ON THE FORMAL VERIFICATION OF 1-LIPSCHITZ NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Certifying neural networks against ℓ_2-norm adversarial perturbations is essential for safety-critical deployment. While 1-Lipschitz architectures offer scalable, computationally inexpensive ℓ_2 certified radii, a persistent gap remains between their empirical robustness and guaranteed bounds. To bridge this gap, we investigate formal verification (e.g., LiRPA) on 1-Lipschitz models. Focusing on Gradient Norm Preserving (GNP) networks—a state-of-the-art approach for ℓ_2 certified robustness—we establish a necessary condition for LiRPA to outperform the Lipschitz certified radius, proving that this condition is not guaranteed when parsing standard GNP architectures off-the-shelf. To resolve this limitation, we introduce a LiRPA-compatible graph factorization of GNP activations (e.g. GroupSort2). Leveraging this formulation, we conduct the first evaluation comparing LiRPA against standard Lipschitz certified radii across diverse hyperparameter configurations. Notably, our analysis reveals the first family of networks where analytical Lipschitz bounds outperform consistently LiRPA. We identify two key hyperparameter dimensions governing this behavior: optimization trade-offs and Lipschitz constraint strictness. Specifically, relaxing optimization to enable higher clean accuracy, combined with looser Lipschitz constraints, creates a regime where LiRPA yields provably superior certificates. Based on this insight, we introduce a hybrid certification framework that unifies the scalable geometry of 1-Lipschitz networks with the localized precision of linear bound propagation. This hybrid strategy scales LiRPA certification to deeper architectures. Empirically, on Imagenette—where standalone solvers fail—our framework scales to a VGG16 architecture, achieving 84.5% clean accuracy and advancing certified robust accuracy from 58.0% (standard Lipschitz bounds) to 72.5%.

1 INTRODUCTION

Deep neural networks are notoriously vulnerable to small, imperceptible adversarial perturbations (Szegedy et al., 2014; Goodfellow et al., 2015). Because empirical defenses are frequently circumvented by adaptive attacks (Carlini & Wagner, 2017; Athalye et al., 2018), formally certifying robustness against ℓ_2-norm bounded perturbations has become an essential endeavor. While approaches like Randomized Smoothing (Cohen et al., 2019) provide scalable ℓ_2 robustness bounds, their guarantees are statistical. To safely deploy models in critical environments, we require *deterministic* guarantees (Tsuzuku et al., 2018).

To achieve this deterministic, provable ℓ_2 robustness, multiple research trajectories have emerged, notably formal methods and Lipschitz-constrained networks. **Formal methods** evaluates robustness by casting the neural network into solvers (e.g., SMT Katz et al. (2017) or MILP Tjeng et al. (2019)). Linear bound propagation methods (e.g., the Linear Relaxation Perturbation based Analysis, LiRPA family) (Wang et al., 2021; Zhang et al., 2018; Singh et al., 2019; Gowal et al., 2019; Xu et al., 2021) have become the most scalable alternative for neural networks. These verifiers offer the critical advantage of being training and architecture agnostic. LiRPA provides local a posteriori guarantees, but they ultimately remain computationally intractable for deep architectures (Xu et al., 2020). In contrast, 1**-Lipschitz networks** (Anil et al., 2019; Serrurier et al., 2021; Li et al., 2019; Trockman & Kolter, 2021; Boissin et al., 2025) apply constraints during training to bound the global Lipschitz constant. This approach yields computationally inexpensive ℓ_2 certificates (Equation (1)), which we refer to as Lipschitz-certified radius. Among these, Gradient Norm Preserving (GNP) networks

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.

currently represent the state-of-the-art architectures for achieving the tightest Lipschitz-certified radii (Anil et al., 2019). Despite targeting the exact same geometric notions of robustness, these two communities remain largely separate.

In deterministic certification, a persistent challenge is the gap between guaranteed certified bounds and empirical robustness (e.g., AutoAttack Croce & Hein (2020)), a limitation that also persists in 1-Lipschitz certificates (Boissin et al., 2025). To bridge this certification gap, this paper investigates the application of formal verification to 1-Lipschitz models. By establishing a unified theoretical and empirical analysis comparing native Lipschitz-certified radii against formal LiRPA bounds, our main contributions are:

- **Theoretical Limitations of Native Lipschitz Certificates (Section 3):** We systematically analyze why Lipschitz-certified radius underestimate empirical robustness, proving that structural bottlenecks and optimization dynamics (e.g., training temperature) prevent networks from converging to the exact Signed Distance Function (SDF).
- **Prerequisites for Formal Verification & Graph Factorization (Section 4):** We establish the condition required for formal verification to outperform Lipschitz bounds on GNP architectures (Sections 4.1 and 4.2). To satisfy these conditions, we introduce a novel LiRPA-compatible graph factorization of GroupSort2 (GS2), the intrinsic activation function of standard GNP networks (Section 4.3). By fully leveraging the linear dependency between adjacent neurons, this optimized representation tightens the verification bounds and reduces the memory cost.
- **Scalable Hybrid Verification Framework (Section 5):** We propose a scalable hybrid certification strategy that combines the global geometric guarantees of 1-Lipschitz backbones with the localized precision of linear bound propagation.
- **Empirical Evaluation & Scalability (Section 6):** We empirically verify the superiority of our graph factorization in Section 6.2. Furthermore, we provide a comprehensive analysis demonstrating the impact of key hyperparameters on certification tightness (Section 6.3). We finally validate the scalability of our hybrid framework by successfully deploying formal verification on deep architectures like VGG16 (Section 6.4).

2 Related Works

From 1-Lipschitz to GNP Architectures Throughout this work, we consider a classification neural network $f : \mathbb{R}^n \to \mathbb{R}^C$, where n represents the input dimension and C denotes the number of classes. A network f is globally L-Lipschitz with respect to the ℓ_2-norm if its output variations are bounded by L, such that $\|f(x) - f(y)\|_2 \leq L\|x - y\|_2$ for all $x, y \in \mathbb{R}^n$. Without loss of generality, 1-Lipschitz neural networks remains mathematically equivalent to the general L-Lipschitz case for classification tasks: logit scaling by a constant does not alter decision boundaries, and this scaling factor is naturally absorbed by the temperature hyperparameter τ during τ-Cross Entropy training (Béthune et al., 2022). We construct these networks by composing 1-Lipschitz layers. A primary advantage of this architectural constraint is that it yields computationally inexpensive, robustness certificates. To certify a prediction around an input sample x_0 with true class label c, certification frameworks evaluate the logit margin conditional on a correct classification: $M(x_0) = \mathbb{I}_{\{\arg\max_k f(x_0)_k = c\}} \left(f(x_0)_c - \max_{j \neq c} f(x_0)_j\right)$. Because f is globally 1-Lipschitz, the margin function M inherits a $\sqrt{2}$-Lipschitz constant with respect to the ℓ_2-norm. As derived by Tsuzuku et al. (2018), this guarantees that the prediction remains invariant within an ℓ_2-ball $\mathcal{B}_2(x_0, \epsilon_{lip}) = \{x \mid \|x - x_0\|_2 \leq \epsilon_{lip}\}$, yielding the certified radius:

$$\epsilon_{lip}(x_0) = \frac{M(x_0)}{\sqrt{2}}. \tag{1}$$

However, because standard Lipschitz-certified radii ϵ_{lip} are inherently pessimistic and underestimate the true robustness (i.e., the exact geometric distance to the decision boundary), recent research focuses on closing this certification gap between certified lower bounds and empirical upper bounds (Li et al., 2019; Béthune et al., 2022). To overcome these certification bottlenecks, the field evolved toward GNP architectures (Anil et al., 2019; Trockman & Kolter, 2021; Boissin et al., 2025). By pairing orthogonal weight layers with norm-preserving non-linearities such as GroupSort2 (GS2), defined as $GS2(x, y) = (\min(x, y), \max(x, y))$ (Anil et al., 2019), GNP networks aim to satisfy the *Eikonal condition* (maintaining unit gradient norm almost everywhere). This allows the network's

output margin to approximate the true Signed Distance Function (SDF), maximizing Lipschitz-certified radius tightness (Boissin et al., 2025). Applying orthogonality to convolutional layers remains challenging. While various frequency and early spatial-domain methods exist (Trockman & Kolter, 2021; Xu et al., 2022; Singla & Feizi, 2021; Li et al., 2019; Su et al., 2022), two modern spatial-domain paradigms overcome prior computational and striding limitations to establish a rigorous baseline: Adaptive Orthogonal Convolutions (AOC) (Boissin et al., 2025) for exact structural orthogonality, and Reshaped Kernel Orthogonalization (RKO) (Li et al., 2019; Björck & Bowie, 1971) for a relaxed contractive bound (≤ 1). Despite these architectural advancements, structural bottlenecks and optimization dynamics can still cause Lipschitz-certified radii to underestimate true robustness. Consequently, recent verification literature often labels these analytical certificates as "LipNaive" (Chiu et al., 2025), presuming them to be inferior to the state-of-the-art formal methods we introduce in the following section.

Formal Verification and the Relaxation Barrier. To bridge the gap between analytical certificates and true robustness, formal verification uses backward bound propagation (Urban & Miné, 2021). LiRPA frameworks (Zhang et al., 2018) bound non-linear activations with affine bounds ($a_L x + b_L \leq \sigma(x) \leq a_U x + b_U$). Propagating these inequalities backward yields lower bounds on logit differences within an ℓ_2-ball $B_2(x_0, \epsilon) = \{x \mid \|x - x_0\|_2 \leq \epsilon\}$. For a correctly classified input x_0 (class c), certifying robustness at radius ϵ requires proving $f(x)_c > f(x)_j$ for all $j \neq c$ across $x \in B_2(x_0, \epsilon)$. LiRPA computes a radius-dependent lower bound for each pairwise logit difference $g_j(x) = f(x)_c - f(x)_j$, such that $\underline{g}_j(x_0, \epsilon) \leq \min_{x \in B_2(x_0, \epsilon)} (g_j(x))$ for all $j \neq c$. Robustness is guaranteed if $\min_{j \neq c} \underline{g}_j(x_0, \epsilon) > 0$. Thus, the LiRPA certified radius is:

$$\epsilon_{LiRPA}(x_0) = \sup \left\{ \epsilon \geq 0 \;\middle|\; \min_{j \neq c} \underline{g}_j(x_0, \epsilon) > 0 \right\}. \tag{2}$$

Backward LiRPA and Verification Challenges. Backward LiRPA methods—such as CROWN (Zhang et al., 2018), its advanced variants α-CROWN (Wang et al., 2021) and SDP-CROWN (Chiu et al., 2025)—offer a promising approach to recover the unproven robustness left by Lipschitz-certified radius. By composing affine bounds backward from output to input, they provide localized robustness guarantees. However, scaling these state-of-the-art solvers to deep GNP architectures presents structural and computational hurdles that motivate our contributions. First, the theoretical formulation of SDP-CROWN is built specifically for ReLU, making it natively incompatible with the bivariate GS2 non-linearity. Second, bound propagation inherently faces the Convex Relaxation Barrier (Salman et al., 2019), where over-approximation errors compound with depth depending on how the activation graph is parsed (Cohen et al., 2026). Finally, backward methods face a fundamental scaling wall: they are inherently recursive, and propagating an affine bound from an output dimension d_{out} through a latent dimension d_{latent} demands a prohibitive memory footprint scaling as $\mathcal{O}(d_{out} \times d_{latent})$. Overcoming these exact mathematical and computational barriers is the primary focus of this work.

The Intersection of Lipschitz Continuity and Bound Propagation. The relationship between LiRPA bound propagation and Lipschitz continuity has been leveraged in two distinct ways. Early methods like Fast-Lip (Weng et al., 2018) and RecurJac (Zhang et al., 2019) utilize linear relaxations to estimate the local Lipschitz constants of unconstrained networks. More recently, frameworks like Super-DeepG (Cohen et al., 2026) bridge the two paradigms for geometric robustness, using analytical Lipschitz bounds to handle input transformations (e.g., rotations) before applying LiRPA to the network's latent space. However, its theoretical formulation is restricted to low-dimensional perturbation domains. Unlike these approaches—which use verification to estimate Lipschitz constants or handle separate geometric spaces—our work investigates applying formal bound propagation to the logit margin of constrained 1-Lipschitz architectures.

Alternative Objective-Driven and Scaling Approaches. Beyond structural constraints, certified training (Huang et al., 2021; Shi et al., 2022) incorporates LiRPA-propagated bounds into the objective function as a penalty during optimization. However, computing bound propagation at every training step makes these methods notoriously difficult to scale. A parallel line of research focuses on scaling Lipschitz frameworks to larger datasets and architectures. Frameworks like GloRoNet (Leino et al., 2021), LiResNet (Hu et al., 2023; 2024), and BRONet (Lai et al., 2025) combine specialized blocks with objective-driven regularizations, while LipNeXt (Hu et al., 2026) replaces standard convolutions with spatial shifts to reach massive parameter scales. Although these architectures optimize margin

objectives to improve robustness at scale, there is no formal guarantee that their resulting Lipschitz-certified radius is optimal. Furthermore, existing formal verification methods cannot currently be applied to close this estimation gap: standard LiRPA solvers struggle to scale to such depths, and their reliance on residual connections lacks native support in SDP-CROWN.

3 Theoretical Limitations of Native Lipschitz Certificates

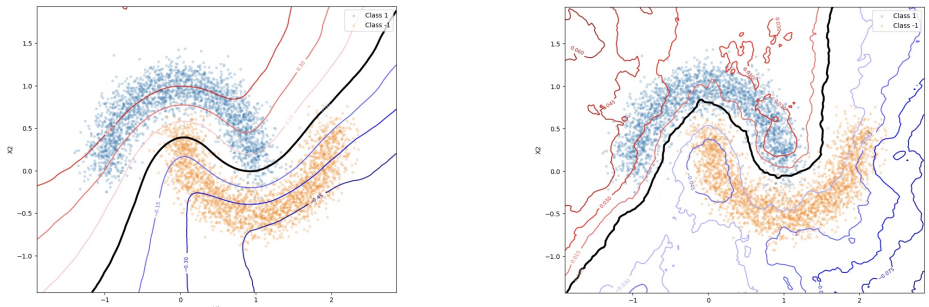
To understand why constrained 1-Lipschitz architectures often underestimate true robustness, we first review the geometric meaning of Lipschitz-certified radii. For theoretical clarity, we recall the binary classification setting where the network outputs a single scalar logit $f : \mathbb{R}^n \to \mathbb{R}$. In this setup, the decision boundary is the zero-level set $\mathcal{B} = \{x \in \mathbb{R}^n \mid f(x) = 0\}$, and the analytical Lipschitz-certified radii is the absolute value of the output (Tsuzuku et al., 2018): $\epsilon_{lip}(x) = |f(x)|$. As demonstrated in (Béthune et al., 2022), this analysis naturally extends to the multi-class regime by replacing $f(x)$ with the logit margin $M(x)$. For a correctly classified input x, we distinguish between its **true robustness**—the exact geometric distance to the decision boundary, $\epsilon_{true}(x) = \inf_{x' \in \mathcal{B}} \|x - x'\|_2$—and its **certified robustness**, the guaranteed lower bound provided by the scalar evaluation, $\epsilon_{lip}(x) = |f(x)|$. Because $\epsilon_{lip}(x) \leq \epsilon_{true}(x)$, we aim to minimize the certification gap $\Delta\epsilon(x) = \epsilon_{true}(x) - \epsilon_{lip}(x)$.

In order to have a tight certificate ($\Delta\epsilon = 0$), $f(x)$ must converge to the true Signed Distance Function (SDF) associated to the decision boundary $\mathcal{B}$ (Brau et al., 2023). A necessary condition for a function to be an SDF is satisfying the Eikonal equation (i.e., $||\nabla f(x)||_2 = 1$ almost everywhere). However, it is not a sufficient condition, we give a visual intuition in A.1. Achieving tight certificates is inherently prevented by two primary bottlenecks.

Structural Bottlenecks in Convolutions. The structural choices made to enforce 1-Lipschitz constraints act as a principal factor in the tightness of Lipschitz-certified radii. Specifically, the mathematical method used for orthogonalizing convolutions dictates the reachability of the Eikonal condition. For instance, Adaptive Orthogonal Convolution (AOC) enforces exact structural orthogonality by design. Conversely, methods relying on iterative approximation, such as Reshaped Kernel Orthogonalization (RKO) via Björck-Bowie iterations, correspond to an optimization-based, relaxed constraint. Furthermore, even when exact orthogonality is enforced, structural expansions within the network make it mathematically impossible to satisfy the Eikonal condition globally:

Proposition 3.1 (Eikonal Destruction)**.** *Let $W \in \mathbb{R}^{d_{out} \times d_{in}}$ be an expansion layer ($d_{out} > d_{in}$) with orthonormal columns ($W^T W = I_{d_{in}}$). For any outgoing gradient column vector $g_{out} \in \mathbb{R}^{d_{out}}$, the backward operator W^T preserves the gradient norm if and only if $g_{out} \in Im(W) = \left\{Wx \in \mathbb{R}^{d_{out}} \mid x \in \mathbb{R}^{d_{in}}\right\}$. Otherwise, the norm is strictly attenuated.*

We provide the proof and a concrete counterexample demonstrating this attenuation in Appendix A.1. Because standard networks rely on channel expansion, global gradients are attenuated, preventing the network from reaching the SDF.



(a) 2-Moons classifier level-set: Low τ. (b) 2-Moons classifier level-set: High τ.

Figure 1: The impact of temperature on the 2D optimization landscape.

Optimization Dynamics and Temperature (τ). Respecting the Eikonal condition is only a necessary condition for approximating the SDF. So even in architectures that preserve gradient norms and bypass structural bottlenecks, optimization dynamics can prevent convergence to the true SDF (Béthune et al.,

2022). The training temperature τ plays a role in changing the complexity of the optimal decision boundary, and consequently on the complexity of the SDF associated to this boundary. At low temperatures, the optimization landscape remains smooth, allowing the network to approximate the SDF (Figure 1a). Conversely, as τ increases, the network must satisfy larger logit margins. Because the global gradient norm is capped at 1, the model cannot steepen its slopes. Instead, it is forced to inflate the geometric distance by aggressively weaving and folding the decision boundary around local data clusters (Figure 1b). An empirical evaluation quantifying how both architectural choices (AOC vs. RKO) and temperature regimes impact this certification gap is presented in Appendix A.5.

Based on this theoretical understanding of why native analytical bounds leave unproven robustness, our objective is to recover this missing margin via post-hoc formal verification. In the following section, we establish the mathematical foundations required to successfully adapt bound propagation methods to these constrained architectures.

4 Unlocking Formal Verification for GNP Architectures

In this section, we analyze the relationship between the robustness radius certified by LiRPA and Lipschitz-certified radius. We first establish the necessary condition required for LiRPA to outperform Lipschitz-certified radius. Because the standard GS2 formulation fails to guarantee this condition, we develop a novel graph factorization that respects it unconditionally.

4.1 Prerequisites for Outperforming Lipschitz-certified radius

To the best of our knowledge, no prior work has established the mathematical prerequisites for linear bound propagation (LiRPA) to out-certify Lipschitz-certified radius. By evaluating the worst-case minimum of the verifier's bounding plane, we demonstrate that achieving a tighter certificate requires a strict norm-constrained property: the equivalent affine bounding vector must be at most L-Lipschitz.

Theorem 4.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be an L-Lipschitz classifier with a Lipschitz-certified radius ε_{lip}. If a linear lower bound $f(x) \geq \omega^\top x + b$ yields a certified radius ε_{LiRPA} such that $\varepsilon_{LiRPA} \geq \varepsilon_{lip}$, then the bounding function must be at most L-Lipschitz, requiring $\|\omega\|_2 \leq L$.*

The full derivation, along with a counterexample demonstrating that this condition is necessary but not sufficient, is provided in Section B.1. Without loss of generality, we focus all subsequent theoretical analysis on 1-Lipschitz networks ($L = 1$).

4.2 Standard Relaxations in GNP Networks

We now consider GNP networks with GS2 activations. Standard linear backward relaxation handle the non-linearity by applying independent linear relaxations to the $\min$ and $\max$ nodes. Intuitively, as both of these functions are bivariate, this approach concretizes x and y into independent bounding intervals. Consequently, the relaxation ignores the linear correlation between the two variables. When backpropagating through the network, this loss of information inflates the LiRPA bounds.

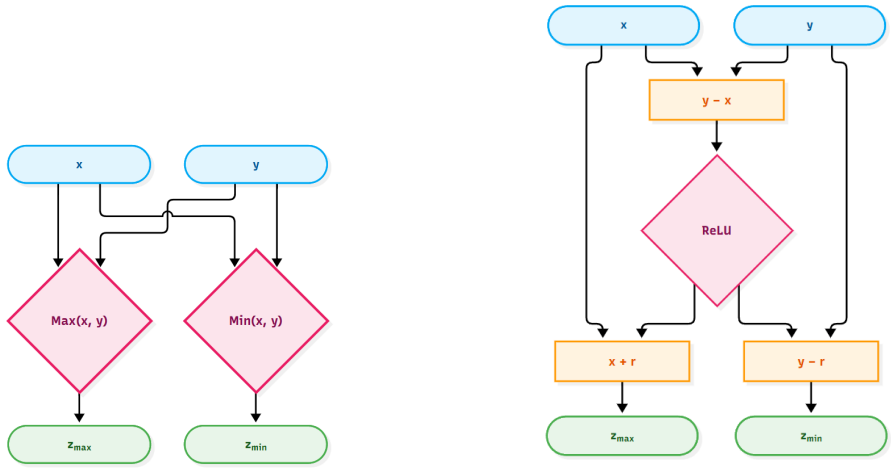
Theorem 4.2. *Let f be a Gradient Norm Preserving feedforward network ($L = 1$). Under the LiRPA framework (α-CROWN), there exist cases for these networks where the equivalent global bounding vector ω violates the necessary condition for tighter certification ($\|\omega\|_2 > 1$).*

The proof is detailed in Appendix B.2. Because Theorem 4.2 demonstrates that the standard formulation fundamentally fails to meet the prerequisite established in Theorem 4.1, we develop a graph factorization that unconditionally guarantees it

4.3 Graph Factorization for LiRPA Compatibility

To structurally guarantee that the bounding operator satisfies the necessary condition dictated by Theorem 4.1 ($\|\omega\|_2 \leq 1$), we propose a graph factorization of the GS2 activation. Our approach first exploits the fact that the sorting operation is inherently sum-preserving: $\min(x, y) + \max(x, y) = x + y$. By enforcing this conservation law by construction, we avoid independent bivariate relaxations. To transition from bivariate to univariate relaxation, we use a single ReLU operation to enforce delayed concretization. By forcing the verifier to evaluate the symbolic difference $(y - x)$ before the

non-linear relaxation is applied, we preserve the exact spatial correlation between adjacent neurons, yielding the computational graph illustrated in Figure 2.



(a) Standard `MaxMin` (b) Optimized GS2

Figure 2: Comparison of computational graphs for standard and factorized GS2 formulations.

The proposed Graph Factorization is formally defined as follows:

Definition 4.3 (Optimized GS2 Graph)**.** For an input pair $(x, y) \in \mathbb{R}^2$, the graph factorization enforcing delayed concretization and sum-preservation is defined by:

$$r = \text{ReLU}(y - x); \quad z_{\max} = x + r; \quad z_{\min} = y - r \tag{3}$$

yielding the exact bounded output $GS2(x, y) = (z_{\min}, z_{\max})$.

Theorem 4.4. *For a GNP network ($L = 1$), applying the optimized graph factorization guarantees that the equivalent LiRPA bounding operator satisfies the necessary condition ($\|\omega\|_2 \leq 1$).*

The proof is deferred to Appendix B.3. By factorizing the GS2 graph, the equivalent LiRPA bounding vector unconditionally satisfies the necessary condition. We empirically verify that LiRPA bounds with this reformulation consistently outperforms conventional GS2 bounds in Section 6.2.

Beyond enforcing this constraint, our reformulation replaces the two bivariate `Min` and `Max` non-linear nodes of conventional `GS2` parsing with a single univariate `ReLU` (Definition 4.3). Consequently, this cuts the total number of non-linear nodes in half. This memory reduction is useful to scale bound propagation, as validated empirically in Section 6.2. The shift to ReLU domain also resolves a critical compatibility barrier. Because bivariate activations introduce complex cross-terms, applying SDP relaxations to them is non-trivial (see the sketch proof in Section B.4). By reformulating GS2 into the univariate ReLU domain, we bypass this bottleneck and unlock the ability to use SDP-CROWN.

Although this graph factorization satisfies the necessary condition for tighter certification, the application of standalone LiRPA solvers is restricted to shallow architectures. To bypass this scalability bottleneck, the next section introduces a hybrid verification framework.

5 Scalable Certification via Hybrid Verification

To overcome the depth and memory limitations of backward LiRPA and enable the verification of deep architectures (e.g., VGG-like depths), we propose a sequential hybrid certification strategy. We decompose the overall network $f : \mathbb{R}^n \to \mathbb{R}^C$ into a L-Lipschitz feature-extracting backbone $g : \mathbb{R}^n \to \mathbb{R}^m$ and a shallow sub-network head $h : \mathbb{R}^m \to \mathbb{R}^C$, such that $f(x) = h(g(x))$. Our hybrid framework executes the certification process in two distinct phases (Figure 3), leveraging the L-Lipschitz bounds of the backbone to bypass the scalability limits of deep bound propagation.

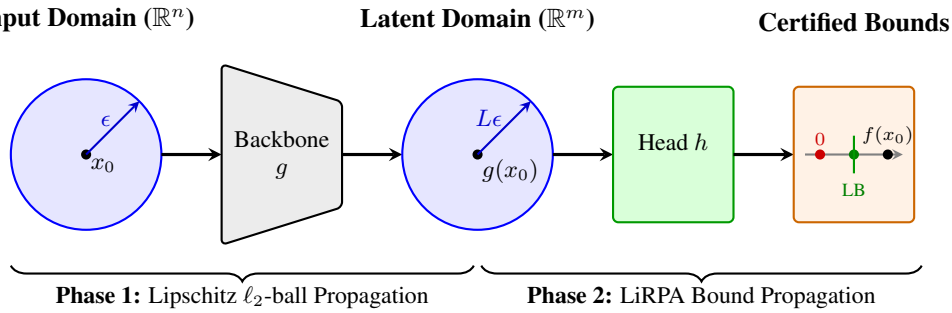


Figure 3: Sequential Hybrid Certification Pipeline with a L-Lipschitz backbone.

Phase 1: The Lipschitz Backbone. We utilize the early L-Lipschitz layers of the backbone g. Given an input perturbation $\|\delta\|_2 \leq \epsilon$, it is guaranteed that the latent representation remains bounded within an ℓ_2-ball of identical radius: $\|g(x_0 + \delta) - g(x_0)\|_2 \leq \epsilon$ (Phase 1 in Figure 3). This property allows us to propagate the perturbation through deep, high-dimensional feature extractors.

Phase 2: Localized Symbolic Handover. With the perturbation soundly encapsulated into an ℓ_2-ball in the latent space, we execute symbolic verification algorithms exclusively on the head h. State-of-the-art LiRPA verifiers are initialized exclusively on h, computing the final certified lower bound over the localized latent domain $\mathcal{Z} = \{z \in \mathbb{R}^m \mid \|z - g(x_0)\|_2 \leq \epsilon\}$ (Phase 2 in Figure 3).

We evaluate the practical trade-offs of this hybrid strategy in Section 6, identifying the optimal operational setup and presenting its empirical performance in Section 6.4.

6 Hyperparameter Dimensions Governing Certification Bounds

To the best of our knowledge, we present the first comprehensive evaluation of advanced formal verifiers on constrained 1-Lipschitz GNP-NNs. Beyond exploring the theoretical limitations of Lipschitz-certified radii and the effectiveness of our GS2 graph factorization, the main contribution of this section is identifying the key hyperparameters that drive the relative performance between Lipschitz-certified radii and formal verification. By mapping these dynamics, we establish the specific architectural constraints and optimization regimes in which each certification method excels, optimizing the deployment of our scalable hybrid framework.

6.1 Experimental Setup

We evaluate advanced formal verifiers on constrained 1-Lipschitz GNP-NNs on MNIST ($\epsilon = 0.3$), CIFAR-10 ($\epsilon = 8/255$), and Imagenette ($\epsilon = 8/255$). We utilize scalable base architectures ranging from MLPs to VGG-16, adapted directly from the standard verification literature (Chiu et al., 2025) for fair comparison. Models are orthogonalized using either Adaptive Orthogonal Convolutions (AOC) (Boissin et al., 2025) or Reshaped Kernel Orthogonalization (RKO) (Serrurier et al., 2021). We measure robustness across three core metrics for a target radius ϵ: Empirical Robust Accuracy (ERA) as an empirical upper bound via AutoAttack (Croce & Hein, 2020), analytical Certified Robust Accuracy (CRA) via the Lipschitz-certified radius (L is re-estimated using Power Iteration for RKO architectures), and Verified Robust Accuracy (VRA) derived via LiRPA bound propagation using our GS2 graph factorization (Theorem 4.3). Those metrics are evaluated over a standardized subset of 200 test points (Chiu et al., 2025) using α-CROWN and SDP-CROWN. Full reproducibility details—including exact network topologies, τ-cross entropy optimization grids, and verifier configurations—are deferred to Section C.5.

6.2 Enhancing formal verification bounds on GNP networks

To empirically validate the structural and computational advantages of our GS2 graph factorization (Section 4.3), we benchmark our formulation against conventional independent relaxations using α-CROWN and CROWN frameworks.

Table 1: **Impact of GS2 ReLU Factorization on CROWN and α-CROWN VRA for CIFAR-10.** All models use the robust training setting. MNIST results are different in Section D

		CROWN VRA (%)		α-CROWN VRA (%)	
Model	**Arch.**	**Factored**	**Native**	**Factored**	**Native**
CNN-A	GS2 RKO (Björck)	**65.5**	3.5	**65.5**	54.0
	GS2 AOC	50.5	0.0	**51.0**	16.0
ConvLarge	GS2 RKO (Björck)	**60.5**	0.0	**61.0**	OOM
	GS2 AOC	**35.0**	0.0	**35.0**	OOM

Verification Accuracy. As shown in Table 1, our GS2 graph factorization yields massive gains in Verified Robust Accuracy (VRA) across all evaluated settings. As demonstrated by CROWN verification, these gains stem directly from an increased tightness of the bounds within the underlying abstract domain. On the CNN-A architecture, factorization increase performance across both verifiers, driving CROWN VRA from 3.5% to 65.5% (RKO) and 0.0% to 50.5% (AOC), while simultaneously push α-CROWN VRA to 65.5% and 51.0%, respectively.

Empirical Memory Scalability. On the deeper ConvLarge architecture, conventional GS2 relaxations trigger Out-of-Memory (OOM) in α-CROWN. Because α-CROWN is a mandatory internal step for SDP-CROWN—the state-of-the-art solver for ℓ_2 robustness—this computational bottleneck prevents the deployment of advanced verifiers entirely. By halving the non-linear node count, our factorized representation resolves this issue. Beyond its theoretical guarantees, it successfully scales state-of-the-art tools to deeper networks, enabling them to run seamlessly and achieve up to 61.0% VRA under RKO and 35.0% under AOC.

6.3 Benchmark Results

Table 2: **Robustness under ℓ_2-norm perturbations on CIFAR-10 ($n = 3072, \epsilon = 8/255$).**. MNIST results can be found in Section D

			Upper Bound	Analytical	Formal Verification (VRA)	
Model (Setting)	**Architecture**	**Clean (%)**	**ERA (%)**	**CRA (%)**	**α-CROWN**	**SDP-CROWN**
CNN-A (Accurate)	GS2 RKO (Björck)	78.0	50.5	8.0	0.0	**18.0**
	GS2 AOC	68.0	40.5	**32.0**	0.0	30.5
CNN-A (Robust)	GS2 RKO (Björck)	72.5	68.5	66.5	65.5	**67.0**
	GS2 AOC	66.5	64.5	**64.0**	51.0	63.5
ConvLarge (Accurate)	GS2 RKO (Björck)	85.5	83.5	21.5	0.0	**25.0**
	GS2 AOC	78.0	73.5	61.5	0.0	**63.5**
ConvLarge (Robust)	GS2 RKO (Björck)	71.5	70.5	68.5	61.0	**69.0**
	GS2 AOC	71.0	70.0	**69.0**	35.0	68.5
ConvLarge (from Chiu et al. (2025))	Standard CNN	74	72.5	47.5	2.5	**63.5**

Table 7 demonstrates that GNP-NNs break the established taxonomy that SDP-CROWN outperforms Lipschitz-certified radius (Chiu et al., 2025). Not only do GNP architectures provide highly competitive native robustness (e.g., a 69.0% CRA for the ConvLarge AOC versus a 63.5% VRA for the standard unconstrained ConvLarge), but their Lipschitz-certified radii consistently surpass formal verification. Because AOC architectures operate close to the Eikonal condition, there is very little unproven robustness left for formal verifiers to explore; evidenced by the ConvLarge (Robust) AOC model yielding a 69.0% analytical CRA but dropping to a 68.5% VRA under SDP-CROWN. To ensure the statistical validity and generalization of these findings, Section E.4 details the variance of our training regimes and confirms our core conclusions on an other test subset. The execution times for the verification solvers are also reported in this appendix.

The optimal certification method is thus strictly dictated by the optimization regime and architectural constraints. In the **Robust regime** (low temperature), AOC architectures closely approximate the Eikonal ideal. This yields tight Lipschitz-certified radii that consistently match or beat formal verification without any computational overhead. Conversely, reaching the **Accurate regime** requires higher temperatures and relaxed structural constraints (RKO) to maximize network expressivity. Formal verifiers become essential in this regime, utilizing localized bound propagation to bypass the

global penalty and recover unproven robustness (e.g., SDP-CROWN successfully lifts the collapsed 8.0% CRA of the CNN-A Accurate RKO model to an 18.0% VRA).

Despite their necessity in these relaxed regimes, state-of-the-art backward solvers cannot be deployed directly on deep, high-capacity architectures like VGG due to the $\mathcal{O}(d_{out} \times d_{latent})$ memory bottleneck and depth recursion limits outlined in Section 2. This computational wall directly motivates our hybrid certification framework introduced in Section 6.4.

6.4 Scaling Verification via Hybridization

Motivated by overcoming the scalability barrier of standalone formal methods, we evaluate our hybrid verification framework on larger VGG architectures. Guided by the conclusions in Section 6.3, we maximize the framework's potential by applying the 1-Lipschitz constraint exclusively to the feature-extracting backbone. The classification head—specifically the final two linear layers—is left entirely unconstrained, utilizing standard ReLU and regularized via weight decay. For certification, we propagate Lipschitz ball through the backbone and use LiRPA on the unconstrained head.

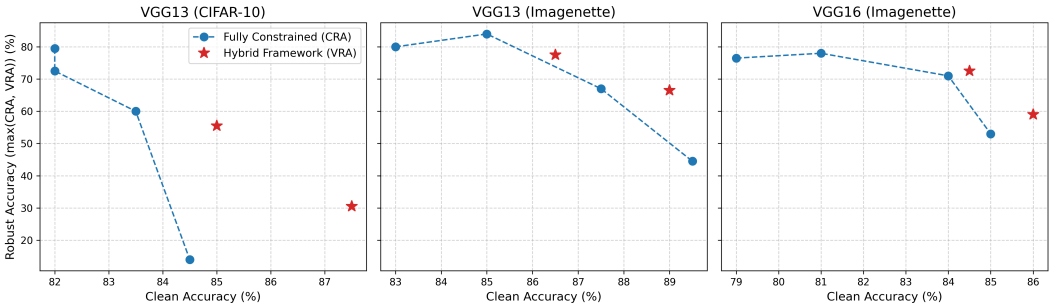


Figure 4: Pareto frontier of Clean Accuracy vs. Robust Accuracy ($\max(\text{CRA}, \text{VRA})$) on CIFAR-10 and Imagenette ($\epsilon = 8/255$). The full results detailing the performance of hybrid verification on both fully constrained and backbone-constrained models are deferred to Section E.4.

As illustrated in Figure 4, the hybrid verification successfully escapes the Pareto frontier that bottlenecks fully constrained architectures (Béthune et al., 2022). With an identical parameter count, the hybrid approach matches the clean accuracy of the "Accurate" baselines while significantly lifting certified robustness. For instance, the hybrid VGG16 achieves 84.5% clean accuracy and 72.5% VRA, whereas estimating its global Lipschitz constant yields a CRA of only 58.0% (Section E.4). This empirically proves the utility of formal methods in recovering unproven robustness at scale without succumbing to memory exhaustion.

7 Conclusion

In this work, we bridged the gap between constrained 1-Lipschitz architectures and post-hoc formal verification. We demonstrated that for GNP networks, Lipschitz-certified radii outperform state-of-the-art LiRPA solvers. However, formal methods remain essential for recovering the unproven robustness hidden within contractive or high-temperature optimization regimes. We enabled this recovery by introducing a LiRPA-compatible GS2 graph factorization. To bypass the memory bottlenecks of deep bound propagation, we also proposed a hybrid certification strategy. By using LiRPA on the latent space of a 1-Lipschitz backbone, we successfully scaled formal verification to VGG architectures. Ultimately, understanding how verifiers interact with 1-Lipschitz architectures establishes the baseline for future scaling endeavors. By synthesizing architectural constraints with localized verification, this hybrid framework paves the way to certify massive, state-of-the-art robust models—such as LiResNet and LipNeXt—that currently remain beyond the reach of LiRPA solvers.

AI USE STATEMENT

In this work, generative AI tools were utilized across writing, code development, and proof checking under strict author oversight. Specifically, AI tools were used to polish prose, correct grammatical errors, and assist with LaTeX formatting throughout the main manuscript and appendices. For software development, AI assistance was limited to boilerplate utilities and data logging scripts; all core algorithmic implementations—including our custom LiRPA graph factorization and hybrid verification framework—were human-written and rigorously tested. Additionally, AI tools were used as an auxiliary check for the structural clarity and soundness of the pre-existing, human-written mathematical proofs in Appendix A. Some appendix sections communicating hyperparameter configurations (Appendix C) were drafted with AI assistance, and all listed settings were manually cross-checked against our execution scripts and sweeps for strict fidelity. All core concepts, mathematical formulations, and novel ideas in this paper are solely the authors' work, with no AI system used for concept generation or ideation. The authors take full responsibility for the final content and scientific accuracy of this work.

REPRODUCIBILITY STATEMENT

To ensure full reproducibility, we provide complete mathematical proofs for Section 4 in Appendix A. Anonymized source code containing our custom LiRPA graph factorization, network architectures, and benchmark scripts are available at `https://anonymous.4open.science/r/robust-benchmark-DAF2/`. Full details regarding network architectures, hyperparameters, and compute infrastructure can be found in Appendix C.

ETHICS STATEMENT

This work focuses on the formal verification of deep neural networks to ensure deterministic safety bounds. By providing mathematical guarantees for model robustness, our research aims to improve the reliability and safety of AI systems deployed in critical environments. We do not foresee any direct negative societal impacts or ethical concerns arising from this theoretical framework.

ACKNOWLEDGMENTS

REFERENCES

Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International conference on machine learning*, pp. 291–301. PMLR, 2019.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pp. 274–283. PMLR, 2018.

Louis Béthune, Thibaut Boissin, Mathieu Serrurier, Franck Mamalet, Corentin Friedrich, and Alberto Gonzalez Sanz. Pay attention to your loss: understanding misconceptions about lipschitz neural networks. *Advances in Neural Information Processing Systems*, 35:20077–20091, 2022.

Åke Björck and Clazett Bowie. An iterative algorithm for computing the best estimate of an orthogonal matrix. *SIAM Journal on Numerical Analysis*, 8(2):358–364, 1971.

Thibaut Boissin, Franck Mamalet, Thomas Fel, Agustin Martin Picard, Thomas Massena, and Mathieu Serrurier. An adaptive orthogonal convolution scheme for efficient and flexible cnn architectures. In *International Conference on Machine Learning*, 2025.

Fabio Brau, Giulio Rossolini, Alessandro Biondi, and Giorgio Buttazzo. Robust-by-design classification via unitary-gradient neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 14264–14272, 2023.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. Ieee, 2017.

Hong-Ming Chiu, Hao Chen, Huan Zhang, and Richard Y Zhang. Sdp-crown: Efficient bound propagation for neural network verification with tightness of semidefinite programming. In *Forty-second International Conference on Machine Learning*, 2025.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

Noemie Cohen, Melanie Ducoffe, Timothee Fayard, Christophe Gabreau, Claire Pagetti, and Xavier Pucel. Certified geometric robustness - Super-Deepg . In *2026 ACM/IEEE 17th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 242–252, May 2026.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

Aaron Defazio, Xingyu Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The road less scheduled. *Advances in Neural Information Processing Systems*, 37: 9974–10007, 2024.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015. URL `https://arxiv.org/abs/1412.6572`.

Sven Gowal, Krishnamurthy Dj Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. Scalable verified training for provably robust image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Kai Hu, Andy Zou, Zifan Wang, Klas Leino, and Matt Fredrikson. Unlocking deterministic robustness certification on imagenet. *Advances in Neural Information Processing Systems*, 2023.

Kai Hu, Klas Leino, Zifan Wang, and Matt Fredrikson. A recipe for improved certifiable robustness. In *International Conference on Learning Representations*, volume 2024, pp. 39506–39517, 2024.

Kai Hu, Haoqi Hu, and Matt Fredrikson. Lipnext: Scaling up lipschitz-based certified robustness to billion-parameter models. *arXiv preprint arXiv:2601.18513*, 2026.

Yujia Huang, Huan Zhang, Yuanyuan Shi, J Zico Kolter, and Anima Anandkumar. Training certifiably robust neural networks with efficient local lipschitz bounds. *Advances in Neural Information Processing Systems*, 34:22745–22757, 2021.

Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In Rupak Majumdar and Viktor Kunčak (eds.), *Computer Aided Verification*, pp. 97–117, Cham, 2017. Springer International Publishing. ISBN 978-3-319-63387-9.

Bo-Han Lai, Pin-Han Huang, Bo-Han Kung, and Shang-Tse Chen. Enhancing certified robustness via block reflector orthogonal layers and logit annealing loss. *arXiv preprint arXiv:2505.15174*, 2025.

Klas Leino, Zifan Wang, and Matt Fredrikson. Globally-robust neural networks. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021. doi: 10.48550/arXiv.2102.08452.

Qiyang Li, Saminul Haque, Cem Anil, James Lucas, Roger B Grosse, and Jörn-Henrik Jacobsen. Preventing gradient attenuation in lipschitz constrained convolutional networks. *Advances in neural information processing systems*, 32, 2019.

Hadi Salman, Greg Yang, Huan Zhang, Cho-Jui Hsieh, and Pengchuan Zhang. A convex relaxation barrier to tight robustness verification of neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Mathieu Serrurier, Franck Mamalet, Alberto González-Sanz, Thibaut Boissin, Jean-Michel Loubes, and Eustasio del Barrio. Achieving robustness in classification using optimal transport with hinge regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 505–514, 2021. Reference for deel-lip library.

Zhouxing Shi, Yihan Wang, Huan Zhang, J Zico Kolter, and Cho-Jui Hsieh. Efficiently computing local lipschitz constants of neural networks via bound propagation. *Advances in Neural Information Processing Systems*, 35:2350–2364, 2022.

Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. In *Proceedings of the ACM on Programming Languages (POPL)*, volume 3, pp. 1–30, 2019.

Sahil Singla and Soheil Feizi. Skew orthogonal convolutions. In *International Conference on Machine Learning (ICML)*, pp. 9756–9766. PMLR, 2021.

Jiahao Su, Wonmin Byeon, and Furong Huang. Scaling-up diverse orthogonal convolutional networks by a paraunitary framework. In *International Conference on Machine Learning*, pp. 20546–20579. PMLR, 2022.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Vincent Tjeng, Kai Y. Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. In *International Conference on Learning Representations*, 2019.

Asher Trockman and J Zico Kolter. Orthogonalizing convolutional layers with the cayley transform. In *International Conference on Learning Representations*, 2021.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. *Advances in neural information processing systems*, 31, 2018.

Caterina Urban and Antoine Miné. A review of formal methods applied to machine learning, 2021. URL `https://arxiv.org/abs/2104.02466`.

Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and Zico Kolter. Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

Lily Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. Towards fast computation of certified robustness for relu networks. In *International Conference on Machine Learning*, pp. 5276–5285. PMLR, 2018.

Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems*, 33:1129–1141, 2020.

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 *International Conference on Learning Representations*, 2021.

Xiaojun Xu, Linyi Li, and Bo Li. Lot: Layer-wise orthogonal training on improving l2 certified robustness. In *Advances in Neural Information Processing Systems*, volume 35, pp. 18904–18915, 2022.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Huan Zhang, Pengchuan Zhang, and Cho-Jui Hsieh. Recurjac: An efficient recursive algorithm for bounding jacobian matrix of neural networks and its applications. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5757–5764, 2019.

A PROOFS AND COUNTEREXAMPLES FOR SECTION 3

A.1 THE SIGNED DISTANCE FUNCTION (SDF)

To achieve perfectly tight analytical certificates where $\Delta\epsilon(x) = 0$, a 1-Lipschitz neural network must learn the exact metric space of the classification problem. Following Béthune et al. (2022), we define this ideal target formally.

Definition A.1 (Signed Distance Function)**.** Given a binary classifier $c : \mathbb{R}^n \to \{-1, +1\}$ and its associated decision frontier $\mathcal{B}$, the Signed Distance Function (SDF) is defined for all $x \in \mathbb{R}^n$ as:

$$f^*(x) = c(x) \inf_{x' \in \mathcal{B}} ||x - x'||_2$$

When the network perfectly approximates this function, the certified robustness perfectly matches the true robustness. Because $f^*(x)$ represents physical Euclidean distance, its gradient norm must be exactly 1 almost everywhere (the Eikonal condition).

A.2 VISUAL INTUITION: THE OPTIMIZATION GAP

Even if an architecture is mathematically capable of satisfying the Eikonal condition, optimization dynamics frequently prevent the network from converging to the true SDF. To illustrate the origins of the estimation gap $\Delta\epsilon$, Figure 5 provides a 1D toy visualization.

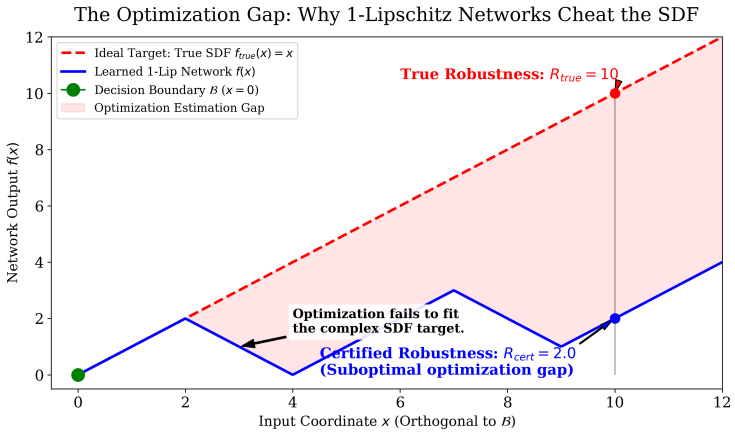


Figure 5: 1D illustration of the optimization gap. To satisfy loss margins without exceeding the strict 1-Lipschitz global constraint, the network (blue line) folds heavily. While the GNP constraint, the function fails to capture the true monotonic distance of the SDF (red dashed line), resulting in a pessimistic certified radius ($\epsilon_{lip} \ll \epsilon_{true}$).

A.3 PROOF OF PROPOSITION 3.1 (EIKONAL DESTRUCTION)

Proof of Proposition 3.1 (Eikonal Destruction). Consider a scalar-valued neural network $f : \mathbb{R}^{d_0} \to \mathbb{R}$ composed of k layers:

$$f(x) = (h_k \circ \sigma \circ h_{k-1} \circ \sigma \cdots \circ \sigma \circ h_1)(x)$$

where $h_i(z) = W_i z + b_i$ is an affine transformation with weight matrix $W_i \in \mathbb{R}^{d_i \times d_{i-1}}$, and σ is a non-linear activation function.

By the multivariate chain rule, the derivative of the output with respect to the input is the product of the layer-wise Jacobians:

$$J_f = W_k D_{k-1} W_{k-1} \dots D_1 W_1$$

where $D_i = J_\sigma(z_i)$ is the Jacobian of the activation function at layer i. Because the network output $f(x)$ is a scalar, this derivative J_f is naturally a row vector.

However, by standard convention, the gradient $\nabla f(x)$ is defined as a column vector matching the dimensions of the input. Therefore, the gradient is the transpose of the derivative Jacobian:

$$\nabla f(x)^T = W_k D_{k-1} W_{k-1} \dots D_1 W_1$$

Applying the transpose operator to the matrix product reverses the order of multiplication, yielding the standard backpropagation formula:

$$\nabla f(x) = W_1^T D_1^T \dots W_{k-1}^T D_{k-1}^T W_k^T$$

This sequence strictly necessitates that the backward operator mapping the downstream gradient to the upstream gradient for any given linear layer i is left-multiplication by its transpose W_i^T.

Let $W \in \mathbb{R}^{d_{out} \times d_{in}}$ be an expanding layer ($d_{out} > d_{in}$) constrained to have orthonormal columns such that $W^T W = I_{d_{in}}$.

By the Fundamental Theorem of Linear Algebra, the domain $\mathbb{R}^{d_{out}}$ of the backward operator W^T can be orthogonally decomposed into $\mathbb{R}^{d_{out}} = \text{im}(W) \oplus \ker(W^T)$. Any incoming gradient $g_{out} \in \mathbb{R}^{d_{out}}$ from the layer above can be uniquely decomposed into $g_{out} = g_{\parallel} + g_{\perp}$, where $g_{\parallel} \in \text{im}(W)$ and $g_{\perp} \in \ker(W^T)$.

Applying the backward pass evaluates to:

$$g_{in} = W^T(g_{\parallel} + g_{\perp}) = W^T g_{\parallel} + W^T g_{\perp} = W^T g_{\parallel} + 0$$

Because $g_{\parallel} \in \text{im}(W)$, there exists a vector $v \in \mathbb{R}^{d_{in}}$ such that $g_{\parallel} = Wv$. Therefore, $g_{in} = W^T W v = v$. Since $W^T W = I_{d_{in}}$, W perfectly preserves the ℓ_2-norm on its image, meaning $||g_{\parallel}||_2 = ||Wv||_2 = ||v||_2 = ||g_{in}||_2$.

By the Pythagorean theorem:

$$||g_{in}||_2^2 = ||g_{out}||_2^2 - ||g_{\perp}||_2^2$$

Because $d_{out} > d_{in}$, $\ker(W^T)$ is non-trivial. If g_{out} possesses any non-zero component orthogonal to the range of W ($g_{\perp} \neq 0$), that component is annihilated by the null space, and the overall gradient norm is strictly attenuated ($||g_{in}||_2 < ||g_{out}||_2$). ■ □

A.4 Counterexample: Non-Linearity Structurally Prevents Global Eikonal Convergence

To prove that the non-linearity structurally prevents global Eikonal convergence across the domain, we demonstrate that no single valid weight matrix can satisfy the Eikonal condition for all inputs without violating the network's global 1-Lipschitz constraint.

Let $W \in \mathbb{R}^{2 \times 1}$ be an expanding layer defined as $W = (1, 0)^T$. Observe that $W^T W = 1$, satisfying the strict isometry constraint. Let the activation function be the Gradient Norm Preserving (GNP) GroupSort2 Anil et al. (2019), where $GS2(a, b) = (\min(a, b), \max(a, b))$.

Let $W_{head} \in \mathbb{R}^{1 \times 2}$ be an arbitrary classification head defined as $W_{head} = (w_1, w_2)$. To maintain the global 1-Lipschitz property of the network, the spectral norm of this linear layer must not exceed 1 ($\|W_{head}\|_2 \leq 1$). For a row vector, this imposes the strict co-isometry constraint:

$$w_1^2 + w_2^2 \leq 1$$

Assume a scalar input $x \in \mathbb{R}$. The forward linear transformation yields $z = Wx = (x, 0)^T$. We now evaluate the gradient for both the positive and negative domains of x:

Case 1: $x > 0$

Because $x > 0$, the activation layer sorts z, resulting in $h = GS2(x, 0) = (0, x)^T$. The final scalar logit output is:

$$\hat{y} = W_{head} h = w_1(0) + w_2(x) = w_2 x$$

The input gradient is therefore $\nabla_x \hat{y} = w_2$. To satisfy the Eikonal condition ($||\nabla_x \hat{y}||_2 = 1$) in this region, we strictly require $|w_2| = 1$, which implies:

$$w_2^2 = 1$$

Case 2: $x < 0$

Because $x < 0$, the activation layer sorts z differently, resulting in $h = GS2(x, 0) = (x, 0)^T$. The final scalar logit output is:

$$\hat{y} = W_{head}h = w_1(x) + w_2(0) = w_1 x$$

The input gradient is therefore $\nabla_x \hat{y} = w_1$. To satisfy the Eikonal condition in this region, we strictly require $|w_1| = 1$, which implies:

$$w_1^2 = 1$$

Conclusion

To satisfy the Eikonal condition globally across the entire domain, the classification head must simultaneously satisfy $w_1^2 = 1$ (for $x < 0$) and $w_2^2 = 1$ (for $x > 0$). This implies:

$$w_1^2 + w_2^2 = 1 + 1 = 2$$

However, this strictly violates the fundamental 1-Lipschitz co-isometry constraint of the classification head, which legally mandates that $w_1^2 + w_2^2 \leq 1$. Because we have a mathematical contradiction ($2 \leq 1$), it is impossible to formulate a classification head that is both 1-Lipschitz and globally Eikonal. This mathematically proves that strict Eikonal destruction is structurally unavoidable across the domain due to the non-linear sorting operation.

Because this strict contradiction occurs in the minimal 1D-to-2D expansion, it serves as a definitive counterexample proving that orthogonal upsampling layers combined with sorting non-linearities structurally prevent global Eikonal convergence in the general case.

A.5 Empirical Observation of the Certification Gap

To explicitly quantify the impact of the theoretical bottlenecks formalized in Section 3, we measure the divergence between the empirical upper bound of robustness (ERA) and the analytical lower bound (CRA) on our standard benchmark models.

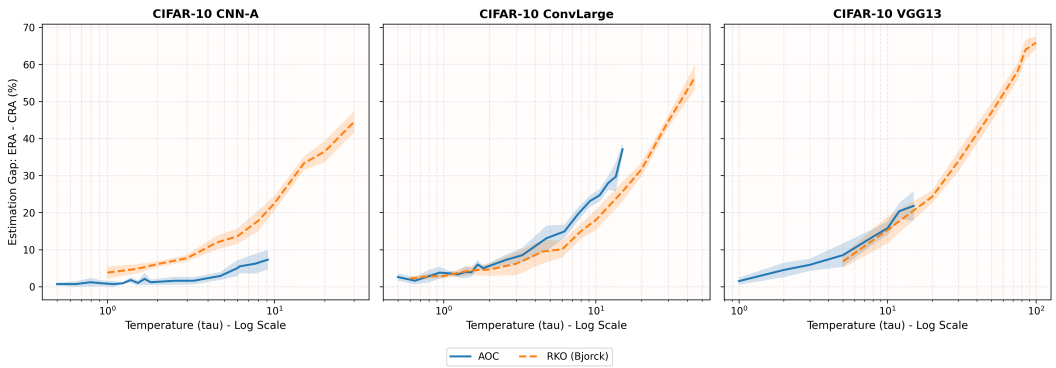


Figure 6: Comparison of the empirical robustness upper bound versus the analytical certified lower bound across different training temperatures.

Consistent with our theoretical analysis, the certification gap grows with the **training temperature**. As temperature increases, the tightness of the Lipschitz-certified radii systematically degrades. The magnitude of this gap also depends heavily on the chosen **orthogonalization method**. Models utilizing **AOC** maintain a remarkably tight certification gap; their strict layer-wise orthogonality strongly preserves gradient norms, leaving very little unproven robustness. Conversely, **RKO architectures** enforce a relaxed, contractive bound (singular values ≤ 1). At high temperatures, this flexibility pushes the network's local Lipschitz constant below 1, creating a significant estimation gap. Ultimately, high temperatures combined with contractive RKO architectures leave substantial unproven robustness for formal verifiers to explore, a divergence that directly motivates our benchmark in Section 6.3.

B PROOFS FOR ADAPTING FORMAL VERIFICATION (SECTION 4)

B.1 PROOF OF THEOREM 4.1

Proof of Theorem 4.1. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a binary classifier with a global ℓ_2 Lipschitz constant L. For a positively classified input x_0, the Lipschitz robustness radius simplifies to $\varepsilon_{\text{lip}} = \frac{f(x_0)}{L}$ (Tsuzuku et al., 2018). LiRPA certifies robustness by computing a linear lower bound $f(x) \geq \omega^\top x + b$ valid within a target radius $\varepsilon_{\text{LiRPA}}$. By evaluating the worst-case minimum of this bounding plane, we establish a necessary condition for LiRPA to yield a tighter certificate than the Lipschitz-certified radius.

By the definition of the LiRPA lower bound evaluated at the nominal point x_0, we have $f(x_0) \geq \omega^\top x_0 + b$. Using the assumption that robustness is certified and the certified radius matches or exceeds the Lipschitz radius ($\varepsilon_{\text{LiRPA}} \geq \varepsilon_{\text{lip}}$), we expand both expressions:

$$\frac{\omega^\top x_0 + b}{\|\omega\|_2} \geq \varepsilon_{\text{LiRPA}} \geq \varepsilon_{\text{lip}} \tag{4}$$

Substituting the theoretical radius $\varepsilon_{\text{lip}} = \frac{f(x_0)}{L}$ and the lower bound $f(x_0)$ into the inequality yields:

$$\frac{f(x_0)}{\|\omega\|_2} \geq \frac{\omega^\top x_0 + b}{\|\omega\|_2} \geq \frac{f(x_0)}{L} \tag{5}$$

Given that x_0 is positively classified ($f(x_0) > 0$), dividing the outer inequality by $f(x_0)$ yields $\frac{1}{\|\omega\|_2} \geq \frac{1}{L}$, which simplifies directly to $\|\omega\|_2 \leq L$. □

This condition is not sufficient. Consider the binary classifier $f(x) = \sigma(x+1) + \sigma(-x+1)$, where σ is the ReLU activation function. The network is globally 1-Lipschitz ($L = 1$). Evaluated at the nominal point $x_0 = 0$, the network outputs $f(0) = 2$, yielding a theoretical Lipschitz robustness radius of $\varepsilon_{\text{lip}} = 2/1 = 2$.

Using standard linear relaxations evaluated over the interval $x \in [-2, 2]$, we assign lower bound slopes $\alpha_1 = 1$ (for $x + 1$) and $\alpha_2 = 0$ (for $-x + 1$). This yields the global lower bound equation $f(x) \geq x + 1$, providing equivalent parameters $\omega = 1, b = 1$.

While the necessary condition from Theorem 4.1 is successfully met ($\|\omega\|_2 = 1 \leq L$), evaluating the actual certified radius for this bound yields $\varepsilon_{\text{LiRPA}} = \frac{1(0)+1}{1} = 1$. Since $1 < \varepsilon_{\text{lip}}$, the necessary condition is necessary, but not sufficient. □

B.2 PROOF OF THEOREM 4.2

In Linear Relaxation-based Perturbation Analysis (LiRPA), linear lower bounds ($f(x) \geq \omega^\top x + b$) are computed by propagating symbolic linear equations backward through the computational graph. For an M-layer network, this backward propagation accumulates linear transformations across layers, resulting in an equivalent global bounding weight vector expressed as the product:

$$\omega^\top = \prod_{\ell=M}^{1} D_\ell W_\ell D_{\sigma_\ell} A_\ell \tag{6}$$

where W_ℓ denotes the weight matrices, D_ℓ and D_{σ_ℓ} represent diagonal selection matrices capturing linear operation and activation status, and A_ℓ is a block-diagonal matrix encoding the layer's local linear relaxation operators.

Proof of Theorem 4.2. To evaluate the norm of the overall equivalent weight vector $\omega^\top = \prod_{\ell=M}^{1} D_\ell W_\ell D_{\sigma_\ell} A_\ell$, we analyze the spectral properties of the local bounding operator A_ℓ. Because A_ℓ is block-diagonal, its spectral norm is dictated by its maximum block: $\|A_\ell\|_2 = \max_i \|A_{ii}\|_2$.

For `GroupSort2` networks, each A_{ii} is a 2×2 matrix mapping a pair of pre-activation inputs (x, y) to their sorted outputs $(z_{\min}, z_{\max})$. Let $[l_x, u_x]$ and $[l_y, u_y]$ denote the concrete bounding intervals over the perturbation region. If these intervals overlap, we are in a non-linear regime.

We construct the equivalent local bounding operator A_{ii} by extracting the exact upper bounding hyperplanes and optimization parameters implemented in the α-CROWN algorithm via the `auto_LiRPA` library. The bounding plane for $z_{\max}$ utilizes parameter $\alpha_u \in (0, 1]$, and the symmetrical bounding plane for $z_{\min}$ utilizes parameter $\alpha_l \in [0, 1]$. Demonstrating cases where these hyperplanes mathematically force the spectral norm above 1 is sufficient to prove global norm inflation.

For a general local operator $A_{ii} = \begin{pmatrix} \alpha_l & 1-\alpha_l \\ a & b \end{pmatrix}$, evaluating the characteristic polynomial $P(\lambda) = \det(A_{ii}^\top A_{ii} - \lambda I)$ exactly at $\lambda = 1$ yields:

$$P(1) = 2\alpha_l(1-\alpha_l) - \left[a^2(2\alpha_l - \alpha_l^2) + b^2(1-\alpha_l^2) + 2ab\alpha_l(1-\alpha_l)\right] \tag{7}$$

Because $P(\lambda)$ is an upward-opening parabola, $P(1) < 0$ mathematically guarantees a root $\lambda_{\max} > 1$, resulting in $\|A_{ii}\|_2 > 1$.

Case 1: Shifted Intervals. Without loss of generality, assume the intervals overlap but are strictly shifted: $l_y < l_x < u_y < u_x$. For $z_{\max}$, the `auto_LiRPA` upper bound yields the slope row $(a \quad b)$, where:

$$a = \frac{u_y - u_x}{\alpha_u(l_x - u_x)} \quad \text{and} \quad b = \frac{(\alpha_u - 1)(u_y - u_x)}{\alpha_u(u_y - l_y)} \tag{8}$$

For $z_{\min}$, the upper bound yields the slope row $(\alpha_l \quad 1-\alpha_l)$. **Counterexample:** Let the input intervals be $[l_x, u_x] = [1, 3]$ and $[l_y, u_y] = [0, 2]$. Let the optimizer select $\alpha_u = 1$ and $\alpha_l = 1$. This yields $a = 0.5$ and $b = 0$. Substituting these values into Equation 7 evaluates to:

$$P(1) = 0 - \left[(0.5)^2(1) + 0 + 0\right] = -0.25 \tag{9}$$

Because $P(1) < 0$, we have $\|A_{ii}\|_2 > 1$.

Case 2: One-Sided Ties. Without loss of generality, assume $l_y < l_x < u_x = u_y$. For $z_{\max}$, the upper bound yields the slope row $(a \quad b)$, where:

$$a = \alpha_u \frac{u_x - \max(l_x, l_y)}{u_x - l_x} \quad \text{and} \quad b = \alpha_u \frac{u_x - \max(l_x, l_y)}{u_y - l_y} \tag{10}$$

For $z_{\min}$, the upper bound yields the slope row $(\alpha_l \quad 1-\alpha_l)$. **Counterexample:** Let $[l_x, u_x] = [1, 2]$ and $[l_y, u_y] = [0, 2]$. Let the optimizer select $\alpha_u = 1$ and $\alpha_l = 1$. This yields $a = 1$ and $b = 0.5$. Substituting into Equation 7 gives:

$$P(1) = 0 - \left[(1)^2(1) + 0 + 0\right] = -1 \tag{11}$$

Because $P(1) < 0$, we have $\|A_{ii}\|_2 > 1$.

Case 3: Strict Containment. Without loss of generality, assume $l_y < l_x < u_x < u_y$. For $z_{\max}$, `auto_LiRPA` symmetrically yields the slope row $(b \quad a)$, where:

$$b = \frac{(1-\alpha_u)(u_y - u_x)}{\alpha_u(u_x - l_x)} \quad \text{and} \quad a = \frac{u_y - u_x}{\alpha_u(u_y - l_y)} \tag{12}$$

For $z_{\min}$, the slope row remains $(\alpha_l \quad 1-\alpha_l)$. Constructing the local operator gives $A_{ii} = \begin{pmatrix} \alpha_l & 1-\alpha_l \\ b & a \end{pmatrix}$, which inverts the positions of a and b in Equation 7. **Counterexample:** Let $[l_x, u_x] = [1, 2]$ and $[l_y, u_y] = [0, 3]$. Let the optimizer select $\alpha_u = 0.5$ and $\alpha_l = 0$. This yields $a = 2/3$. Substituting into the adjusted polynomial gives:

$$P(1) = 0 - \left[b^2(0) + (2/3)^2(1) + 0\right] = -4/9 \tag{13}$$

Because $P(1) < 0$, we have $\|A_{ii}\|_2 > 1$.

Conclusion: As demonstrated by the exact counterexamples across all unstable overlapping regimes, there exist valid spatial bounding states under the α-CROWN formulation where $P(1) < 0$. Therefore, the equivalent global bounding vector fails to guarantee the necessary condition ($\|\omega\|_2 \leq 1$). $\square$

B.3 Proof of Theorem 4.4 (Optimized Factorization Isometry)

Proof of Theorem 4.4. Under our optimized graph factorization, the sorted outputs are computed as:

$$z_{\min} = y - \text{ReLU}(y - x), \quad z_{\max} = x + \text{ReLU}(y - x) \tag{14}$$

In linear relaxation frameworks, the non-linear shared node $r = \text{ReLU}(y - x)$ is bounded by linear functions of its input. Let $\gamma \in [0, 1]$ denote the effective relaxation slope dynamically chosen by the framework for this ReLU, and let the relaxation intercepts be absorbed into constant bias terms.

Without loss of generality, we analyze the equivalent linear upper bound coefficients (the lower bound follows an identical symmetric structure). Applying the linear relaxation directly to the forward equations yields:

$$z_{\min} \leq y - \gamma(y - x) + \beta_{\min} \tag{15}$$
$$z_{\max} \leq x + \gamma(y - x) + \beta_{\max} \tag{16}$$

Distributing the slope γ and regrouping the terms by the pre-activation variables x and y, we obtain:

$$z_{\min} \leq \gamma x + (1 - \gamma)y + \beta_{\min} \tag{17}$$
$$z_{\max} \leq (1 - \gamma)x + \gamma y + \beta_{\max} \tag{18}$$

We extract the local bounding operator A_{ii}, mapping the inputs (x, y) to the bounded outputs $(z_{\min}, z_{\max})$:

$$\begin{pmatrix} z_{\min} \\ z_{\max} \end{pmatrix} \leq \underbrace{\begin{pmatrix} \gamma & 1-\gamma \\ 1-\gamma & \gamma \end{pmatrix}}_{A_{ii}} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \beta_{\min} \\ \beta_{\max} \end{pmatrix} \tag{19}$$

To determine the spectral norm of A_{ii}, we compute its eigenvalues via the characteristic polynomial $\det(A_{ii} - \lambda I) = 0$:

$$(\gamma - \lambda)^2 - (1 - \gamma)^2 = 0 \tag{20}$$

Solving this difference of squares yields exactly two roots:

$$\lambda_1 = \gamma + (1 - \gamma) = 1, \quad \lambda_2 = \gamma - (1 - \gamma) = 2\gamma - 1 \tag{21}$$

Because verification frameworks mathematically restrict the scalar ReLU relaxation slope such that $0 \leq \gamma \leq 1$ across all geometric regimes, the second eigenvalue is strictly bounded by $-1 \leq 2\gamma - 1 \leq 1$.

Since A_{ii} is perfectly symmetric, its spectral norm is simply its maximum absolute eigenvalue ($\|A_{ii}\|_2 = \max\{|\lambda_1|, |\lambda_2|\}$). With the maximum absolute eigenvalue rigidly locked at 1 regardless of the chosen relaxation slope γ, the spectral norm unconditionally evaluates exactly to $\|A_{ii}\|_2 = 1$. □

B.4 SDP-CROWN Compatibility and Computational Complexity

Directly adapting the SDP-CROWN algorithm to the native `min` and `max` formulation of `GroupSort2` is computationally prohibitive because the solver is fundamentally engineered for univariate functions. Its efficiency relies entirely on an closed-form solution for standard ReLU relaxations, a mathematical property that bivariate operations inherently lack.

We can reformulate a standard ReLU activation $y = \max(0, x)$ as a simple quadratic constraint: $y(y - x) = 0$. Lifting this into a Semi-Definite Programming (SDP) framework yields a dual problem characterized by 2×2 matrices. This low-dimensional system admits an analytical, closed-form solution via its characteristic polynomial, allowing the verifier to compute bounds instantly.

Conversely, the primary component of the `GroupSort2` activation, $z = \max(x, y)$, is governed by a significantly more complex quadratic constraint:

$$z^2 - zx - zy + xy = 0$$

This constraint introduces cross-terms (zx, zy, xy) that intimately couple all three variables. Unlike the decoupled ReLU, relaxing the native $\max(x, y)$ constraint requires forming an SDP matrix over the basis $[1, x, y, z]$. Consequently, the associated dual problem translates into a system of 4×4 Positive Semi-Definite (PSD) constraints.

Unlike 2×2 matrices, 4×4 PSD systems do not generally admit closed-form analytical solutions for their dual variables. Any verifier attempting to bound a native $\max$ primitive directly within an SDP framework must therefore rely on iterative, numerical interior-point solvers. This introduces a massive computational bottleneck, rendering deep network verification intractable.

Therefore, our translation into the ReLU domain ($z = x + \text{ReLU}(y - x)$) is not just geometrically necessary to preserve 1-Lipschitz constraint (as proven in Theorem 4.4), but acts as the fundamental bridge required to preserve the $\mathcal{O}(1)$ scalability of state-of-the-art ℓ_2 verifiers like SDP-CROWN.

C Experimental Details and Hyperparameters

To ensure full reproducibility of our results, this section provides a comprehensive breakdown of our network topologies, optimization hyperparameters, data augmentation pipelines, and verification setups.

C.1 Detailed Network Architectures

As introduced in the main text, our base network topologies are adapted from the standard verification literature, specifically SDP-CROWN Chiu et al. (2025). To enforce a strict 1-Lipschitz constant across the entire network, we replace standard components with their isometric equivalents:

- **Linear and Convolutional Layers:** Replaced with Adaptive Orthogonal Convolutions (AOC) Boissin et al. (2025) and RKO.
- **Activations:** We utilize GroupSort2 Anil et al. (2019) reformulated as our proposed SDP-CROWN safe reformulation.
- **Pooling:** Standard max or average pooling operations are entirely omitted, as they complicate Lipschitz bound calculations. All spatial downsampling is achieved using strided orthogonal convolutions.

Table 3 details the base structural shapes and capacities for all networks tested in our empirical benchmarks.

C.2 Training Protocol and Hyperparameter Optimization

All models were trained using the Schedule-Free optimizer Defazio et al. (2024). For our primary results, we performed a grid search over the learning rate and the temperature parameter τ of the τ-cross entropy loss : $\mathcal{L}_\tau(f(x), y) = \mathcal{L}_{CE}(\tau \cdot f(x), y)$Béthune et al. (2022). Models were selected based on their performance on a validation split comprising 10% of the training data. We specifically isolated two optimal checkpoints per architecture: one maximizing clean accuracy and one maximizing Certified Robust Accuracy (CRA). We denote by τ_{min} and τ_{max} the values associated with the robust and the accurate model. We present the selected temperature in Table 4.

For our empirical ERA/CRA gap analysis in Figure 6, 10 to 20 (depending on the model) τ values were sampled between empirically established thresholds τ_{min} and τ_{max} (where the network maintains stable convergence).

C.3 Data Augmentation Strategies

Standardizing data augmentation is critical when evaluating certifiable robustness, as heavy regularizations can shift the clean vs. robust accuracy trade-off. To ensure standardized inputs across all experiments, CIFAR-10 images were normalized using a mean of $(0.4914, 0.4822, 0.4465)$ and a standard deviation of 0.225 across all channels. MNIST images were converted to tensors without additional standardization. For Imagenette, images were normalized using a standard ImageNet mean of $(0.485, 0.456, 0.406)$ and a uniform standard deviation of 0.225 across all channels.

Table 3: Base architectures used for empirical benchmarks on MNIST, CIFAR-10, and Imagenette.

Model name	**Model structure**
MLP (MNIST)	Linear(784, 100) - Linear(100, 100) - Linear(100, 10)
ConvSmall (MNIST)	Conv(1, 16, 4, 2, 1) - Conv(16, 32, 4, 2, 1) - Linear(1568, 100) - Linear(100, 10)
CNN-A (CIFAR-10)	Conv(3, 16, 4, 2, 1) - Conv(16, 32, 4, 2, 1) - Linear(2048, 100) - Linear(100, 10)
ConvLarge (CIFAR-10)	Conv(3, 32, 3, 1, 1) - Conv(32, 32, 4, 2, 1) - Conv(32, 64, 3, 1, 1) - Conv(64, 64, 4, 2, 1) - Linear(4096, 512) - Linear(512, 512) - Linear(512, 10)
VGG-13 (CIFAR-10)	Conv(3, 64, 3, 1, 1) - Conv(64, 64, 3, 1, 1) - Conv(64, 64, 3, 2, 1) - Conv(64, 128, 3, 1, 1) - Conv(128, 128, 3, 1, 1) - Conv(128, 128, 3, 2, 1) - Conv(128, 256, 3, 1, 1) - Conv(256, 256, 3, 1, 1) - Conv(256, 256, 3, 2, 1) - Linear(4096, 512) - Linear(512, 512) - Linear(512, 10)
VGG-13 (Imagenette)	Conv(3, 64, 3, 1, 1) - Conv(64, 64, 3, 2, 1) - Conv(64, 128, 3, 1, 1) - Conv(128, 128, 3, 2, 1) - Conv(128, 256, 3, 1, 1) - Conv(256, 256, 3, 2, 1) - Conv(256, 512, 3, 1, 1) - Conv(512, 512, 3, 2, 1) - Conv(512, 512, 3, 1, 1) - Conv(512, 512, 3, 2, 1) - Linear(25088, 512) - Linear(512, 512) - Linear(512, 10)
VGG-16 (Imagenette)	[Conv(3, 64, 3, 1, 1) $\times 2$ - Conv(64, 64, 3, 2, 1)] - [Conv(64, 128, 3, 1, 1) $\times 2$ - Conv(128, 128, 3, 2, 1)] - [Conv(128, 256, 3, 1, 1) $\times 3$ - Conv(256, 256, 3, 2, 1)] - [Conv(256, 512, 3, 1, 1) $\times 3$ - Conv(512, 512, 3, 2, 1)] - [Conv(512, 512, 3, 1, 1) $\times 3$ - Conv(512, 512, 3, 2, 1)] - Linear(25088, 512) - Linear(512, 512) - Linear(512, 10)

Note: `Conv(3, 16, 4, 2, 0)` stands for a convolutional layer with 3 input channels, 16 output channels, a 4×4 kernel, stride 2, and padding 0. Brackets [] in VGG-16 denote repeated functional blocks. `Linear(1568, 100)` represents a fully connected layer with 1568 input features and 100 output features.

During training, we tested four distinct tiers of data augmentation for both CIFAR-10 (adapted for 32×32 with padding 4) and Imagenette (adapted for 224×224) to evaluate model robustness and capacity under varying degrees of input regularization:

- **None:** Baseline evaluation utilizing base tensor conversion, standard normalization, and (for Imagenette) a resize to 256 followed by a center crop of 224.
- **Light:** Standard protocol utilizing random resized crops and random horizontal flipping ($p = 0.5$).
- **Medium:** Color jittering (brightness, contrast, saturation, and hue all set to 0.1) combined with random affine transformations (maximum rotation of $5°$, maximum translation of 5%), applied after random resized crops and flipping.
- **Heavy:** State-of-the-art augmentation combining RandAugment (2 operations, magnitude 9), random horizontal flipping, and Random Erasing ($p = 0.25$, scale 0.02 to 0.1, ratio 0.3 to 3.3).

C.4 Formal Verification Setup

To rigorously evaluate certified robust accuracy, we extract our formal bounds using the `auto_LiRPA` library. The optimization parameters are distinctly tuned depending on the verification method:

Table 4: Temperature (τ) sweeps bounding the active optimization phase for each architecture and orthogonalization method. Extreme temperatures outside these bounds resulted in severe underfitting or numerical collapse and were filtered for visual clarity.

Architecture	Layer type	τ_{min}	τ_{max}
MLP	GS2	0.5225	100.0
ConvSmall	RKO (Björck)	0.20	100.0
	AOC (GNP)	0.40	100.0
CNN-A	RKO (Björck)	1.00	30.0
	AOC (GNP)	0.50	9.14
ConvLarge	RKO (Björck)	0.60	45.0
	AOC (GNP)	0.50	15.0
VGG-13 (CIFAR-10)	RKO (Björck)	5.00	100.0
	AOC (GNP)	1.00	15.0
VGG-13 (Imagenette)	RKO (Björck)	1.00	70.0
	AOC (GNP)	1.00	12.0
VGG-16 (Imagenette)	RKO (Björck)	5.00	40.0
	AOC (GNP)	0.50	6.00

- **Alpha-CROWN:** We enable the optimization of intermediate bounds for a maximum of 300 iterations. The dual variables (α) are optimized using a learning rate of 0.5, and the process is governed by an early stopping patience of 20 iterations.
- **SDP-CROWN:** Because models trained at varying temperatures (τ) present different optimization landscapes, we execute two distinct hyperparameter sweeps for SDP-CROWN and systematically retain the tightest bound for each sample:
  - *Standard Profile:* 300 optimization iterations, an α learning rate of 0.5, a λ learning rate of 0.05, and an early stopping patience of 20 iterations.
  - *Extended Profile (High τ):* 1000 optimization iterations, aggressive α and λ learning rates of 0.5 with a learning rate decay factor of 0.998, and an extended early stopping patience of 100 iterations.

C.5 Hardware

All experiments, including model training and verification, were conducted using PyTorch on a cluster equipped with A100 and H100 NVIDIA GPUs.

D MNIST Results

Table 5: **Effect of GS2 ReLU Factorization on CROWN and α-CROWN VRA for MNIST.** All models use the robust training setting.

		CROWN VRA (%)		α-CROWN VRA (%)	
Model	**Arch.**	**Factored**	**Native**	**Factored**	**Native**
MLP	GS2	88.0	19.5	88.0	78.0
ConvSmall	GS2 RKO (Björck)	63.5	0.0	65.5	44.0
	GS2 AOC	0.5	0.0	1.5	0.0

Table 6: **Robustness under ℓ_2-norm perturbations on MNIST ($n = 784, \epsilon = 0.3$).** We report the Clean Accuracy, Empirical Robust Accuracy (ERA), the analytical Certified Robust Accuracy (CRA), and the Verified Robust Accuracy (VRA) computed via LiRPA solvers.

			Upper Bound	Analytical	Formal Verification (VRA)	
Model (Setting)	**Architecture**	**Clean (%)**	**ERA (%)**	**CRA (%)**	**α-CROWN**	**SDP-CROWN**
MLP (Accurate)	GS2	98.0	92.0	**87.0**	28.5	71.0
MLP (Robust)	GS2	93.5	91.0	**90.0**	88.0	88.0
ConvSmall (Accurate)	GS2 RKO (Björck)	99.5	95.0	34.5	0.0	**41.5**
	GS2 AOC	99.0	94.5	**47.5**	0.0	42.5
ConvSmall (Robust)	GS2 RKO (Björck)	88.0	85.0	**82.5**	65.5	79.0
	GS2 AOC	92.5	89.5	**88.0**	1.5	85.5
ConvSmall (from Chiu et al. (2025))	Standard CNN	87.5	84.5	77.5	0.0	**81.5**

E Additional Empirical Results

This section provides extended empirical data supporting the primary claims established in Section 6. It includes a statistical evaluation of model training variance, a validation benchmark on an independent test subset, a computational runtime analysis of the verification solvers, and the comprehensive tabular results for the hybrid certification framework.

E.1 Statistical Impact of Model Training

To ensure the reliability of the evaluated optimization regimes, Table 7 reports the mean and standard deviation of all robustness metrics across five independent training runs.

Table 7: **Robustness under ℓ_2-norm perturbations.** We report the mean and standard deviation of Clean Accuracy, Empirical Robust Accuracy, Certified Robust Accuracy, and Verified Robust Accuracy computed via LiRPA solvers across 5 runs.

			Upper Bound	Analytical	Formal Verification (VRA)	
Model (Setting)	**Architecture**	**Clean (%)**	**ERA (%)**	**CRA (%)**	**α-CROWN**	**SDP-CROWN**
CNN-A (Accurate)	RKO (Björck)	75.7 ± 3.1	53.6 ± 4.0	3.9 ± 1.0	0.0	19.5 ± 1.2
	AOC	65.9 ± 1.9	41.0 ± 2.9	33.9 ± 2.6	0.8 ± 1.0	32.8 ± 2.8
CNN-A (Robust)	RKO (Björck)	71.0 ± 4.1	67.8 ± 3.0	64.0 ± 2.2	63.0 ± 2.4	66.2 ± 2.2
	AOC	64.9 ± 2.9	62.1 ± 1.9	61.4 ± 1.9	48.1 ± 2.2	60.9 ± 1.7
ConvLarge (Accurate)	RKO (Björck)	82.0 ± 3.2	78.8 ± 3.5	13.4 ± 2.2	0.0	29.0 ± 3.8
	AOC	80.3 ± 1.8	74.2 ± 1.7	37.2 ± 1.2	0.0	42.4 ± 2.0
ConvLarge (Robust)	RKO (Björck)	71.7 ± 1.7	70.4 ± 1.6	68.0 ± 1.8	61.1 ± 1.8	69.0 ± 1.5
	AOC	67.1 ± 1.7	65.0 ± 1.8	62.4 ± 1.7	43.7 ± 1.4	62.3 ± 1.8
ConvSmall (Accurate)	RKO (Björck)	98.8 ± 0.6	95.5 ± 1.5	32.0 ± 7.1	0.0	48.2 ± 5.4
	AOC	98.6 ± 0.4	92.0 ± 1.6	41.7 ± 3.3	0.0	37.3 ± 3.0
ConvSmall (Robust)	RKO (Björck)	88.4 ± 0.4	83.9 ± 1.3	82.4 ± 0.9	68.9 ± 2.4	80.5 ± 1.1
	AOC	92.2 ± 0.4	88.7 ± 1.6	87.7 ± 1.6	10.7 ± 11.9	85.7 ± 1.0
MLP (Accurate)	AOC	98.1 ± 0.7	89.0 ± 2.8	82.8 ± 2.6	29.3 ± 2.5	70.5 ± 1.9
MLP (Robust)	AOC	93.7 ± 0.6	90.6 ± 0.4	89.4 ± 0.7	87.8 ± 0.9	88.3 ± 0.6

E.2 Validation on an Independent Test Subset

To confirm that the performance dynamics observed in the main text generalize, Table 8 replicates the robustness evaluation on an independent subset of 200 test points.

Table 8: **Robustness Benchmarks on CIFAR-10 and MNIST (200 test points).** We report the maximum values for Clean Accuracy, Empirical Robust Accuracy (ERA), Certified Robust Accuracy (CRA), and Verified Robust Accuracy computed via LiRPA solvers.

			Upper Bound	Analytical	Formal Verification (VRA)	
Model (Setting)	**Architecture**	**Clean (%)**	**ERA (%)**	**CRA (%)**	**α-CROWN**	**SDP-CROWN**
CIFAR-10 Models						
CNN-A (Accurate)	GS2 RKO (Björck)	73.0	51.0	3.5	0.0	19.0
	GS2 AOC	66.0	44.0	34.0	0.0	33.5
CNN-A (Robust)	GS2 RKO (Björck)	70.0	68.5	64.0	62.5	66.5
	GS2 AOC	65.5	63.5	62.0	50.0	62.0
ConvLarge (Accurate)	GS2 RKO (Björck)	79.0	77.5	13.0	0.0	28.0
	GS2 AOC	70.0	65.5	54.0	0.0	56.5
ConvLarge (Robust)	GS2 RKO (Björck)	70.0	67.5	64.5	62.0	65.0
	GS2 AOC	65.0	63.0	62.5	37.0	62.5
MNIST Models						
MLP (Accurate)	GS2 AOC	96.5	83.0	73.5	13.0	60.5
MLP (Robust)	GS2 AOC	85.0	80.0	79.5	77.5	78.0
ConvSmall (Accurate)	GS2 RKO (Björck)	96.5	90.5	10.0	0.0	25.0
	GS2 AOC	95.5	86.0	36.5	0.0	32.5
ConvSmall (Robust)	GS2 RKO (Björck)	79.5	76.0	74.5	57.0	73.0
	GS2 AOC	82.5	79.0	78.0	0.5	77.5

E.3 Verification Timing and Computational Overhead

Table 9 quantifies the computational cost associated with extracting formal bounds via LiRPA solvers, highlighting the runtime impact of the extensive SDP-CROWN optimization profiles relative to α-CROWN.

Table 9: **Verification Runtimes for ConvLarge on CIFAR-10 ($\epsilon = 8/255$) on H100.** We report the computation time (in seconds) for α-CROWN and the two SDP-CROWN setups (Setup 1 / standard and Setup 2 / extended).

		α-CROWN	SDP-CROWN Time (s)	
Model (Setting)	**Architecture**	**Time (s)**	**Setup 1**	**Setup 2**
ConvLarge (Accurate)	GS2 AOC	78.40	28.03	101.45
ConvLarge (Accurate)	GS2 RKO (Björck)	35.44	24.03	94.43
ConvLarge (Robust)	GS2 AOC	12.26	16.10	35.58
ConvLarge (Robust)	GS2 RKO (Björck)	8.85	11.64	16.58

E.4 Extended Hybrid Framework Data

Table 10 details the performance of the hybrid verification applied to both fully constrained models and backbone-constrained architectures (under varying weight decay regularizations). For hybrid models, the CRA is computed by estimating the Lipschitz constant of the last layers with Power Iteration.

Table 10: **Comprehensive Performance Benchmark on CIFAR-10 and Imagenette ($\epsilon = 8/255$).** Comparison between Hybrid Framework models (varying head weight decay `wd`) and fully constrained 1-Lipschitz baselines.

Model	Setting / Architecture	Clean (%)	ERA (%)	CRA (%)	Hybrid VRA (%)
CIFAR-10 ($\epsilon = 8/255$)					
Fully Constrained Baselines					
VGG13 (Accurate)	GS2 RKO (Björck)	84.5	82.0	14.0	4.5
	GS2 AOC	83.5	78.0	60.0	57.5
VGG13 (Robust)	GS2 RKO (Björck)	82.0	79.0	72.5	72.5
	GS2 AOC	82.0	82.0	79.5	79
Hybrid Framework Models (Weight Decay Tuning)					
VGG13	wd = 0.5	85.0	82.0	39.5	55.5
	wd = 0.1	87.5	81.0	18.0	30.5
Imagenette ($\epsilon = 8/255$)					
Fully Constrained Baselines					
VGG13 (Accurate)	GS2 RKO (Björck)	89.5	88.5	44.5	37.0
	GS2 AOC	87.5	87.0	67.0	63.0
VGG13 (Robust)	GS2 RKO (Björck)	83.0	82.0	80.0	80.0
	GS2 AOC	85.0	85.0	84.0	84.0
VGG16 (Accurate)	GS2 RKO (Björck)	85.0	84.5	53.0	46.5
	GS2 AOC	84.0	83.5	71.0	69.0
VGG16 (Robust)	GS2 RKO (Björck)	79.0	79.0	76.5	76.0
	GS2 AOC	81.0	80.5	80.0	80.0
Hybrid Framework Models (Weight Decay Tuning)					
VGG13	wd = 0.5	86.5	86.0	67.0	77.5
	wd = 0.1	89.0	87.0	37.0	66.5
VGG16	wd = 0.5	84.5	84.0	58.0	72.5
	wd = 0.01	86.0	85.5	11.5	59.0

As demonstrated in Table 10, applying the hybrid verification framework to fully constrained 1-Lipschitz architectures yields no certifiable advantage over Lipschitz-certified radii. Localized bound propagation on their final layers fails to uncover additional robustness, resulting in Verified Robust Accuracies (VRA) that merely match or slightly underperform the analytical Certified Robust Accuracies (CRA).

Conversely, the hybrid framework delivers substantial gains when deployed in its optimal structural configuration: a 1-Lipschitz feature-extracting backbone paired with an unconstrained, weight-decay-regularized classification head. In this specific setup, the framework successfully recovers the unproven robustness introduced by relaxing the head's constraints.