# VERIFICATION OF SIGMOID-LIKE NEURAL NETWORKS VIA SYMBOLIC RELAXATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Formal robustness verification is essential for providing rigorous guarantees that a neural network's predictions remain unchanged with perturbations. For networks with sigmoid-like activations, a central challenge is to construct relaxations that faithfully capture their transcendental relations while remaining amenable to efficient verification. We propose a symbolically derived family of rational function relaxations, parametric in a relaxation order $k$. The relaxations are sound and convergent, and they yield a $k$-complete verification procedure, meaning that robust instances with a strictly positive margin can be certified with a finite relaxation order. Building on this family, we develop a scalable robustness verification method. Experiments on image classification benchmarks show that our method often outperforms the representative linear-relaxation baselines with tighter certified bounds on the robustness margin and improves certified robustness accuracy. Together, these results highlight the potential of rational relaxations beyond conventional linear bounds and suggest a new direction for certified robustness.

## 1 INTRODUCTION

Neural networks (NNs) have demonstrated remarkable capabilities and have been widely adopted across a broad range of applications (Krizhevsky et al., 2012; Silver et al., 2016; Julian et al., 2016). Despite their success, NNs are known to be vulnerable to adversarial examples, where small input perturbations can lead to incorrect predictions (Costa et al., 2024). Such vulnerabilities raise serious concerns when NNs are deployed in safety-critical applications (Katz et al., 2017; Huang et al., 2017). While adversarial attacks can be performed relatively easily (Melamed et al., 2024; Cinà et al., 2025), proving robustness to such attacks is much harder (König et al., 2024). Sampling-based methods can empirically estimate the robustness under a prescribed input distribution (Webb et al., 2019; Tit et al., 2023), but cannot guarantee the absence of vulnerabilities over the whole continuous input domain. Thus, formal robustness verification has gained increasing attention as a principled approach to establishing the reliability and safety of NNs (Kaulen et al., 2025).

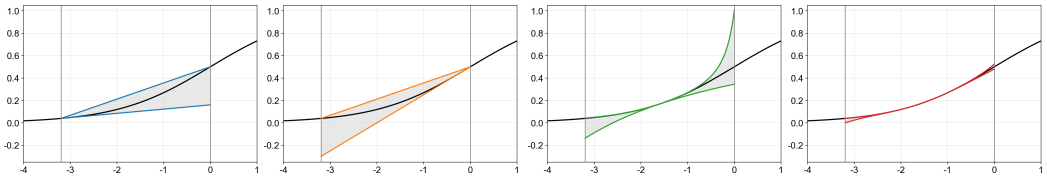
NN verification aims to determine whether an NN satisfies a formally specified property such as robustness for all inputs within a given set. In this context, soundness and completeness are two fundamental criteria of a verifier. Soundness means that, whenever the verifier certifies a property, the property indeed holds for all inputs, whereas completeness means that the verifier can certify every robust problem instance (Dvijotham et al., 2019; Jia & Rinard, 2020).

Sound but incomplete methods primarily include (1) abstract interpretation, which propagate conservative representations including intervals (Singh et al., 2019), star sets (Tran et al., 2019), polyhedra (Müller et al., 2022), polytopes (Mao et al., 2026), as well as polynomial zonotopes (Kochdumper et al., 2023), (2) relaxation-based methods, including convex-hull relaxation (Ma et al., 2025), linear relaxation (Zhang et al., 2018), and quadratic constraints (Fazlyab et al., 2022), and (3) optimization (Palma et al., 2024). These methods may remain inconclusive when the conservative analysis results in spurious behaviors introduced by abstraction or relaxation.

Complete verification methods can be categorized into exact constraint-solving methods, exact reachability methods, and systematic refinement methods. These methods typically target piecewise-linear activations such as ReLU, for which completeness is achieved by case splits into positive and negative values. Exact constraint-solving methods preserve the semantics of the supported activation functions and encode the NN together with the desired property as a satisfiability or optimization



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Naive resp. NeWise linear relaxations. (b) Our relaxations for $k_p = 0$ resp. $k_p = 1$.

Figure 1: Relaxations of the logistic sigmoid function. The first two plots show linear relaxations, while the last two plots show our relaxations for parameters $k_p = 0$ resp. $k_p = 1$.

problem, e.g., in first-order logic (Katz et al., 2017; Wu et al., 2024; Duong et al., 2025) or as mixed-integer programs (Zhao et al., 2024). Exact reachability methods (Tran et al., 2019; Lopez et al., 2023) propagate set representations without overapproximation. Building on sound abstractions, completeness can be recovered through systematic refinement procedures such as branch and bound (Lan et al., 2023; Zhou et al., 2024).

Beyond piecewise-linear activations, sigmoid-like functions are commonly used, including the logistic sigmoid and hyperbolic tangent (Dubey et al., 2022). These functions provide smooth nonlinear transformations, but their transcendental nature makes robustness verification computationally challenging. Indeed, the problem is not known to be decidable (Ivanov et al., 2019), and state-of-the-art verifiers use linear relaxation within a branch-and-bound framework (Shi et al., 2025).

An NN can be viewed as a composition of functions, where nonlinear activations are essential to its expressive power. Most existing verification methods obtain computational tractability by replacing these nonlinear activations with linear relaxations. Although such relaxations are inexpensive and solver-friendly, they inevitably discard part of the nonlinear structure. Consequently, the relaxed model may admit behaviors that are not realizable by the original NN, leading to loose output bounds and failed certification even when the original NN satisfies a given robustness specification.

In this paper, we propose a sound verification framework based on novel nonlinear rational relaxations for sigmoid-like activation functions. Identifying such relaxations over input intervals is nontrivial. We develop symbolic relaxations over the rational function field and reduce the verification problem to tractable optimization problems while preserving rigorous bounds. Moreover, our framework is scalable through a relaxation parameter $k_p$ controlling the precision, and we show that, in the limit, our reduction preserves completeness. Compared to linear relaxation methods, as shown in Fig. 1, our proposed relaxations can match the nonlinear structure of sigmoid-like activations directly. To the best of our knowledge, this is the first complete relaxation method for sigmoid-like activations without relying on exhaustive branch-and-bound refinement.

To address practical feasibility, we propose an instantiation of our framework, called *Syrt* (for **sy**mbolic **r**elaxa**t**ion), that also uses linear relaxations whenever they are sufficiently precise. We demonstrate experimentally that Syrt achieves stronger robustness results than the state-of-the-art linear-relaxation verifier NeWise (Zhang et al., 2022). Syrt can often even outperform the state-of-the-art verifier GenBaB (Shi et al., 2025), which is the branch-and-bound extension of the well-known $\alpha$-CROWN approach (Xu et al., 2021). The main contributions of this article are threefold:

1. We develop symbolic relaxations for sigmoid-like activation functions over the rational function field. The resulting relaxations are sound, scalable in a single parameter, and retain completeness in the limit.

2. We provide a feasible verification approach (Syrt) based on the proposed symbolic relaxations. Syrt transforms robustness specifications into certified optimization problems and allows the relaxation precision to be systematically improved via the tunable parameter.

3. We implement the proposed approach and evaluate it on image classification benchmarks. The results demonstrate that our approach can effectively verify neural networks and provides tighter certified bounds than linear relaxation baselines, often even outperforming the state-of-the-art branch-and-bound baseline.

## 2 Related Work

We focus our discussion on verification approaches for NNs with sigmoid-like activations; more general verification approaches have already been summarized in the previous section. Abstract-interpretation and bound-propagation methods include DeepCert (Wu & Zhang, 2021), NeWise (Zhang et al., 2022), and Ti-Lin (Xiao et al., 2025). Nonlinear and set-based approaches include sparse polynomial optimization (Newton & Papachristodoulou, 2023), Taylor-model-based verification (Ivanov et al., 2021), polynomial-zonotope reachability (Kochdumper et al., 2023), and relaxed star-set reachability (Choi et al., 2025). Constraint-solving and refinement-based approaches include VeriNet (Henriksen & Lomuscio, 2020), iSAT-based sigmoid reasoning (Grundt et al., 2021), nonlinear CEGAR (counterexample-guided abstraction refinement) in Marabou 2.0 (Wu et al., 2024), and branch and bound in GenBaB (Shi et al., 2025). Although these methods provide sound certificates and increasingly precise approximations, general completeness for NNs involving transcendental activations remains unresolved.

## 3 Preliminaries

**Neural Networks** For an NN with $m$ hidden layers, each layer $\ell$ ($1 \leq \ell \leq m$) has $d_\ell$ neurons, the input layer has $d_0$ neurons, and the output layer has $d_{m+1}$ neurons. For each layer $\ell$, let $x^{(\ell)} \in \mathbb{R}^{d_{\ell-1}}$ denote the input vector, $W^{(\ell)} \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ denote the weight matrix, $b^{(\ell)} \in \mathbb{R}^{d_\ell}$ denote the bias vector, and $\sigma^{(\ell)} \colon \mathbb{R}^1 \to \mathbb{R}^1$ denote the neuron-wise activation function. Similarly, the output layer uses $W^{(m+1)} \in \mathbb{R}^{d_{m+1} \times d_m}$ and $b^{(m+1)} \in \mathbb{R}^{d_{m+1}}$. Abusing notation, we also apply activations $\sigma^{(\ell)}$ to vectors component-wise. We write $y^{(\ell)} = h^{(\ell)}(x^{(\ell)})$ for the pre-activation values, where $h^{(\ell)} \colon \mathbb{R}^{d_{\ell-1}} \to \mathbb{R}^{d_\ell}$ is defined as $h^{(\ell)}(x^{(\ell)}) = W^{(\ell)} \cdot x^{(\ell)} + b^{(\ell)}$, and $z^{(\ell)} = \sigma^{(\ell)}(y^{(\ell)})$ for the post-activation values. Moreover, we set $x^{(\ell+1)} = z^{(\ell)}$. The output of the NN is obtained at the output layer as

$$y^{(m+1)} = h^{(m+1)}(x^{(m+1)}) = W^{(m+1)} x^{(m+1)} + b^{(m+1)}.$$

Equivalently, an $m$-layer NN is the function composition $f \colon \mathbb{R}^{d_0} \to \mathbb{R}^{d_{m+1}}$ defined as

$$f = h^{(m+1)} \circ \sigma^{(m)} \circ h^{(m)} \circ \cdots \circ \sigma^{(1)} \circ h^{(1)} \tag{1}$$

**Sigmoid-Like Activations** Sigmoid-like activation functions have been widely used in NNs (Bouraya & Belangour, 2024). They are typically monotonic and their graphs saturate toward finite horizontal asymptotes. In this paper, we focus on the widely-used activations *logistic sigmoid* and *hyperbolic tangent*, defined as

$$\mathrm{sigmoid}(x) = \frac{1}{1+e^{-x}} \quad \text{and} \quad \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}},$$

which are analytic functions with exponential terms.

**Robustness** In the following, we consider NNs as standard classifiers over $d_{m+1}$ classes. Let $t$ be the index of the target (ground-truth) class and let $f_t$ denote the corresponding $t$-th entry in the output vector. The classification goal is that $f_t > f_{t'}$ for all other class indices $t' \neq t$.

A classifier is called *robust* if a reasonable perturbation to its inputs does not alter the classification result. We consider $l_p$-norm perturbations, i.e., the perturbed input $x$ and the original input $x^{(1)}$ have distance less than some threshold $\epsilon$, i.e., $\|x - x^{(1)}\|_p \leq \epsilon$. Let $B_p(x^{(1)}, \epsilon)$ denote the ball in the $l_p$-norm with radius $\epsilon$ around $x^{(1)}$. Since analyzing global robustness over all possible inputs is often infeasible, we target *local robustness* for specific inputs.

**Definition 3.1** (Local robustness)**.** Given an NN $f \colon \mathbb{R}^{d_0} \to \mathbb{R}^{d_{m+1}}$, a robustness threshold $\epsilon \in \mathbb{R}^+$, and an input $x^{(1)} \in \mathbb{R}^{d_0}$ with target-class index $t$, the NN is (locally) *robust* if for any other class index $t' \in \{1, \ldots, d_{m+1}\} \setminus \{t\}$ we have that

$$\forall x \in B_p(x^{(1)}, \epsilon) \colon f_t(x) > f_{t'}(x).$$

For $l_p = l_\infty$, using Eq. (1) and letting $x_l = x^{(1)} - \epsilon, x_u = x^{(1)} + \epsilon$ (where $-/+$ are applied componentwise), the robustness verification problem can be formulated as the optimization problem

$$\inf_x \quad f_t(x) - f_{t'}(x) \quad \text{s.t.} \quad x_l \leq x \leq x_u.$$

Since NNs are not globally convex, computing the infimum over a prescribed domain can be difficult in practice. For our purpose, however, a certified (global) lower bound over the domain $[x_l, x_u]$ is sufficient, as we only need to prove that the lower bound is positive. For an NN $f$, let $\mathcal{L}$ denote the operator to obtain a certified lower bound. We call $\mathcal{L}(f_t(x) - f_{t'}(x))$ the *robustness margin* between the classes $t$ and $t'$. If that margin is positive, i.e.,

$$\forall x \in B_\infty(x^{(1)}, \epsilon) \colon \mathcal{L}(f_t(x) - f_{t'}(x)) > 0$$

for all $t' \neq t$, then $f$ is robust on input $x^{(1)}$. Moreover, a larger robustness margin is desirable, as it witnesses robustness to stronger perturbations.

## 4 Symbolic Relaxation for Sigmoid-Like Activations

Building on the preliminaries, we now present our symbolic relaxations for sigmoid-like activations.

Given a sigmoid-like function $\sigma$, we propose a family of lower- and upper-bounding relaxations $\sigma_l(x, k)$ and $\sigma_u(x, k)$ in the rational function field over $\mathbb{R}$ that sandwich $\sigma$. The family is parametric in $k$ and, for higher $k$, the relaxations approach $\sigma$. Moreover, as we show later in this section, the relaxations converge with arbitrary precision and are monotonic functions, and as we shall see in the experimental evaluation, the convergence is fast, so we do not require high values of $k$ in practice. Together, this makes the relaxations appealing both theoretically and practically.

We start by defining a family of $k$-relaxations.

**Definition 4.1.** Given a function $\sigma(x) \colon \mathbb{R}^1 \to \mathbb{R}^1$ and a bounded input interval $[x_l, x_u]$, two families of functions $\mathcal{F}_l = \{\sigma_l(x, k) \mid k \in \mathbb{N}_+\}$ and $\mathcal{F}_u = \{\sigma_u(x, k) \mid k \in \mathbb{N}_+\}$ in the rational function field over $\mathbb{R}$ are called a family of *$k$-relaxations* of $\sigma(x)$ if they satisfy, for all $x \in [x_l, x_u]$,

$$\sigma_l(x, k) < \sigma_l(x, k+1) < \cdots < \sigma(x) \tag{2}$$

and

$$\sigma(x) < \cdots < \sigma_u(x, k+1) < \sigma_u(x, k). \tag{3}$$

Next, we propose two $k$-relaxations for the common activations $\mathrm{sigmoid}(x)$ and $\tanh(x)$, satisfying Eq. (2) and Eq. (3). Proofs of these and further theorems are found in Appendix A.

**Theorem 4.2** ($k$-relaxation for $\mathrm{sigmoid}(x)$)**.** *Given $\sigma(x) = \frac{1}{1+e^{-x}}$, $x \in [x_l, x_u]$, $c \in [x_l, x_u]$, and $k \in \mathbb{N}_+$ such that $k \geq k_0$, the two sequences of functions*

$$\sigma_l(x, k) = 1 - \frac{1}{1 + e^c \cdot \sum_{i=0}^{2k-1} \frac{(x-c)^i}{i!}}, \quad \sigma_u(x, k) = \frac{1}{1 + e^{-c} \cdot \sum_{i=0}^{2k-1} \frac{(c-x)^i}{i!}} \tag{4}$$

*describe a family of $k$-relaxations of $\sigma(x)$, where $k_0 = \min\{k \in \mathbb{N}_+ \mid k > \frac{c - x_l - 1}{2}, 1 + e^c \cdot \sum_{i=0}^{2k-1} \frac{(x-c)^i}{i!} > 0\}$ for $\sigma_l(x)$ and $k_0 = \min\{k \in \mathbb{N}_+ \mid k > x_u - c + \frac{1}{2}, 1 + e^{-2c} \cdot \sum_{i=0}^{2k-1} \frac{(c-x)^i}{i!} > 0\}$ for $\sigma_u(x)$.*

**Theorem 4.3** ($k$-relaxation for $\tanh(x)$)**.** *Given $\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, $x \in [x_l, x_u]$, $c \in [x_l, x_u]$, and $k \in \mathbb{N}_+$ such that $k \geq k_0$, the two sequences of functions*

$$\sigma_l(x, k) = 1 - \frac{2}{1 + e^{2c} \cdot \sum_{i=0}^{2k-1} \frac{2^i (x-c)^i}{i!}}, \quad \sigma_u(x, k) = \frac{2}{1 + e^{-2c} \cdot \sum_{i=0}^{2k-1} \frac{2^i (c-x)^i}{i!}} - 1 \tag{5}$$

*describe a family of $k$-relaxations of $\sigma(x)$, where $k_0 = \min\{k \in \mathbb{N}_+ \mid k > c - x_l - \frac{1}{2}, 1 + \sum_{i=0}^{2k-1} \frac{2^i (x-c)^i}{i!} > 0\}$ for $\sigma_l(x)$ and $k_0 = \min\{k \in \mathbb{N}_+ \mid k > x_u - c + \frac{1}{2}, 1 + e^{-2c} \cdot \sum_{i=0}^{2k-1} \frac{2^i (c-x)^i}{i!} > 0\}$ for $\sigma_u(x)$.*

Importantly, the proposed relaxation is structurally different from a direct Taylor expansion of $\sigma(x)$. A truncated Taylor series provides a local polynomial approximation, but does not, by itself, guarantee a valid lower or upper bound on $\sigma(x)$ over the verification domain. In contrast, our construction is specifically designed to provide sound one-sided bounds, yielding a certified relaxation that is valid over the prescribed domain.

Our $k$-relaxations have two parameters: the relaxation order $k$ and the expansion point $c$.

The relaxation order $k$ must be chosen at least as large as $k_0$ according to the side conditions. Hence, in the following, we let $k = k_0 + k_p$ and focus on the free parameter $k_p \in \mathbb{N}$ instead. The parameter $k_p$ provides a way to control scalability: the higher we choose it, the more precise the relaxations get, at the price of higher complexity.

The expansion point $c$ can be chosen freely in the interval $[x_l, x_u]$. As $k_0$ depends on $c$, its choice may also be impactful. Still, in this paper, we focus on the parameter $k_0$, as it can be controlled arbitrarily, whereas $c$ is bounded.

While the relaxations approach $\sigma$, we still need to show that they converge.

**Theorem 4.4** (Convergence)**.** *The families $\mathcal{F}_l$ and $\mathcal{F}_u$ from Theorem 4.2 and Theorem 4.3 converge over the interval $[x_l, x_u]$, i.e., for $x \in [x_l, x_u]$, we have that*

$$\lim_{k\to+\infty} \sigma_l(x,k) = \sigma(x) \quad \textit{and} \quad \lim_{k\to+\infty} \sigma_u(x,k) = \sigma(x).$$

Finally, we establish that, for any parameter choice, the relaxations grow monotonically.

**Lemma 4.5.** *Each relaxation $\sigma_l(x,k) \in \mathcal{F}_l$, $\sigma_u(x,k) \in \mathcal{F}_u$ as defined in Theorem 4.2 and Theorem 4.3 is monotonically increasing in $x$ over $[x_l, x_u]$.*

## 5 Verification Approach for Neural Networks

Having established our univariate relaxations for $\sigma\colon \mathbb{R}^1 \to \mathbb{R}^1$ together with their most important properties, we now present two algorithms for robustness verification based on our proposed symbolic $k$-relaxations. The first algorithm enjoys completeness, i.e., can achieve arbitrary precision. The second algorithm is a variant that trades off completeness for practical speedups.

To obtain symbolic bounds for a deep NN, we follow prior work (Zhang et al., 2018; 2022) and apply our relaxations via symbolic substitution and function composition. Note that the composition does not leave the rational function field, and the order between two univariate functions (cf. Eq. (2) and Eq. (3)) is preserved after composition in the multivariate case. We choose the lower resp. upper bound depending on the sign of a neuron's incoming weights. Formally, let $W_-^\ell$ and $W_+^\ell$ be the versions of matrix $W^\ell$ with only the negative resp. positive entries (and all other entries replaced by zeros). Then, setting $x_l^{(1)} = x_l$ and $x_u^{(1)} = x_u$, the relaxations for the pre-activation values of layer $\ell$ are

$$\begin{aligned} y_l^{(\ell)} &= h_l^{(\ell)}(x^{(\ell)}) = W_-^{(\ell)} \cdot x_u^{(\ell)} + W_+^{(\ell)} \cdot x_l^{(\ell)} + b^{(\ell)}, \\ y_u^{(\ell)} &= h_u^{(\ell)}(x^{(\ell)}) = W_+^{(\ell)} \cdot x_u^{(\ell)} + W_-^{(\ell)} \cdot x_l^{(\ell)} + b^{(\ell)}, \end{aligned} \tag{6}$$

and the relaxations for the post-activation values are defined componentwise for the $i$-th neuron as

$$x_l^{(\ell+1,i)} = z_l^{(\ell,i)} = \sigma_l^{(\ell)}(y_l^{(\ell,i)}, k), \quad x_u^{(\ell+1,i)} = z_u^{(\ell,i)} = \sigma_u^{(\ell)}(y_u^{(\ell,i)}, k). \tag{7}$$

Using Lemma 4.5, this describes a family of $k$-relaxations.

Fig. 2 illustrates a simple example of the proposed relaxation. Given symbolic lower and upper bounds (i.e., expressions) on the hidden layer, we derive symbolic bounds for the $i$-th neuron. For example, $\frac{1+(x^{1,1}+x^{1,2})}{2+(x^{1,1}+x^{1,2})} = x_l^{2,1} < x^{2,1} < x_u^{2,1} = \frac{1}{2-(x^{1,1}+x^{1,2})}$. Unlike linear relaxations that may fail to certify certain class separations, our relaxation can achieve arbitrary precision.

Next, we incorporate these relaxations for NNs in a general framework for robustness verification.

**Robustness Verification Framework** Recall that robustness verification can be posed as an optimization problem. Integrating our relaxations, we can reformulate the problem as the following feasibility problem:

$$\begin{aligned} & y_l^{(m+1,t)} - y_u^{(m+1,t')} > 0 \quad \text{s.t.} && (8) \\ & y_l^{(\ell)} = W_-^{(\ell)} \cdot x_u^{(\ell)} + W_+^{(\ell)} \cdot x_l^{(\ell)} + b^{(\ell)}, y_u^{(\ell)} = W_+^{(\ell)} \cdot x_u^{(\ell)} + W_-^{(\ell)} \cdot x_l^{(\ell)} + b^{(\ell)}, && (1 \le \ell \le m+1) \\ & x_l^{(\ell+1,i)} = \sigma_l^{(\ell)}(y_l^{(\ell,i)}, k), x_u^{(\ell+1,i)} = \sigma_u^{(\ell)}(y_u^{(\ell,i)}, k), && (1 \le \ell \le m) \\ & x_l^{(1)} = x_l, \quad x_u^{(1)} = x_u. \end{aligned}$$

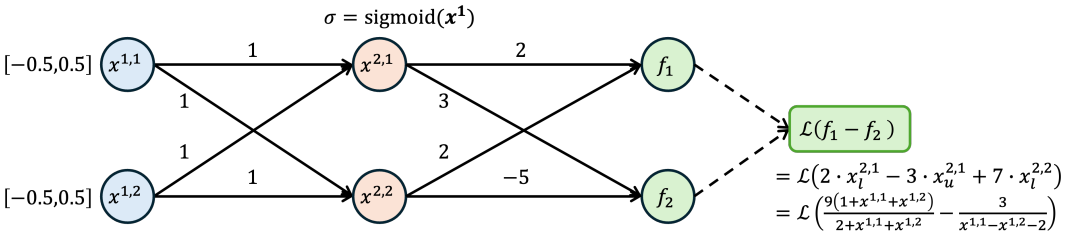



Figure 2: Example rational relaxation of a one-hidden-layer sigmoid NN for binary classification. A positive margin verifies that the two classes are robustly separated. Rational relaxation avoids direct reasoning over transcendental expressions by reformulating verification as rational-function optimization, yielding a more tractable algebraic problem while preserving rigorous certification.

In summary, System (8) provides conservative symbolic output bounds by retaining the affine transformations exactly (cf. Eq. (6)) and replacing the activation function of each neuron with sound lower and upper relaxations over its pre-activation interval (cf. Eq. (7)). These neuron-wise expressions are composed layer by layer to form an overapproximation of the original NN's behavior. Minimizing the margin over the relaxed model yields a certified lower bound $\mathcal{L}(f_t(x) - f_{t'}(x))$.

**Soundness and Completeness** Our framework is sound, i.e., whenever System (8) has a solution, then the NN is robust. This holds because the relaxations are conservative, which is immediate for the (exact) affine transformations and follows directly from Theorem 4.2 resp. Theorem 4.3 for $\sigma$.

Recall that a verifier is complete if, whenever the NN is robust, the verifier can reach that conclusion. Indeed, our relaxations have that property in the following sense. If the robustness margin is positive, then there exists a relaxation order $k^*$ such that System (8) has a solution for all $k \geq k^*$.

**Definition 5.1** ($k$-completeness)**.** A relaxation framework is *$k$-complete* if the relaxation margin $\mathcal{L}_k$ derived for System (8) converges to the true margin $\mathcal{L}$ as the relaxation parameter $k$ increases, i.e.,

$$\lim_{k \to +\infty} \mathcal{L}_k = \mathcal{L}.$$

**Corollary 5.2.** *The proposed relaxation framework is sound and $k$-complete.*

*Proof.* We already argued for soundness. Theorem 4.4 shows convergence of the relaxation. For any $\delta > 0$, if the original robustness property holds with a strictly positive margin $\mathcal{L} > 0$, there exists a finite parameter $k^*$ such that $\mathcal{L} - \mathcal{L}_{k^*} < \delta$, and thus $\mathcal{L}_k > 0$ for all $k \geq k^*$. □

Thus, simply incrementing $k$ and solving the corresponding feasibility problem in a loop yields a complete algorithm. Note that there is no conflict with the fact that robustness verification is not known to be decidable: we have only reasoned about the feasibility of System (8), but a constraint solver may not always be able to derive a positive verdict.

**Hybrid Algorithm** Non-convex problems like System (8) are generally difficult to solve. We propose a practical variant called *Syrt* that sacrifices completeness for better efficiency by combining both linear and nonlinear relaxations.

Algorithm 1 shows the pseudocode. For each opponent class $t'$, we rewrite the output margin $f_t(x) - f_{t'}(x)$ as a function of the last hidden layer by combining the corresponding output weights and biases as our objective $\mathcal{O}$ (line 2). Next, we build the constraints $\mathcal{C}$ of System (8), i.e., we formulate lower and upper symbolic relaxations for each hidden neuron and add them ("ADD") to $\mathcal{C}$.

For our nonlinear relaxations, we require concrete bounds on the pre-activation values $(\hat{y}_l^{(\ell,i)}, \hat{y}_u^{(\ell,i)})$ (line 7) to determine the relaxation parameters $c^{(\ell,i)}$ and $k_0^{(\ell,i)}$ (cf. Section 4). In our implementation, we get these bounds via linear relaxation (Zhang et al., 2022), which is coarse but fast. The final relaxation order is $k = k_0^{(\ell,i)} + k_p$ for each neuron (line 11).

Instead of always using our proposed nonlinear relaxations, in simple cases, we use a linear relaxation $\overline{\sigma}$ for efficiency. Linear relaxations are precise over intervals where $\sigma$ is also close to a straight

**Algorithm 1** Syrt (Robustness Verification)

**Input:** Neural network $f$; input region $(x_l, x_u)$; target class $t$; relaxation parameter $k_p$
**Output:** Certified lower-bound margin $\mathcal{L}$

1: **for** each opponent class $t' \neq t$ **do**
2: $\quad\mathcal{O} \leftarrow (W_t^{(m+1)} - W_{t'}^{(m+1)})x^{(m+1)} + b_t^{(m+1)} - b_{t'}^{(m+1)}$
3: $\quad\mathcal{C} \leftarrow$ InitializeConstraintProgram()
4: $\quad$**for** each hidden layer $\ell = 1, \ldots, m$ **do**
5: $\quad\quad$**for** each neuron $i = 1, \ldots, d_\ell$ **do**
6: $\quad\quad\quad$Add($\mathcal{C}$, $y_l^{(\ell,i)} = \ldots, y_u^{(\ell,i)} = \ldots$) *// given in Eq. (8)*
7: $\quad\quad\quad$$(\hat{y}_l^{(\ell,i)}, \hat{y}_u^{(\ell,i)}) \leftarrow$ PreactivationBounds()
8: $\quad\quad\quad$**if** IsLinearCase($\sigma^{(\ell)}$, $\hat{y}_l^{(\ell,i)}$, $\hat{y}_u^{(\ell,i)}$) **then**
9: $\quad\quad\quad\quad$Add($\mathcal{C}$, $x_l^{(\ell+1,i)} = \overline{\sigma}_l^{(\ell)}(y_l^{(\ell,i)})$, $x_u^{(\ell+1,i)} = \overline{\sigma}_u^{(\ell)}(y_u^{(\ell,i)})$)
10: $\quad\quad\quad$**else**
11: $\quad\quad\quad\quad$$k_0^{(\ell,i)} \leftarrow$ LowestOrder($\hat{y}_l^{(\ell,i)}$, $\hat{y}_u^{(\ell,i)}$)
12: $\quad\quad\quad\quad$Add($\mathcal{C}$, $x_l^{(\ell+1,i)} = \sigma_l^{(\ell)}(y_l^{(\ell,i)}, k_0^{(\ell,i)} + k_p)$, $x_u^{(\ell+1,i)} = \sigma_u^{(\ell)}(y_u^{(\ell,i)}, k_0^{(\ell,i)} + k_p)$)
13: $\quad\quad\quad$**end if**
14: $\quad\quad$**end for**
15: $\quad$**end for**
16: $\quad\mathcal{L}_{t,t'} \leftarrow$ SolveLowerBound$\big(\mathcal{O}, \mathcal{C}\big)$
17: **end for**
18: **return** $\mathcal{L} = \min\{\mathcal{L}_{t,t'} \mid t' \neq t\}$

line. For sigmoid-like activations, one can typically find ranges where this is the case. Specifically, we apply linear relaxation if $\hat{y}_l^{(\ell)} \geq 5$ or $\hat{y}_u^{(\ell)} \leq -5$ for the two activation functions we consider (line 8). In these ranges, the slope is $\leq 0.00665$ for the logistic sigmoid and $\leq 0.00019$ for the hyperbolic tangent, respectively.

In the end, we solve the objective $\mathcal{O}$ subject to the collected constraints $\mathcal{C}$ (line 16) and obtain the final margin as the minimum over all opponent classes $t'$ (line 18).

## 6 Experimental Evaluation

We empirically validate Syrt on NNs trained on image classification tasks. We consider three common datasets (MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), and CIFAR-10 (Krizhevsky et al., 2009)) and six pretrained feedforward architectures with either logistic sigmoid or hyperbolic tangent activations from prior work (Zhang et al., 2022). We use two state-of-the-art baselines based on linear relaxation. The first baseline is *NeWise* (Zhang et al., 2022). The second baseline is *GenBaB* (Shi et al., 2025), which is the iterative branch-and-bound extension of $\alpha$-*CROWN* (Xu et al., 2021). For details on the experimental setup, see Appendix B.

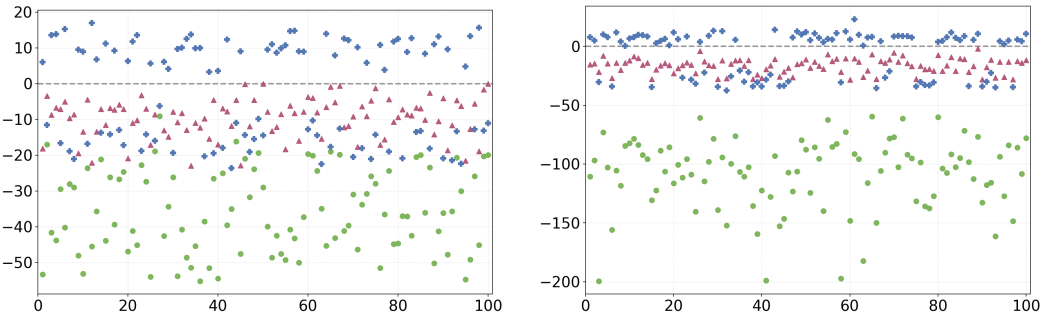
**Metrics** As qualitative measure, we report the certified robustness accuracy, i.e., the percentage of the sample images whose robustness can be certified. As quantitative measure, we report the robustness margin $\mathcal{L}$, i.e., the difference between the logit[1] of the target class $t$ and another competing class $t'$, aggregated over all samples (mean and standard deviation). To retain consistency across the three approaches, we select the competing class $t'$ as the non-target class with the highest value computed by NeWise. Recall that a positive margin proves robustness and higher margins imply better robustness. We also report the aggregated differences of the margins obtained by all approaches.

**Hyperparameters** The most important parameter of Syrt is the relaxation parameter $k_p \in \mathbb{N}$, which affects the degree of the symbolic expression and can be increased to trade off scalability for precision. Recall that the final relaxation order is $k = k_0 + k_p$, where $k_0$ is fixed analytically for the given neuron. To demonstrate the high precision, we use the lowest possible value (i.e., $k_p = 0$) in the main experiment. We study the effect of increasing $k_p$ in an ablation study. Moreover, the

[1]The raw output neuron's value before (softmax) normalization.

| Metric | $\text{sigmoid}_{3\times50}^{\text{MNIST}}$ | | $\text{sigmoid}_{3\times50}^{\text{FMNIST}}$ | | $\text{sigmoid}_{3\times50}^{\text{CIFAR}}$ | |
|---|---|---|---|---|---|---|
| | Syrt | GenBaB | Syrt | GenBaB | Syrt | GenBaB |
| acc. | **14**% | 8% | **54**% | 2% | **18**% | 0% |
| $\mathcal{L}$ | $-4.93 \pm 4.05$ | $-2.34 \pm 1.79$ | $-1.97 \pm 13.64$ | $-9.67 \pm 5.46$ | $-11.64 \pm 7.74$ | $-10.51 \pm 3.62$ |
| $\Delta\mathcal{L}$ | $24.12 \pm 19.90$ | $26.95 \pm 19.12$ | $33.91 \pm 23.53$ | $26.66 \pm 7.81$ | $13.10 \pm 8.64$ | $14.23 \pm 4.69$ |
| Metric | $\tanh_{3\times100}^{\text{MNIST}}$ | | $\tanh_{2\times100}^{\text{FMNIST}}$ | | $\tanh_{3\times50}^{\text{CIFAR}}$ | |
| | Syrt | GenBaB | Syrt | GenBaB | Syrt | GenBaB |
| acc. | **64**% | 0% | **52**% | 13% | 0% | **11**% |
| $\mathcal{L}$ | $-6.01 \pm 18.67$ | $-16.55 \pm 6.25$ | $-0.68 \pm 9.56$ | $-3.97 \pm 3.56$ | $-1.39 \pm 0.70$ | $-0.38 \pm 0.37$ |
| $\Delta\mathcal{L}$ | $102.53 \pm 23.14$ | $91.98 \pm 25.27$ | $40.85 \pm 24.51$ | $38.31 \pm 15.98$ | $2.79 \pm 1.07$ | $3.85 \pm 1.22$ |

Table 1: Performance comparison of Syrt (ours) with the baselines (NeWise and GenBaB). The metrics are certified robustness accuracy (acc.), robustness margin ($\mathcal{L}$), and the difference of the robustness margins between the respective approach and NeWise ($\Delta\mathcal{L}$). The naming of each benchmark instance follows the template $\sigma_{m\times n}^{D}$, where the NN has $m$ hidden layers of $n$ neurons each with activation function $\sigma$ and $D$ denotes the dataset.



(a) $\text{sigmoid}_{3\times50}^{\text{FMNIST}}$ instance. (b) $\tanh_{3\times100}^{\text{MNIST}}$ instance.

Figure 3: Visualization of the robustness margins $\mathcal{L}$ achieved by Syrt (+), GenBaB (▲), and NeWise (●) on three NN architectures.

expansion point $c^{(\ell,i)} \in [x_l^{(\ell,i)}, x_u^{(\ell,i)}]$ also affects the symbolic relaxation. In our implementation, we simply use the midpoint $c^{(\ell,i)} = (x_l^{(\ell,i)} + x_u^{(\ell,i)})/2$.

**Verification Results** Table 1 reports the main results. Syrt successfully certifies many of the samples that cannot be certified by NeWise and outperforms GenBaB in five of the six benchmark instances. Notably, for one instance, Syrt certifies 64% while GenBaB cannot certify any sample. These results demonstrate the potential of our symbolic relaxation approach, and also show that the precision of linear relaxation is inherently limited, often failing to verify a large number of cases that our approach could verify.

Fig. 3 visualizes the robustness margins obtained by the different approaches on two benchmark instances from Table 1. Recall that robustness is proven whenever the margin is positive (marked by the dashed line). As can be seen, Syrt often outperforms the baselines. The full plots for all benchmark instances are given in Appendix C. These plots show similar trends, with the exception of the last instance ($\tanh_{3\times50}^{\text{CIFAR}}$), where GenBaB consistently outperforms Syrt.

Across all benchmarks, the linear relaxation is used for 30.57% of the 110,000 hidden neurons. The instance where it is used the most is $\tanh_{3\times50}^{\text{CIFAR}}$, at 76.33%, which also explains why GenBaB outperforms Syrt (as Syrt mostly degenerates to a linear relaxation). Moreover, the value $k_0 = 1$ is used most commonly (25.19% of the neurons).

Table 2: Effect of $k_p$ on the certified robustness accuracy and margin, using the same benchmark instances as in Table 1.

| Instance | $k_p = 0$ | | $k_p = 1$ | | $k_p = 2$ | |
|---|---|---|---|---|---|---|
| | acc. | $\mathcal{L}$ | acc. | $\mathcal{L}$ | acc. | $\mathcal{L}$ |
| $\text{sigmoid}_{3\times50}^{\text{MNIST}}$ | 58% | $0.48 \pm 2.02$ | 65% | $0.87 \pm 1.97$ | 66% | $0.93 \pm 1.94$ |
| $\text{sigmoid}_{2\times200}^{\text{FMNIST}}$ | 87% | $1.67 \pm 1.79$ | 90% | $2.02 \pm 1.78$ | 90% | $2.02 \pm 1.79$ |
| $\text{sigmoid}_{3\times50}^{\text{CIFAR}}$ | 67% | $0.36 \pm 0.63$ | 68% | $0.38 \pm 0.62$ | 68% | $0.38 \pm 0.62$ |
| $\tanh_{3\times100}^{\text{MNIST}}$ | 17% | $-7.29 \pm 8.15$ | 18% | $-7.56 \pm 7.71$ | 15% | $-7.76 \pm 7.16$ |
| $\tanh_{2\times100}^{\text{FMNIST}}$ | 37% | $-2.10 \pm 3.94$ | 37% | $-1.40 \pm 4.07$ | 42% | $-1.03 \pm 3.80$ |
| $\tanh_{3\times50}^{\text{CIFAR}}$ | 74% | $0.38 \pm 0.58$ | 79% | $0.50 \pm 0.56$ | 79% | $0.50 \pm 0.55$ |

**Ablation Study** Theorem 4.4 states that higher values for $k$ (resp. $k_p$) yield a tighter approximation. While we used $k_p = 0$ in the previous experiment, we now investigate the effect of increasing that value. The results are summarized in Table 2.

In most cases, we see wider robustness margins and an increase in certified robustness accuracy for higher values of $k_p$. For the benchmark instance $\text{sigmoid}_{3\times50}^{\text{MNIST}}$, we can verify an additional 8% of the samples for $k_p = 2$. However, for the benchmark instance $\tanh_{3\times100}^{\text{MNIST}}$, the robustness margin actually shrinks again. Since the symbolic expression gets more complex for higher $k_p$, and since we have not raised the solver timeout for this experiment, the solver can sometimes only find looser bounds than before in the given time limit (even if the optimal bounds would be tighter).

Overall, the improvements are noteworthy but modest compared to the basic version with $k_p = 0$. We thus conclude that low values of $k_p$ suffice in practice.

**Concluding Remarks and Limitations** Our approach empirically achieves superior precision over the linear baseline NeWise. This may be expected due to the nonlinear advantage, but for low values of $k$, our bounds may in fact be looser, as the relaxations do not approximate $\sigma$ uniformly over the given interval and may diverge fast near the boundaries (cf. Fig. 1). This also motivated our hybrid algorithm. On the other hand, approaches based purely on linear relaxations are more efficient since they only require a linear solver. Our framework also relies on an external way to compute pre-activation bounds, and the more precise these bounds are, the better the results we obtain. In principle, we could reiterate our approach to obtain even tighter bounds, but we found that the additional cost is typically not worth it. Moreover, the linear baselines have received lots of engineering effort, including GPU parallelization, unlike our current implementation.

## 7 Conclusion

We have presented a framework for verifying robustness of neural networks with sigmoid-like activations based on symbolic relaxations. Our framework transforms the verification problem into a constraint problem over rational functions using sound symbolic bounds whose precision can be systematically improved by increasing the relaxation order to achieve $k$-completeness. By integrating the resulting constraints with linear relaxations, our hybrid algorithm Syrt provides a trade-off between verification precision and computational cost. Experiments show that Syrt is practical and can outperform state-of-the-art baselines.

Directions for future work include an integration of Syrt into a portfolio approach, e.g., first applying a linear approach and only falling back to Syrt when unsuccessful. Moreover, we want to study sign determination toward complete verification and automatic selection of $k_p$.

### AI USE STATEMENT

We have used generative AI tools for polishing the paper such as spellchecking as well as for assisting with the implementation of our prototype software. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, or interpreting results. Moreover, the following use of AI is not applicable to our work: generating synthetic datasets, cleaning and reformatting datasets, or supporting qualitative and thematic data analysis. We have reviewed all AI-assisted context. We take responsibility for the final content of this work, including text or artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

We state the number and range of values tried per (hyper-)parameter during development of the paper, along with the criterion used for selecting the final parameter setting. All source code required for conducting and analyzing the experiments is included in the supplementary material. We specify the computing infrastructure used for running experiments (hardware and software), including CPU model, amount of memory, operating system, as well as names and versions of relevant software libraries and frameworks. We describe evaluation metrics used and explain the motivation for choosing these metrics. We list all final (hyper-)parameters used for each model/algorithm in the paper's experiments. All datasets drawn from the existing literature are publicly available and accompanied by appropriate citations.

## REFERENCES

Sara Bouraya and Abdessamad Belangour. A comparative analysis of activation functions in neural networks: unveiling categories. *Bulletin of Electrical Engineering and Informatics*, 13(5):3301–3308, 2024. URL https://beei.org/index.php/EEI/article/view/7274.

Sung Woo Choi, Mykhailo Ivashchenko, Luan Viet Nguyen, and Hoang-Dung Tran. Reachability analysis of sigmoidal neural networks. *ACM Trans. Embed. Comput. Syst.*, 24(3):47:1–47:25, 2025. URL https://doi.org/10.1145/3627991.

Antonio Emanuele Cinà, Francesco Villani, Maura Pintor, Lea Schönherr, Battista Biggio, and Marcello Pelillo. $\sigma$-zero: Gradient-based optimization of $\ell 0$-norm adversarial examples. In *ICLR*. OpenReview.net, 2025. URL https://openreview.net/forum?id=JMPOqoe4tl.

Joana Cabral Costa, Tiago Roxo, Hugo Proença, and Pedro Ricardo M. Inácio. How deep learning sees the world: A survey on adversarial attacks & defenses. *IEEE Access*, 12:61113–61136, 2024. URL https://doi.org/10.1109/ACCESS.2024.3395118.

Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503:92–108, 2022. URL https://doi.org/10.1016/j.neucom.2022.06.111.

Hai Duong, ThanhVu Nguyen, and Matthew B. Dwyer. Neuralsat: A high-performance verification tool for deep neural networks. In *CAV*, volume 15932 of *LNCS*, pp. 409–423. Springer, 2025. URL https://doi.org/10.1007/978-3-031-98679-6_19.

Krishnamurthy (Dj) Dvijotham, Robert Stanforth, Sven Gowal, Chongli Qin, Soham De, and Pushmeet Kohli. Efficient neural network verification with exactness characterization. In *UAI*, volume 115 of *PMLR*, pp. 497–507. AUAI Press, 2019. URL http://proceedings.mlr.press/v115/dvijotham20a.html.

Mahyar Fazlyab, Manfred Morari, and George J. Pappas. Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming. *IEEE Trans. Autom. Control.*, 67(1):1–15, 2022. URL https://doi.org/10.1109/TAC.2020.3046193.

Dominik Grundt, Sorin Liviu Jurj, Willem Hagemann, Paul Kröger, and Martin Fränzle. Verification of sigmoidal artificial neural networks using isat. In *SNR*, volume 361 of *EPTCS*, pp. 45–60, 2021. URL https://doi.org/10.4204/EPTCS.361.6.

Patrick Henriksen and Alessio R. Lomuscio. Efficient neural network verification via adaptive refinement and adversarial search. In *ECAI*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pp. 2513–2520. IOS Press, 2020. URL https://doi.org/10.3233/FAIA200385.

Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks. In *CAV*, volume 10426 of *LNCS*, pp. 3–29. Springer, 2017. URL https://doi.org/10.1007/978-3-319-63387-9_1.

Radoslav Ivanov, James Weimer, Rajeev Alur, George J. Pappas, and Insup Lee. Verisig: verifying safety properties of hybrid systems with neural network controllers. In *HSCC*, pp. 169–178. ACM, 2019. URL https://doi.org/10.1145/3302504.3311806.

Radoslav Ivanov, Taylor J. Carpenter, James Weimer, Rajeev Alur, George J. Pappas, and Insup Lee. Verisig 2.0: Verification of neural network controllers using taylor model preconditioning. In *CAV*, volume 12759 of *LNCS*, pp. 249–262. Springer, 2021. URL https://doi.org/10.1007/978-3-030-81685-8_11.

Kai Jia and Martin C. Rinard. Efficient exact verification of binarized neural networks. In *NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1385974ed5904a438616ff7bdb3f7439-Abstract.html.

Kyle D Julian, Jessica Lopez, Jeffrey S Brush, Michael P Owen, and Mykel J Kochenderfer. Policy compression for aircraft collision avoidance systems. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pp. 1–10. IEEE, 2016. doi: 10.1109/DASC.2016.7778091.

Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In *CAV*, volume 10426 of *LNCS*, pp. 97–117. Springer, 2017. URL https://doi.org/10.1007/978-3-319-63387-9_5.

Konstantin Kaulen, Tobias Ladner, Stanley Bak, Christopher Brix, Hai Duong, Thomas Flinkow, Taylor T. Johnson, Lukas Koller, Edoardo Manino, ThanhVu H. Nguyen, and Haoze Wu. The 6th international verification of neural networks competition (VNN-COMP 2025): Summary and results. *CoRR*, abs/2512.19007, 2025. URL https://doi.org/10.48550/arXiv.2512.19007.

Niklas Kochdumper, Christian Schilling, Matthias Althoff, and Stanley Bak. Open- and closed-loop neural network verification using polynomial zonotopes. In *NFM*, volume 13903 of *LNCS*, pp. 16–36. Springer, 2023. URL https://doi.org/10.1007/978-3-031-33170-1_2.

Matthias König, Annelot W. Bosman, Holger H. Hoos, and Jan N. van Rijn. Critically assessing the state of the art in neural network verification. *J. Mach. Learn. Res.*, 25:12:1–12:53, 2024. URL https://jmlr.org/papers/v25/23-0119.html.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pp. 1106–1114, 2012. URL https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

Jianglin Lan, Benedikt Brückner, and Alessio Lomuscio. A semidefinite relaxation based branch-and-bound method for tight neural network verification. In *AAAI*, pp. 14946–14954. AAAI Press, 2023. URL https://doi.org/10.1609/aaai.v37i12.26745.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998. URL https://doi.org/10.1109/5.726791.

Diego Manzanas Lopez, Sung Woo Choi, Hoang-Dung Tran, and Taylor T. Johnson. NNV 2.0: The neural network verification tool. In *CAV*, volume 13965 of *LNCS*, pp. 397–412. Springer, 2023. URL https://doi.org/10.1007/978-3-031-37703-7_19.

Zhongkui Ma, Zihan Wang, and Guangdong Bai. Convex hull approximation for activation functions. *Proc. ACM Program. Lang.*, 9(OOPSLA2):1007–1033, 2025. URL https://doi.org/10.1145/3763086.

Yuhao Mao, Yani Zhang, and Martin Vechev. Expressiveness of multi-neuron convex relaxations in neural network certification. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 86672–86695, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/8b7d8779c020fa3671ad267152362d56-Paper-Conference.pdf.

Odelia Melamed, Gilad Yehudai, and Adi Shamir. MALT powers up adversarial attacks. In *NeurIPS*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/4e8f257e054abd24c550d55e57cec274-Abstract-Conference.html.

Mark Niklas Müller, Gleb Makarchuk, Gagandeep Singh, Markus Püschel, and Martin T. Vechev. PRIMA: general and precise neural network certification via scalable convex hull approximations. *Proc. ACM Program. Lang.*, 6(POPL):1–33, 2022. URL https://doi.org/10.1145/3498704.

Matthew Newton and Antonis Papachristodoulou. Sparse polynomial optimisation for neural network verification. *Autom.*, 157:111233, 2023. URL https://doi.org/10.1016/j.automatica.2023.111233.

Alessandro De Palma, Harkirat Singh Behl, Rudy Bunel, Philip H. S. Torr, and M. Pawan Kumar. Scaling the convex barrier with sparse dual algorithms. *J. Mach. Learn. Res.*, 25:61:1–61:51, 2024. URL https://jmlr.org/papers/v25/21-0076.html.

Zhouxing Shi, Qirui Jin, Zico Kolter, Suman Jana, Cho-Jui Hsieh, and Huan Zhang. Neural network verification with branch-and-bound for general nonlinearities. In *TACAS*, volume 15696 of *LNCS*, pp. 315–335. Springer, 2025. URL https://doi.org/10.1007/978-3-031-90643-5_17.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nat.*, 529(7587):484–489, 2016. URL https://doi.org/10.1038/nature16961.

Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin T. Vechev. An abstract domain for certifying neural networks. *Proc. ACM Program. Lang.*, 3(POPL):41:1–41:30, 2019. URL https://doi.org/10.1145/3290354.

Karim Tit, Teddy Furon, and Mathias Rousset. Gradient-informed neural network statistical robustness estimation. In *AISTATS*, volume 206 of *PMLR*, pp. 323–334. PMLR, 2023. URL https://proceedings.mlr.press/v206/tit23a.html.

Hoang-Dung Tran, Diego Manzanas Lopez, Patrick Musau, Xiaodong Yang, Luan Viet Nguyen, Weiming Xiang, and Taylor T. Johnson. Star-based reachability analysis of deep neural networks. In *FM*, volume 11800 of *LNCS*, pp. 670–686. Springer, 2019. URL https://doi.org/10.1007/978-3-030-30942-8_39.

Stefan Webb, Tom Rainforth, Yee Whye Teh, and M. Pawan Kumar. Statistical verification of neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1xcx3C5FX.

Haoze Wu, Omri Isac, Aleksandar Zeljic, Teruhiro Tagomori, Matthew L. Daggitt, Wen Kokke, Idan Refaeli, Guy Amir, Kyle Julian, Shahaf Bassan, Pei Huang, Ori Lahav, Min Wu, Min Zhang, Ekaterina Komendantskaya, Guy Katz, and Clark W. Barrett. Marabou 2.0: A versatile formal analyzer of neural networks. In *CAV*, volume 14682 of *LNCS*, pp. 249–264. Springer, 2024. URL https://doi.org/10.1007/978-3-031-65630-9_13.

Yiting Wu and Min Zhang. Tightening robustness verification of convolutional neural networks with fine-grained linear approximation. In *AAAI*, pp. 11674–11681. AAAI Press, 2021. URL https://doi.org/10.1609/aaai.v35i13.17388.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL http://arxiv.org/abs/1708.07747.

Yuan Xiao, Yuchen Chen, Shiqing Ma, Chunrong Fang, Tongtong Bai, Mingzheng Gu, Yuxin Cheng, Yanwei Chen, and Zhenyu Chen. Tightening robustness verification of maxpool-based neural networks via minimizing the over-approximation zone. In *CVPR*, pp. 20695–20705. Computer Vision Foundation / IEEE, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Xiao_Tightening_Robustness_Verification_of_MaxPool-based_Neural_Networks_via_Minimizing_the_CVPR_2025_paper.html.

Kaidi Xu, Huan Zhang, Shiqi Wang, Yihan Wang, Suman Jana, Xue Lin, and Cho-Jui Hsieh. Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. In *ICLR*. OpenReview.net, 2021. URL https://openreview.net/forum?id=nVZtXBI6LNn.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *NeurIPS*, pp. 4944–4953, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/d04863f100d59b3eb688a11f95b0ae60-Abstract.html.

Zhaodi Zhang, Yiting Wu, Si Liu, Jing Liu, and Min Zhang. Provably tightest linear approximation for robustness verification of sigmoid-like neural networks. In *ASE*, pp. 80:1–80:13. ACM, 2022. URL https://doi.org/10.1145/3551349.3556907.

Haoruo Zhao, Hassan L. Hijazi, Haydn Thomas Jones, Juston Moore, Mathieu Tanneau, and Pascal Van Hentenryck. Bound tightening using rolling-horizon decomposition for neural network verification. In *CPAIOR*, volume 14743 of *LNCS*, pp. 289–303. Springer, 2024. URL https://doi.org/10.1007/978-3-031-60599-4_20.

Duo Zhou, Christopher Brix, Grani A. Hanasusanto, and Huan Zhang. Scalable neural network verification with branch-and-bound inferred cutting planes. In *NeurIPS*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/33d93e4dc57453e7667b20f62e7c0681-Abstract-Conference.html.

# A PROOFS

This appendix contains the proofs of all theoretical claims in the paper.

## A.1 PROOF OF THEOREM 4.2

Let $\sigma(x) = \frac{1}{1+e^{-x}}$ for $x \in [x_l, x_u]$, and let $k \in \mathbb{N}_+$ and $c \in [x_l, x_u]$. We divide the proof into two steps for the lower and upper bounds.

**Lower bound $\sigma_l$:** Given $k_0 = \min\{k \in \mathbb{N}_+ \mid k > \frac{c-x_l-1}{2}, 1 + e^c \cdot \sum_{i=0}^{2k-1} \frac{(x-c)^i}{i!} > 0\}$, for $k \geq k_0$ we need to show that $\sigma_l(x,k) = 1 - \frac{1}{1+e^c \cdot \sum_{i=0}^{2k-1} \frac{(x-c)^i}{i!}} < \sigma_l(x,k+1) < \cdots < \sigma(x)$ holds.

*Proof.* For $k > \frac{c-x_l-1}{2}$, $\frac{(x-c)^{2k}}{(2k+1)!} \cdot (2k+1+x-c) = \frac{(x-c)^{2k}}{(2k)!} + \frac{(x-c)^{2k+1}}{(2k+1)!} > 0$ holds. With Taylor expansion of $e^x$ on $c$, we have

$$\begin{aligned} e^c \cdot \sum_{i=0}^{2k-1} \frac{(x-c)^i}{i!} &< e^c \cdot \sum_{i=0}^{2k+1} \frac{(x-c)^i}{i!} < \cdots \\ &< e^c \cdot \sum_{i=0}^{+\infty} \frac{(x-c)^i}{i!} = e^x. \end{aligned}$$

Therefore, for $1 + e^c \cdot \sum_{i=0}^{2k-1} \frac{(x-c)^i}{i!} > 0$, we have

$$\begin{aligned} 1 - \frac{1}{1 + e^c \cdot \sum_{i=0}^{2k-1} \frac{(x-c)^i}{i!}} &< 1 - \frac{1}{1 + e^c \cdot \sum_{i=0}^{2k+1} \frac{(x-c)^i}{i!}} \\ &< \cdots < \frac{1}{1+e^{-x}}, \end{aligned}$$

directly yielding $\sigma_l(x,k) < \sigma_l(x,k+1) < \cdots < \sigma(x)$. □

**Upper bound $\sigma_u$:** Given $k_0 = \min\{k \in \mathbb{N}_+ \mid k > \frac{x_u-c-1}{2}, 1 + e^{-c} \cdot \sum_{i=0}^{2k-1} \frac{(c-x)^i}{i!}\}$, for $k > k_0$ we need to show that $\sigma(x) < \cdots < \sigma_u(x,k+1) < \sigma_u(x,k) = \frac{1}{1+e^{-c} \cdot \sum_{i=0}^{2k-1} \frac{(c-x)^i}{i!}}$ holds.

*Proof.* For $k > \frac{x_u-c-1}{2}$, $\frac{(c-x)^{2k}}{(2k)!} + \frac{(c-x)^{2k+1}}{(2k+1)!} > 0$ holds. With Taylor expansion of $e^{-x}$ on $c$, we have

$$\begin{aligned} e^{-c} \cdot \sum_{i=0}^{2k-1} \frac{(c-x)^i}{i!} &< e^{-c} \cdot \sum_{i=0}^{2k+1} \frac{(c-x)^i}{i!} < \cdots \\ &< e^{-c} \cdot \sum_{i=0}^{+\infty} \frac{(c-x)^i}{i!} = e^{-x}. \end{aligned}$$

Therefore, for $1 + e^{-c} \cdot \sum_{i=0}^{2k-1} \frac{(c-x)^i}{i!} > 0$, we have

$$\begin{aligned} \frac{1}{1+e^{-x}} &< \cdots < \frac{1}{1 + e^{-c} \cdot \sum_{i=0}^{2k+1} \frac{(c-x)^i}{i!}} \\ &< \frac{1}{1 + e^{-c} \cdot \sum_{i=0}^{2k-1} \frac{(c-x)^i}{i!}}, \end{aligned}$$

directly yielding $\sigma(x) < \cdots < \sigma_u(x,k+1) < \sigma_u(x,k)$. □

## A.2 PROOF OF THEOREM 4.3

Let $\sigma(x) = \tanh(x)$ for $x \in [x_l, x_u]$, and let $k \in \mathbb{N}_+$ and $c \in [x_l, x_u]$. We divide the proof into two steps for the lower and upper bounds.

**Lower bound $\sigma_l$:** Given $k_0 = \min\{k \in \mathbb{N}_+ \mid k > c - x_l - \frac{1}{2}, 1 + \sum_{i=0}^{2k-1} \frac{2^i(x-c)^i}{i!} > 0\}$, for $k > k_0$ we need to show that $\sigma_l(x,k) = 1 - \frac{2}{1+e^{2c}\cdot\sum_{i=0}^{2k-1}\frac{2^i(x-c)^i}{i!}} < \sigma_l(x,k+1) < \cdots < \sigma(x)$ holds.

*Proof.* For $k > c - x_l - \frac{1}{2}$, $\frac{2^{2k}(x-c)^{2k}}{(2k)!} + \frac{2^{2k+1}(x-c)^{2k+1}}{(2k+1)!} = \frac{2^{2k}(x-c)^{2k}}{(2k+1)!}\cdot(2k+1+2x-2c) > 0$ holds. With Taylor expansion of $e^{2x}$ on $c$, we have

$$\begin{aligned} e^{2c}\cdot\sum_{i=0}^{2k-1}\frac{2^i(x-c)^i}{i!} &< e^{2c}\cdot\sum_{i=0}^{2k+1}\frac{2^i(x-c)^i}{i!} < \cdots \\ &< e^{2c}\cdot\sum_{i=0}^{+\infty}\frac{2^i(x-c)^i}{i!} = e^{2x}. \end{aligned}$$

Therefore, for $1 + e^{2c}\cdot\sum_{i=0}^{2k-1}\frac{2^i(x-c)^i}{i!} > 0$, we have

$$\begin{aligned} &1 - \frac{2}{1+e^{2c}\cdot\sum_{i=0}^{2k-1}\frac{2^i(x-c)^i}{i!}} \\ <\ &1 - \frac{2}{1+e^{2c}\cdot\sum_{i=0}^{2k+1}\frac{2^i(x-c)^i}{i!}} < \cdots \\ <\ &1 - \frac{2}{1+e^{2c}\cdot\sum_{i=0}^{+\infty}\frac{2^i(x-c)^i}{i!}} = 1 - \frac{2}{e^{2x}+1} = \sigma(x), \end{aligned}$$

directly yielding $\sigma_l(x,k) < \sigma_l(x,k+1) < \cdots < \sigma(x)$. □

**Upper bound $\sigma_u$:** Given $k_0 = \min\{k \in \mathbb{N}_+ \mid k > x_u - c + \frac{1}{2}, 1 + e^{-2c}\cdot\sum_{i=0}^{2k-1}\frac{2^i(c-x)^i}{i!} > 0\}$, for $k > k_0$ we need to show that $\sigma(x) < \cdots < \sigma_u(x,k+1) < \sigma_u(x,k) = \frac{2}{1+e^{-2c}\cdot\sum_{i=0}^{2k-1}\frac{2^i(c-x)^i}{i!}} - 1$ holds.

*Proof.* For $k > x_u - c + \frac{1}{2}$, $\frac{2^{2k}(c-x)^{2k}}{(2k)!} + \frac{2^{2k+1}(c-x)^{2k+1}}{(2k+1)!} = \frac{2^{2k}(c-x)^{2k}}{(2k+1)!}\cdot(2k+1+2c-2x) > 0$ holds. With Taylor expansion of $e^{-2x}$ on $c$, we have

$$\begin{aligned} e^{-2c}\cdot\sum_{i=0}^{2k-1}\frac{2^i(c-x)^i}{i!} &< e^{-2c}\cdot\sum_{i=0}^{2k+1}\frac{2^i(c-x)^i}{i!} < \cdots \\ &< e^{-2c}\cdot\sum_{i=0}^{+\infty}\frac{2^i(c-x)^i}{i!} = e^{-2x}. \end{aligned}$$

Therefore, for $1 + e^{-2c}\cdot\sum_{i=0}^{2k-1}\frac{2^i(c-x)^i}{i!} > 0$, we have

$$\begin{aligned} \frac{2}{1+e^{-2x}} - 1 &= \frac{2}{1+e^{-2c}\cdot\sum_{i=0}^{+\infty}\frac{2^i(c-x)^i}{i!}} - 1 < \cdots \\ &< \frac{2}{1+e^{-2c}\cdot\sum_{i=0}^{2k+1}\frac{2^i(c-x)^i}{i!}} - 1 \\ &< \frac{2}{1+e^{-2c}\cdot\sum_{i=0}^{2k-1}\frac{2^i(c-x)^i}{i!}} - 1 = \sigma(x), \end{aligned}$$

directly yielding $\sigma(x) < \cdots < \sigma_u(x,k+1) < \sigma_u(x,k)$. □

### A.3 Proof of Theorem 4.4

We need to show that, on the interval $[x_l, x_u]$, the relaxation in Eq. (4) converges to $\mathrm{sigmoid}(x)$ and the relaxation in Eq. (5) converges to $\tanh(x)$.

*Proof.* As shown in Theorem 4.2 (Appendix A.1) and Theorem 4.3 (Appendix A.2), the relaxation of the denominator approaches the Taylor expansion $e^{\lambda x}$, where $\lambda \in \mathbb{Z}$ and $x \in [x_l, x_u]$. Let $\mathcal{R}(x,k)$ denote the Lagrange remainder of the $k$-relaxation of $e^{\lambda x}$,

$$\begin{aligned}\mathcal{R}(x,k) &= \frac{f^{(2k-1)}(x)}{(2k-1)!}(x-c)^{2k-1} + \frac{f^{(2k)}(x)}{(2k)!}(x-c)^{2k} \\ &= (x-c)^{2k-1} \cdot \frac{\lambda^{2k-1}e^{\lambda x}}{(2k-1)!} + (x-c)^{2k} \cdot \frac{\lambda^{2k}e^{\lambda x}}{(2k)!}.\end{aligned}$$

Letting $a = \max_{x\in[x_l,x_u]}\{e^{\lambda x}\} \in \mathbb{R}$, we have that

$$\lim_{k\to+\infty} a \cdot \frac{(\lambda(x-c))^{2k-1}}{(2k-1)!} = 0,$$

which implies that $\mathcal{R}(x,k) \to 0$. Hence, the proposed relaxations converge to $\sigma(x)$. □

### A.4 Proof of Lemma 4.5

We divide the proof into four steps, one for each relaxation.

**Case $\sigma = \mathrm{sigmoid}$, lower bound $\sigma_l$:** We need to show that $\sigma_l(x,k)$ in Eq. (4) is monotonically increasing.

*Proof.* Let $q(x,k) = e^c \cdot \sum_{i=0}^{2k-1} \frac{(x-c)^i}{i!}$, which implies that $\sigma_l(x,k) = 1 - \frac{1}{1+q(x,k)}$. We analyze the first derivative

$$\frac{d\sigma_l}{dx} = \frac{\frac{d}{dx}q}{(1+q)^2},$$

where

$$\frac{d}{dx}q = e^c \cdot \sum_{i=0}^{2k-2} \frac{(x-c)^i}{i!}.$$

If $x - c \geq 0$, then $\frac{d}{dx}q \geq 0$. Conversely, if $x - c < 0$, then $\frac{d}{dx}q > e^x > 0$. Thus, in both cases, $\frac{d\sigma_l}{dx} \geq 0$, which yields the claim. □

**Case $\sigma = \mathrm{sigmoid}$, upper bound $\sigma_u$:** We need to show that $\sigma_u(x,k)$ in Eq. (4) is monotonically increasing.

*Proof.* denote $q(x,k) = e^{-c} \cdot \sum_{i=0}^{2k-1} \frac{(c-x)^i}{i!}$, which means $\sigma_u(x,k) = \frac{1}{1+q(x,k)}$.

$$\frac{d\sigma_u}{dx} = -\frac{\frac{d}{dx}q}{(1+q)^2},$$

where

$$\frac{d}{dx}q = -e^{-c} \cdot \sum_{i=0}^{2k-2} \frac{2^i(c-x)^i}{i!}$$

when $c - x \geq 0$, $\frac{d}{dx}q \leq 0$. when $c - x < 0$, we have

$$e^{-c} \cdot \sum_{i=0}^{2k-2} \frac{2^i(c-x)^i}{i!} > e^{-x} > 0,$$

then $\frac{d\sigma_u}{dx} \geq 0$, which means $\sigma_u(x,k)$ is monotonically increasing. □

**Case $\sigma = \tanh$, lower bound $\sigma_l$:** We need to show that $\sigma_l(x,k)$ in Eq. (5) is monotonically increasing.

*Proof.* Let $q(x,k) = e^{2c} \cdot \sum_{i=0}^{2k-1} \frac{2^i(x-c)^i}{i!}$, which means $\sigma_l(x,k) = 1 - \frac{2}{1+q(x,k)}$.

$$\frac{d\sigma_l}{dx} = \frac{2 \cdot \frac{d}{dx}q}{(1+q)^2},$$

where

$$\frac{d}{dx}q = 2e^{2c} \cdot \sum_{i=0}^{2k-2} \frac{2^i(x-c)^i}{i!}$$

when $x - c \geq 0$, $\frac{d}{dx}q \geq 0$. when $x - c < 0$, $\frac{d}{dx}q > e^{2x} > 0$.

then $\frac{d\sigma_l}{dx} \geq 0$, which means $\sigma_l(x,k)$ is monotonically increasing. □

**Case $\sigma = \tanh$, upper bound $\sigma_u$:** We need to show that $\sigma_u(x,k)$ in Eq. (5) is monotonically increasing.

*Proof.* denote $q(x,k) = e^{-2c} \cdot \sum_{i=0}^{2k-1} \frac{2^i(c-x)^i}{i!}$, which means $\sigma_u(x,k) = \frac{2}{1+q(x,k)} - 1$.

$$\frac{d\sigma_u}{dx} = -\frac{2 \cdot \frac{d}{dx}q}{(1+q)^2},$$

where

$$\frac{d}{dx}q = -2e^{-2c} \cdot \sum_{i=0}^{2k-2} \frac{2^i(c-x)^i}{i!}$$

when $c - x \geq 0$, $\frac{d}{dx}q \leq 0$. when $c - x < 0$,

$$2e^{-2c} \cdot \sum_{i=0}^{2k-2} \frac{2^i(c-x)^i}{i!} > e^{-2x} > 0,$$

then $\frac{d\sigma_u}{dx} \geq 0$, which means $\sigma_u(x,k)$ is monotonically increasing. □

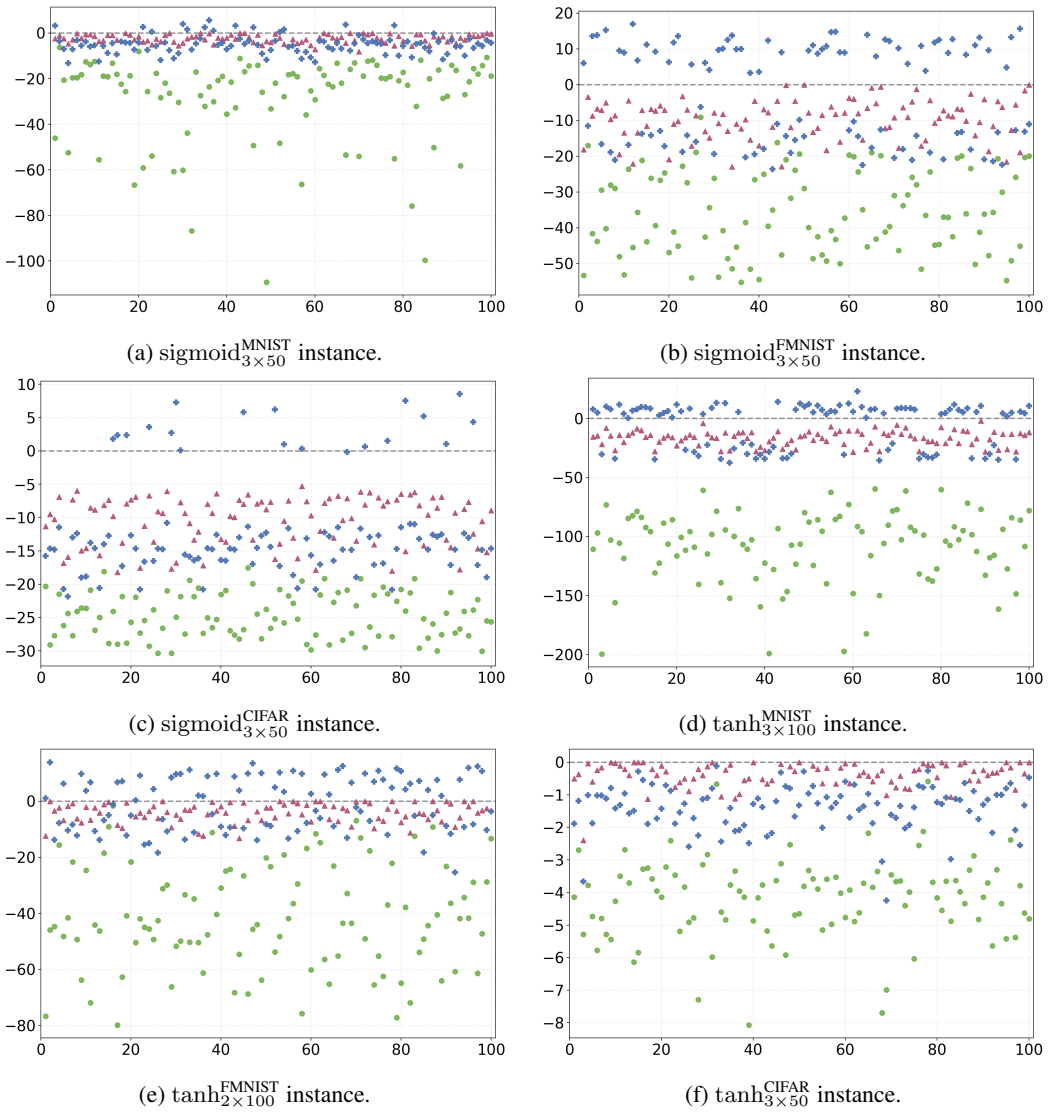
## B Experimental Details

This appendix provides a detailed description of the experimental setup.

All experiments were conducted on a MacOS 26.5 laptop (Apple Silicon M3, arm64, 24 GB RAM), running Python 13.0 and using Gurobi 12.0.3 as the underlying optimization solver. We did not use a GPU. Since all verification approaches are deterministic, we only report results for a single run. We confirmed that additional runs lead to the same results.

In principle, the resulting verification problem can be solved completely using Cylindrical Algebraic Decomposition (CAD), an exact real-algebraic decision procedure, provided that the rational constraints are reduced to an equivalent semialgebraic formulation. CAD can therefore provide a complete decision procedure for the relaxed verification problem. However, its computational cost grows rapidly with the number of variables and polynomial degrees, making it impractical for NNs of realistic dimension. In our implementation, we therefore use Gurobi as the optimization backend and certify the property using the solver-reported global objective bound ("`ObjBound`"[2]), which returns a sound lower bound. This choice provides a practical trade-off between rigorous certification and computational scalability.

[2]https://docs.gurobi.com/projects/optimizer/en/current/reference/attributes/model.html#objbound



(a) $\text{sigmoid}^{\text{MNIST}}_{3\times50}$ instance.

(b) $\text{sigmoid}^{\text{FMNIST}}_{3\times50}$ instance.

(c) $\text{sigmoid}^{\text{CIFAR}}_{3\times50}$ instance.

(d) $\tanh^{\text{MNIST}}_{3\times100}$ instance.

(e) $\tanh^{\text{FMNIST}}_{2\times100}$ instance.

(f) $\tanh^{\text{CIFAR}}_{3\times50}$ instance.

Figure 4: Robustness margins $\mathcal{L}$ achieved by the different approaches across various NN architectures and datasets.

For each verification instance, we set a time limit of 10 seconds for Gurobi, which returns a sound lower bound on the robustness margin $\mathcal{L}$ upon termination, even if timed out. We also use a 10-second timeout for NeWise and GenBaB.

The verification target is the standard local robustness for image classification under $l_\infty$-bounded perturbations with $\epsilon = 5/255$. Following (Zhang et al., 2022), in our evaluation, we select the first 100 samples from the test set that can be correctly classified by the NN but could not be correctly certified by NeWise (i.e., hard instances). For the ablation study, we use $\epsilon = 2/255$ and the first 100 correctly classified samples.

## C Plots of Robustness Margins

This appendix contains additional plots.

Fig. 4 shows the robustness margins achieved by Syrt and the baseline approaches on all six combinations of NN architectures and datasets from our evaluation.