROM value when both networks are considered together:*

$$\mathcal{M}^{\mathcal{N}_1|\mathbf{0}}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j) + \mathcal{M}^{\mathbf{0}|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j) \leq \mathcal{M}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j).$$

Transitivity Property of LROMs Here, we show that LROMs has the transitivity property, which can extend the results to more than two compatible networks.

Theorem 3. *Let (c_i, c_j) be a pair of classes of compatible DNNs $\mathcal{N}_1$, $\mathcal{N}_2$, and $\mathcal{N}_3$. Assume a neighborhood $\boldsymbol{D}^\delta_{\tilde{\boldsymbol{x}}^{(0)}}$ and let $\mathcal{R}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j)$ be a solution to the relaxed minimization problem corresponding to LROM of $\mathcal{N}_1$ w.r.t. $\mathcal{N}_2$. Similarly, let $\mathcal{R}^{\mathcal{N}_2|\mathcal{N}_3}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j)$ be a solution to the relaxed minimization problem corresponding to LROM of $\mathcal{N}_2$ w.r.t. $\mathcal{N}_3$. If we know $\mathcal{R}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j) > 0$ and $\mathcal{R}^{\mathcal{N}_2|\mathcal{N}_3}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j) > 0$, then we can conclude $\mathcal{R}^{\mathcal{N}_1|\mathcal{N}_3}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j) > 0$.*

[1] The objective function captures a stronger condition than mere implication, as it ensures that the margin does not shrink. While our framework is designed for margin preservation, it can be easily adapted to verify only implication by changing the objective to $\boldsymbol{x}_{c_i}^{(N_1)} - \boldsymbol{x}_{c_j}^{(N_1)}$ and moving the second part of the original objective into the constraints as $\boldsymbol{y}_{c_i}^{(N_2)} - \boldsymbol{y}_{c_j}^{(N_2)} > 0$.

Proof. Proof sketches of Theorems 1–3, as well as Corollary 1, are presented in the appendix of [5]. □

4 Evaluation

We evaluate our proposed formulation for formal local implication and investigate the ranges of LROMs across various datasets and DNN structures.[2] Experiments are executed on a MacBook Pro with an 8-core CPU and 32 GB of RAM using the Gurobi solver [12].

4.1 Datasets

We use the following datasets for evaluation:

MNIST dataset [18] contains grayscale handwritten digits. Each digit is depicted through a 28×28 pixel image. We consider the first 100 images of the test set, similar to [28] and [25].

CIFAR10 dataset [17] comprises 32×32 colored images categorized into 10 different classes. In alignment with [28] and [25], we focus on the first 100 images from the test set.

CHB-MIT Scalp EEG database [24] includes 23 individuals diagnosed with epileptic seizures. These recordings are sampled in the international 10–20 EEG system, and our focus is on F7-T7 and F8-T8 electrode pairs, commonly used in seizure detection [27].

MIT-BIH Arrhythmia database [10] involves 48 individuals with 2-channel ECG signals. To establish a classification problem, we consider a subset of 14 cardiac patients who demonstrated at least two different types of heartbeats.

4.2 Neural Networks

In this section, we describe the neural network architecture for each dataset. Detailed information, e.g., accuracy of these DNNs, is available in the appendix of [5].

4.2.1 Original Networks

For the MNIST and CIFAR10 datasets, we use fully-connected DNNs from [28], all of which have undergone robust training as outlined in [6]. We also employ convolutional DNNs described in [28], with results provided in the appendix of [5]. For the CHB-MIT dataset, the personalized DNN for each patient has 2048 input neurons, convolution layers followed by max-pooling, and ends with a dense layer [3]. For the MIT-BIH dataset, the DNN has 320 input neurons, a convolution layer, and a dense layer [3].

4.2.2 Compact Networks

Our experiments involve various techniques to derive compact DNNs. These techniques derive compact DNNs enabling energy-efficient inference on limited resources.

Pruned Networks are created through a pruning procedure applied to DNNs, where certain weights and biases are selectively nullified. For the MNIST and CIFAR10 datasets, we use pruned networks generated by [28] via post-training pruning. Each pruned network removes the smallest weights/biases in each layer, a process called Magnitude-Based Pruning (MBP), resulting in nine pruned networks with pruning rates ranging from 10% to 90%. For the CHB-MIT and MIT-BIH datasets, we apply MBP pruning by setting values below 10% of the maximum weight/bias to zero.

[2] The code is available [4].

Verification-friendly Neural Networks (VNNs) are generated by optimizing weights and biases to maintain their functionality while reducing the number of non-zero weights, as described in [3], and are subsequently used for all networks.

Quantized Networks are obtained by reducing the precision of network weights, converting them from 32-bit floating-point to lower precision. The MNIST and CIFAR10 networks are quantized using post-training float16, int16, int8, and int4 methods provided by [28]. The same quantization methods are applied to the DNNs trained on CHB-MIT and MIT-BIH datasets.

Distilled Networks are networks trained via knowledge distillation to mimic teacher networks' behavior. We consider nine temperatures, i.e., $T = 1, \ldots, 9$, and produce distilled networks for all datasets following [13].

4.3 Results and Analysis

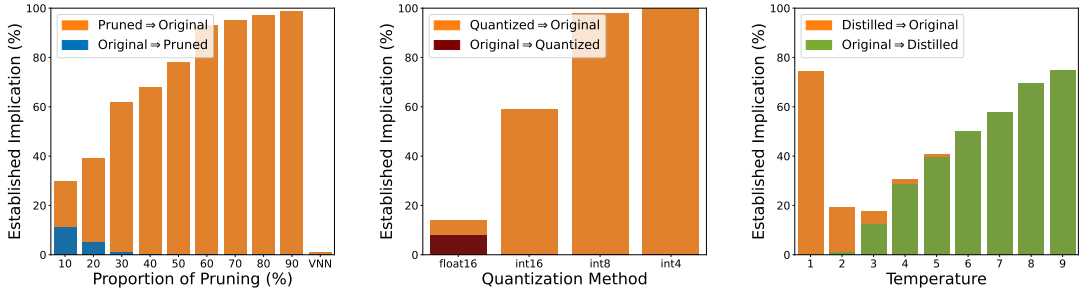
We conduct several experiments with our proposed method for assessing local implications by establishing bounds on LROMs. We exclusively focus on correctly classified samples within each test set. We consider the widths and depths of the networks when defining perturbations. We use $\delta = 0.001$ and $\delta = 0.01$ for the MNIST and CIFAR10 datasets and experiment with several values for the CHB-MIT dataset (δ up to 0.002) and the MIT-BIH dataset (δ up to 0.4). We say we establish local implication of $\mathcal{N}_1$ w.r.t. $\mathcal{N}_2$ on sample vicinity $\boldsymbol{D}^{\delta}_{\tilde{\boldsymbol{x}}^{(0)}}$ (i.e., $\mathcal{N}_2 \overset{\boldsymbol{D}^{\delta}_{\tilde{\boldsymbol{x}}^{(0)}}}{\Longrightarrow} \mathcal{N}_1$), if we show $\mathcal{R}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c, c_j) \geq 0$ for all pairs (c, c_j), where c is the correct class. We omit samples and vicinities when they are clear from the context and write $\mathcal{N}_2 \implies \mathcal{N}_1$ for short. Observe that $\mathcal{R}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c, c_j) \geq 0$ implies $\mathcal{M}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c, c_j) \geq 0$, since $\mathcal{R}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j) \leq \mathcal{M}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c_i, c_j)$ by soundness of our method. Here, we use zero as a threshold in $\mathcal{R}^{\mathcal{N}_1|\mathcal{N}_2}_{\tilde{\boldsymbol{x}}^{(0)},\delta}(c, c_j) \geq 0$ as it corresponds to checking increases or decreases of OM from one network to the other. However, our approach can easily accommodate other thresholds.

Given two compatible networks $\mathcal{N}_2$ and $\mathcal{N}_1$ to be compared on the δ-vicinities of a set of samples, we state that $\mathcal{N}_1$ has more established implications if there are more samples for which we could show $\mathcal{N}_1$ is implied by $\mathcal{N}_2$ on the corresponding vicinities (i.e., we could establish $\mathcal{N}_2 \implies \mathcal{N}_1$) than those for which we could establish $\mathcal{N}_2$ is implied by $\mathcal{N}_1$ (i.e., $\mathcal{N}_1 \implies \mathcal{N}_2$). In this context, if $\mathcal{N}_1$ has more established implications than $\mathcal{N}_2$, then the number of samples with vicinities where we could establish $\mathcal{N}_1$ made the correct decision each time $\mathcal{N}_2$ did is larger than the number of samples with vicinities where we could establish $\mathcal{N}_2$ made the correct decision each time $\mathcal{N}_1$ did. In other words, we could establish formal local implications for $\mathcal{N}_1$ w.r.t. $\mathcal{N}_2$ more often than we could establish it for $\mathcal{N}_2$ w.r.t. $\mathcal{N}_1$.

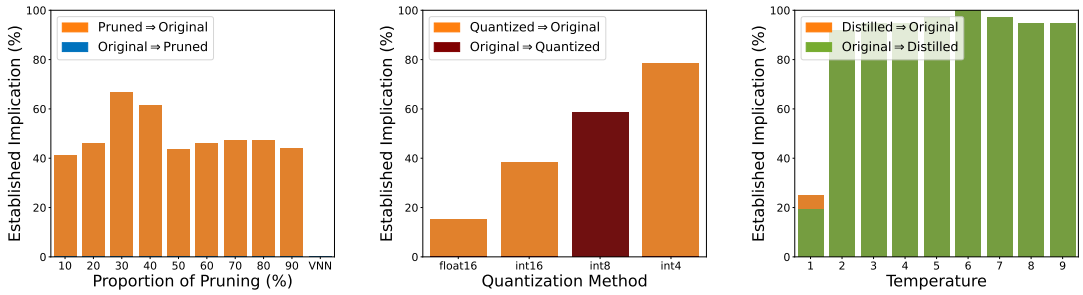
4.3.1 Analysis of Formal Local Implication

Here, we explore formal local implications between original and compact DNNs using LROM.

MNIST Dataset. Figures 2a– 2c present the results of formal local implication for DNNs trained on the MNIST dataset when $\delta = 0.001$. In this figure, the y-axis represents the percentage of samples in the dataset, referred to as established implication, showing the proportion for which the compact network implies the original network on δ-vicinity of considered samples, and the reverse. Compact $\implies$ Original means that whenever the compact network makes a correct decision for a sample, the original network does the same. Figure 2a



(a) MNIST, Pruned [28], [3] (b) MNIST, Quantized [28] (c) MNIST, Distilled [13]



(d) CIFAR10, Pruned [28], [3] (e) CIFAR10, Quantized [28] (f) CIFAR10, Distilled [13]

Figure 2: Stacked bar plots illustrate the established implication of fully-connected DNNs trained on the MNIST and CIFAR10 datasets with $\delta = 0.001$. The y-axis represents the percentage of samples in the dataset, referred to as established implication, showing the proportion for which the Compact network implies the Original network (Compact $\implies$ Original) and vice versa. Compact $\implies$ Original means that whenever the Compact network makes a correct decision for a sample, the Original network does as well.

shows an increase in the established implication as the pruning proportion rises, when investigating how pruned networks imply the behavior of the original networks (Pruned $\implies$ Original). There are two potential explanations for this phenomenon. First, the similarity between the original and less-pruned networks may lead to no network having a higher established implication across the entire perturbation neighborhood. Second, our method may establish implication of more samples in more-pruned networks, due to their sparsity.

The last column of Figure 2a presents the results of investigating the established implication of the VNN generated using [3]. The results show that the VNN is comparable to that of the original network, as both the implication of the VNN by the original network and vice versa are close to zero.

Established implications of quantized networks are depicted in Figure 2b, with quantization precision on the x-axis. Results show original networks are more likely to be implied by the quantized ones than vice versa; i.e., there are more cases where we could establish Quantized $\implies$ Original than cases where we could establish Original $\implies$ Quantized.

Figure 2c illustrates established implication of distilled and original networks implied by each other. The x-axis denotes the temperature of the distilled network, and the y-axis indicates the established implication. This figure demonstrates that the likelihood of distilled networks being implied by original networks increases as temperatures rise, rather than the other way around. The patterns of established implication exhibited by distilled networks differentiate them from pruned and quantized networks, making them a favorable option for creating compact and energy-efficient networks.

The processing time for each sample in the dataset depends on the perturbation, i.e., the value of δ, and the architecture of the original and compact networks. The processing time ($\mu \pm \sigma$) is 15.0 ± 0.7 seconds when $\delta = 0.001$ and 18.3 ± 5.2 seconds when $\delta = 0.01$ for pruned and quantized networks. The processing time for distilled networks is 6.7 ± 0.1 seconds for $\delta = 0.001$ and 6.9 ± 0.2 seconds for $\delta = 0.01$.

CIFAR10 Dataset. Figures 2d– 2f show the results of investigating formal local implication of DNNs trained on the CIFAR10 dataset when $\delta = 0.001$. Although the general patterns in the results of the CIFAR10 DNNs are similar to those of the MNIST DNNs, a few differences are observed. In Figure 2e, the quantized network with int8 precision is more likely to be implied by the original network rather than the reverse. This suggests that the reduction in precision does not significantly impair the network's ability to make the correct decision each time the original one does. Moreover, in Figure 2f, distilled networks consistently exhibit a higher established implication across different temperatures. This behavior indicates that, in this set of experiments, despite fewer parameters, distilled networks tend to better make the correct decision each time the original network does than those obtained with other compaction schemes.

The processing time of DNNs trained on the CIFAR10 dataset is higher than that of the MNIST DNNs, as the number of parameters is larger due to the input size. The processing time ($\mu \pm \sigma$) is 33.1 ± 2.1 seconds when $\delta = 0.001$ and 43.9 ± 8.1 seconds when $\delta = 0.01$ for pruned and quantized networks. The processing time of distilled networks is 15.0 ± 0.9 and 18.1 ± 2.6 seconds for $\delta = 0.001$ and $\delta = 0.01$, respectively.

Table 1: The minimum, maximum, and range ($\mu \pm \sigma$) of LROMs for scenarios where implications of Original networks by Compact ones are investigated using independent and joint analyses at $\delta = 0.001$ and $\delta = 0.01$ on MNIST and CIFAR10. Our joint analysis results in a tighter range, with improvement calculated as $\left(1 - \frac{\text{Range}_{\text{joint}}}{\text{Range}_{\text{ind.}}}\right) \times 100$, indicating the percentage reduction in range.

			MNIST [18]				CIFAR10 [17]			
			Pruned	Quantized	Distilled	VNN	Pruned	Quantized	Distilled	VNN
$\delta = 0.001$	Min (↑)	Ind.	$1.424_{\pm 1.519}$	$-0.485_{\pm 0.296}$	$-19.456_{\pm 23.286}$	$-0.460_{\pm 0.322}$	$0.000_{\pm 0.043}$	$-0.026_{\pm 0.028}$	$-5.927_{\pm 5.049}$	$0.120_{\pm 0.301}$
		Joint	$\mathbf{1.875}_{\pm 1.680}$	$\mathbf{0.010}_{\pm 0.016}$	$\mathbf{-19.237}_{\pm 23.317}$	$\mathbf{-0.167}_{\pm 0.191}$	$\mathbf{0.025}_{\pm 0.046}$	$\mathbf{0.001}_{\pm 0.002}$	$\mathbf{-5.907}_{\pm 5.040}$	$\mathbf{0.128}_{\pm 0.304}$
	Max (↓)	Ind.	$2.391_{\pm 1.898}$	$0.517_{\pm 0.314}$	$-18.118_{\pm 23.399}$	$0.469_{\pm 0.330}$	$0.055_{\pm 0.065}$	$0.029_{\pm 0.030}$	$-5.471_{\pm 4.849}$	$0.162_{\pm 0.315}$
		Joint	$\mathbf{1.940}_{\pm 1.706}$	$\mathbf{0.023}_{\pm 0.019}$	$\mathbf{-18.338}_{\pm 23.367}$	$\mathbf{0.176}_{\pm 0.197}$	$\mathbf{0.030}_{\pm 0.049}$	$\mathbf{0.002}_{\pm 0.002}$	$\mathbf{-5.491}_{\pm 4.858}$	$\mathbf{0.154}_{\pm 0.311}$
	Range (↓)	Ind.	$0.967_{\pm 2.431}$	$1.002_{\pm 0.432}$	$1.338_{\pm 33.011}$	$0.929_{\pm 0.461}$	$0.055_{\pm 0.078}$	$0.055_{\pm 0.041}$	$0.456_{\pm 7.000}$	$0.042_{\pm 0.436}$
		Joint	$\mathbf{0.065}_{\pm 2.394}$	$\mathbf{0.013}_{\pm 0.025}$	$\mathbf{0.899}_{\pm 33.011}$	$\mathbf{0.343}_{\pm 0.274}$	$\mathbf{0.005}_{\pm 0.067}$	$\mathbf{0.001}_{\pm 0.003}$	$\mathbf{0.416}_{\pm 7.000}$	$\mathbf{0.026}_{\pm 0.435}$
	Improvement (↑)		93.3%	98.7%	32.8%	63.1%	90.9%	98.2%	8.8%	38.1%
$\delta = 0.01$	Min (↑)	Ind.	$-3.141_{\pm 2.357}$	$-5.165_{\pm 3.112}$	$-25.995_{\pm 22.597}$	$-4.828_{\pm 2.923}$	$-0.257_{\pm 0.268}$	$-0.287_{\pm 0.294}$	$-8.088_{\pm 6.097}$	$-0.079_{\pm 0.305}$
		Joint	$\mathbf{1.058}_{\pm 1.547}$	$\mathbf{-0.649}_{\pm 0.595}$	$\mathbf{-23.917}_{\pm 22.870}$	$\mathbf{-2.113}_{\pm 1.322}$	$\mathbf{-0.044}_{\pm 0.086}$	$\mathbf{-0.070}_{\pm 0.088}$	$\mathbf{-7.906}_{\pm 5.991}$	$\mathbf{-0.011}_{\pm 0.293}$
	Max (↓)	Ind.	$6.836_{\pm 4.243}$	$5.197_{\pm 3.129}$	$-12.368_{\pm 23.812}$	$4.662_{\pm 2.815}$	$0.314_{\pm 0.321}$	$0.290_{\pm 0.296}$	$-2.966_{\pm 4.091}$	$0.351_{\pm 0.433}$
		Joint	$\mathbf{2.635}_{\pm 1.941}$	$\mathbf{0.681}_{\pm 0.600}$	$\mathbf{-14.496}_{\pm 23.462}$	$\mathbf{1.935}_{\pm 1.211}$	$\mathbf{0.102}_{\pm 0.115}$	$\mathbf{0.073}_{\pm 0.089}$	$\mathbf{-3.144}_{\pm 4.145}$	$\mathbf{0.286}_{\pm 0.381}$
	Range (↓)	Ind.	$9.977_{\pm 4.854}$	$10.362_{\pm 4.413}$	$13.627_{\pm 32.827}$	$9.490_{\pm 4.058}$	$0.571_{\pm 0.418}$	$0.577_{\pm 0.417}$	$5.122_{\pm 7.342}$	$0.430_{\pm 0.530}$
		Joint	$\mathbf{1.577}_{\pm 2.482}$	$\mathbf{1.330}_{\pm 0.845}$	$\mathbf{9.421}_{\pm 32.764}$	$\mathbf{4.048}_{\pm 1.793}$	$\mathbf{0.146}_{\pm 0.144}$	$\mathbf{0.143}_{\pm 0.125}$	$\mathbf{4.762}_{\pm 7.285}$	$\mathbf{0.297}_{\pm 0.481}$
	Improvement (↑)		84.2%	87.2%	30.9%	57.3%	74.4%	75.2%	7.0%	30.9%

4.3.2 *Comparison with Independent Analysis*

Table 1 shows the minimum, maximum and range ($\mu \pm \sigma$) of LROMs for the scenario where we assess if compact networks imply original ones using independent and joint analyses at $\delta = 0.001$ and $\delta = 0.01$ for MNIST and CIFAR10 datasets. Results show that our proposed joint analysis consistently produces higher minimum values and lower maximum values, resulting in a tighter range for LROMs. This occurs because considering two networks in the same setting, as in joint analysis, removes unrealistic scenarios that would not occur in reality, thus preventing a minimum lower than the true minimum and a maximum higher than the true maximum.

Here, range refers to the difference between the minimum and maximum values of each method (Max − Min), where the range calculated by our joint analysis method is consistently lower than that of the independent analysis. Improvement is calculated using $\left(1 - \frac{\text{Range}_{\text{joint}}}{\text{Range}_{\text{ind.}}}\right) \times 100$, which measures the percentage reduction in range from the independent to the joint analysis. For example, for MNIST with $\delta = 0.01$, we have 9.977 for the independent and 1.577 as the joint value, the improvement is $\left(1 - \frac{1.577}{9.977}\right) \times 100 \approx 84.2\%$. This formula standardizes the measurement of relative improvement, indicating that the range of LROMs in the joint analysis is 84.2% narrower than in the independent analysis. This demonstrates that the joint analysis provides a tighter, more consistent range, highlighting its improved precision and reliability.

4.3.3 *Adversarially-Trained Models*

As discussed earlier, our proposed formulation is applicable to any two neural networks. In this section, we analyze three neural networks, two of which are adversarially-trained models designed to defend against adversarial attacks, specifically Projected Gradient Descent (PGD), using different values of ϵ [25]. The PGD-trained networks have ϵ values of 0.1 and 0.3, denoted as PGD1 and PGD3, respectively.

Table 2 presents the results of examining the established implications and certified accuracy of the original (non-defended), PGD1, and PGD3 DNNs. In the established implication section, the column $\Longrightarrow$ row indicates, for example, that in 42% of the samples of the dataset, PGD3 makes a correct decision whenever PGD1 does. The results demonstrate that PGD1 and PGD3 consistently achieve higher established implications than their original counterpart.

These results become even more intriguing when compared to the outcomes of evaluating the robustness of the neural networks using formal verification techniques. For $\delta = 0.001$, the certified accuracy of the original, PGD1, and PGD3 DNNs is 100% when each network is evaluated individually using verification tools. Our formulation reveals that, although robustness evaluations might yield similar results, this does not necessarily imply that the networks behave identically. For example, under a higher perturbation of $\delta = 0.01$, the certified accuracy of the original DNN drops to 89%, whereas the PGD-trained DNNs maintain a certified accuracy of 99%. This demonstrates that the results produced by our formulation accurately capture the differences in the networks' behaviors. Further investigations reveal that increasing δ to 0.04 leads to a more pronounced drop in the certified accuracy of PGD1 compared to PGD3, with PGD1's accuracy falling to 29%, while PGD3's remains at 87%. This is also reflected in the established implication results for $\delta = 0.001$, where PGD1 has 20% verified LROM with respect to PGD3, compared to 42% for PGD3 with respect to PGD1.

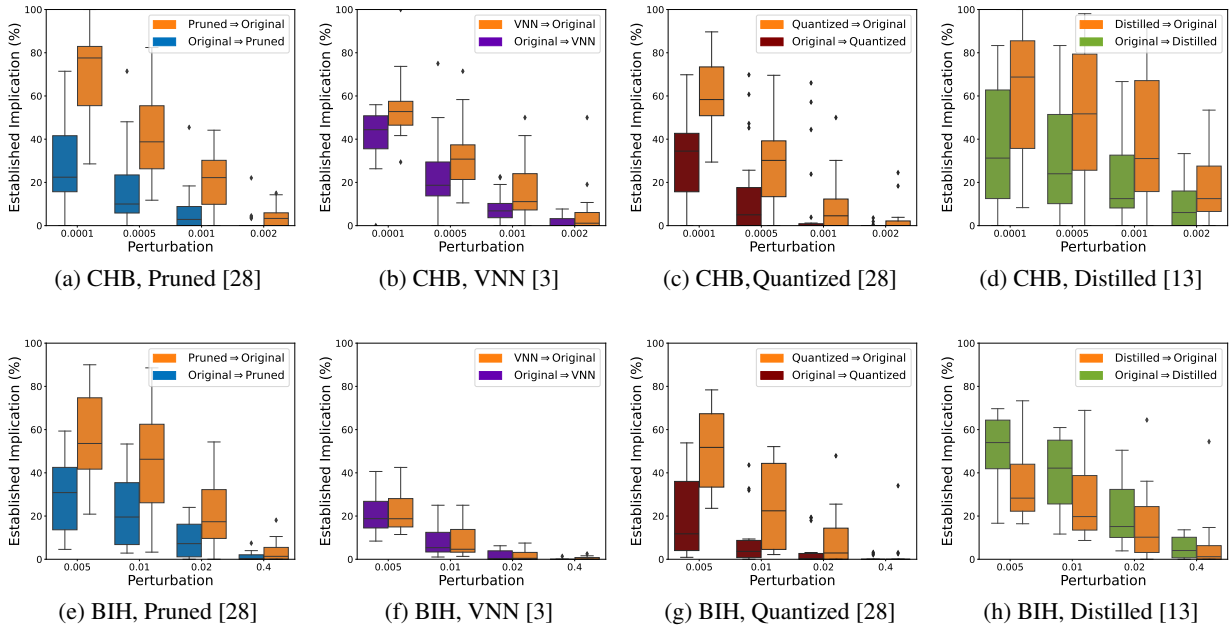
Table 2: Comparison of established implication and certified accuracy of the original, PGD1, and PGD3 DNNs. In the established implication section, the column $\Longrightarrow$ row shows that, e.g., in 42% of the samples, PGD3 makes a correct decision whenever PGD1 does.

	Established Implication			Certified Accuracy		
	Org.	PGD1	PGD3	$\delta = 0.001$	$\delta = 0.01$	$\delta = 0.04$
Org.	-	0%	1%	100%	89%	17%
PGD1	57%	-	20%	100%	99%	29%
PGD3	57%	42%	-	100%	99%	87%

4.3.4 *Two Real-World Medical Datasets*

In this section, we explore the established implication of DNNs trained on two real-world medical datasets, the CHB-MIT and MIT-BIH.

CHB-MIT Dataset: We explore the established implication of convolutional DNNs trained on the CHB-MIT dataset to categorize EEG signals of patients with epileptic seizures as captured in Figure 3a– 3d. Here, the x-axis shows different perturbation values applied to the input of a pair of original and compact networks. The general pattern of the established implication of pruned, quantized, and distilled networks is that we observe original DNNs have a higher established implication than their compact counterparts. Moreover, the number of established cases decreases with increasing perturbation. This can be



(a) CHB, Pruned [28] (b) CHB, VNN [3] (c) CHB, Quantized [28] (d) CHB, Distilled [13]

(e) BIH, Pruned [28] (f) BIH, VNN [3] (g) BIH, Quantized [28] (h) BIH, Distilled [13]

Figure 3: Box plots illustrate the established implication of convolutional DNNs trained on all patients in the CHB-MIT [24] and MIT-BIH [10] datasets for Original and Compact networks. For each patient in the dataset, we evaluate the implication between an Original and a Compact network using the patient's own data and aggregate the results across all patients to present them in the box plots.

caused by an actual decrease in LROM over a neighborhood, or by exacerbated over-approximation generated by the formulation. However, Figure 3b shows that the average established implication of VNNs is comparable to that of their original counterparts.

MIT-BIH dataset: In this section, we assess the established implication of convolutional DNNs trained on the MIT-BIH dataset to categorize ECG signals from patients with cardiac arrhythmia, as demonstrated in Figures 3e– 3h. Similar to DNNs trained on the CHB-MIT dataset, the number of established samples drops as perturbation increases, either due to reduced LROM across a range of perturbed inputs or increased over-approximation generated by the formulation. The behavior of MBP-pruned and quantized networks is also similar to that of CHB-MIT DNNs, with the established implication of original networks being higher than that of their corresponding pruned and quantized ones. However, the results of VNN-pruned networks are slightly different, as the established implication is closer to that of their original counterparts. Moreover, distilled networks display a different pattern, where their established implication is higher than that of their corresponding original networks.

5 Related Work

Evaluating and comparing the performance of networks is crucial in machine learning. For instance, both verification and adversarial techniques have been used to compare robustness of networks and their compacted versions. The previous studies by [19] and [15] for pruning, and by [7] for quantization used established verification techniques [14, 30] to separately assess local robustness of a network and its compacted version.

On the other hand, the work in [29] applies two white box attacks including Fast Gradient Sign Method (FGSM) [11] and PGD [20] to assess local robustness. None of the previous approaches can formally establish, even in the presence of adversarial examples, that a network makes a correct decision each time the other network does.

In this context, ReLUDiff [22] and NeuroDiff [23] present valuable techniques for analyzing functional differences between neural networks, focusing on verifying whether two models behave identically. As noted by Paulsen et al. [22], many tools focus on single-network behavior, limiting their ability to verify relational properties. In contrast, we establish an implication property that ensures one network is at least as correct as another. Instead of neuron-wise tracking like ReLUDiff and NeuroDiff, our method directly optimizes the final-layer difference, allowing comparison between different architectures.

Kleine Büning et al. [16] use MILP over clustered input regions to verify symmetric neural network equivalence, but does not capture directional correctness needed for comparing original and compact models. Unlike Narodytska et al. [21], who verify single binarized networks via SAT, and Eleftheriadis et al. [9], who perform symmetric equivalence checking with SMT over the entire input space, our approach targets local directional guarantees between networks, enabling asymmetric verification.

6 Conclusions

In this work, we propose a formulation to compare two networks in relation to each other over an entire input region. Specifically, we establish the foundation for formal local implication between two networks, i.e., $\mathcal{N}_2 \overset{\boldsymbol{D}}{\Longrightarrow} \mathcal{N}_1$, within an entire input region $\boldsymbol{D}$. In this context, network $\mathcal{N}_1$ consistently makes a correct decision every time network $\mathcal{N}_2$ does in the entire input region $\boldsymbol{D}$. The proposed formulation is relevant in the context of several application domains, e.g., for comparing a trained network and its corresponding compact (e.g., pruned, quantized, distilled) network. We evaluate our formulation using the MNIST, CIFAR10, and two real-world medical datasets, to show its relevance.

Acknowledgements

This work is partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation and by the European Union (EU) Interreg Program. We would like to also thank Kamran Hosseini for his initial input on an earlier draft of this work.

References

[1] S. Baghersalimi, A. Amirshahi, F. Forooghifar, T. Teijeiro, A. Aminifar, and D. Atienza Alonso. M2skd: Multi-to-single knowledge distillation of real-time epileptic seizure detection for low-power wearable systems. *ACM Transactions on Intelligent Systems and Technology*, 2024.

[2] A. Baninajjar, K. Hosseini, A. Rezine, and A. Aminifar. Safedeep: A scalable robustness verification framework for deep neural networks. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[3] A. Baninajjar, A. Rezine, and A. Aminifar. Vnn: Verification-friendly neural networks with hard robustness guarantees. In *Forty-first International Conference on Machine Learning (ICML)*, 2024.

[4] A. Baninajjar, A. Rezine, and A. Aminifar. Code and data for "formal local implication between two neural networks". https://github.com/anahitabn94/Formal_Local_Implication, 2025.

[5] A. Baninajjar, A. Rezine, and A. Aminifar. Formal local implication between two neural networks, 2025. URL https://arxiv.org/abs/2409.16726. Full version of this paper.

[6] P.-y. Chiang, R. Ni, A. Abdelkader, C. Zhu, C. Studer, and T. Goldstein. Certified defenses for adversarial patches. *Proceedings of International Conference on Learning Representations (ICLR)*, pages 1–17, 2020.

[7] K. Duncan, E. Komendantskaya, R. Stewart, and M. Lones. Relative robustness of quantized neural networks against adversarial attacks. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[8] R. Ehlers. Formal verification of piece-wise linear feed-forward neural networks. In D. D'Souza and K. Narayan Kumar, editors, *Automated Technology for Verification and Analysis*, pages 269–286, Cham, 2017. Springer International Publishing. ISBN 978-3-319-68167-2.

[9] C. Eleftheriadis, N. Kekatos, P. Katsaros, and S. Tripakis. On neural network equivalence checking using smt solvers. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 237–257. Springer, 2022.

[10] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. MIT-BIH Arrhythmia Database, 2000. URL https://www.physionet.org/content/mitdb/1.0.0/.

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[12] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. URL https://www.gurobi.com.

[13] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[14] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu. Safety verification of deep neural networks. In *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*, pages 3–29. Springer, 2017.

[15] 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.

[16] M. Kleine Büning, P. Kern, and C. Sinz. Verifying equivalence properties of neural networks with relu activation functions. In *International Conference on Principles and Practice of Constraint Programming*, pages 868–884. Springer, 2020.

[17] A. Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009.

[18] Y. LeCun. The mnist database of handwritten digits. *http://yann. lecun.com/exdb/mnist/*, 1998.

[19] Z. LI, T. Chen, L. Li, B. Li, and Z. Wang. Can pruning improve certified robustness of neural networks? *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

[20] 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.

[21] N. Narodytska, S. Kasiviswanathan, L. Ryzhyk, M. Sagiv, and T. Walsh. Verifying properties of binarized deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[22] B. Paulsen, J. Wang, and C. Wang. Reludiff: Differential verification of deep neural networks. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pages 714–726, 2020.

[23] B. Paulsen, J. Wang, J. Wang, and C. Wang. Neurodiff: scalable differential verification of neural networks using fine-grained approximation. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*, pages 784–796, 2020.

[24] A. Shoeb. CHB-MIT Scalp EEG Database, 2010. URL https://physionet.org/content/chbmit/.

[25] G. Singh, T. Gehr, M. Püschel, and M. Vechev. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages (PACMPL)*, 3:1–30, 2019.

[26] D. Sopic, A. Aminifar, A. Aminifar, and D. Atienza Alonso. Real-time event-driven classification technique for early detection and prevention of myocardial infarction on wearable systems. *IEEE Transactions on Biomedical Circuits and Systems*, 12(5):982–992, 2018.

[27] D. Sopic, A. Aminifar, and D. Atienza. e-glass: A wearable system for real-time detection of epileptic seizures. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2018.

[28] S. Ugare, G. Singh, and S. Misailovic. Proof transfer for fast certification of multiple approximate neural networks. *Proceedings of the ACM on Programming Languages*, 6(OOPSLA1):1–29, 2022.

[29] L. Wang, G. W. Ding, R. Huang, Y. Cao, and Y. C. Lui. Adversarial robustness of pruned neural networks. 2018.

[30] S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C.-J. Hsieh, and J. Z. Kolter. Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. *Advances in Neural Information Processing Systems*, 34:29909–29921, 2021.