Affine-Image Propagation for Tight and Scalable Neural Network Verification

Hong-Ming Chiu*
University of Illinois at Urbana–Champaign
hmchiu2@illinois.edu

Haoyu Li*
University of Illinois at Urbana–Champaign
haoyuli5@illinois.edu

Huan Zhang
University of Illinois at Urbana–Champaign
huan@huan-zhang.com

Richard Y. Zhang
University of Illinois at Urbana–Champaign
ryz@illinois.edu

Abstract

While bound propagation methods are highly effective for certifying neural networks against ℓ_∞ adversaries, tight and scalable verification for ℓ_2 adversaries remains a significant open challenge. Existing methods incur geometric loss by enclosing preactivation sets in elementwise intervals or axis-aligned ellipsoids, and functional loss by restricting the linear bounds constructed over those domains. Consequently, tight ℓ_2 verification is currently restricted to Lipschitz-regularized networks where these approximations are accurate. In this paper, we introduce affine-image SDP-CROWN, a novel bound propagation method that better approximates preactivation domains using affine-image constraints and jointly optimizes the linear bound to adapt to the resulting domain. To bypass the prohibitive computational cost of dense semidefinite constraints required by the affine-image representation, we replace them with scaled diagonal-dominance conditions, allowing a GPU-accelerated implementation. Our method preserves the bound propagation recursion and can be integrated with branch-and-bound. On five MNIST and CIFAR-10 non-Lipschitz-regularized networks, our method improves verified accuracy over state-of-the-art methods by 17% to 57.5%. With branch-and-bound, these improvements increase to between 27% and 63.5%. The code for reproducing our experimental results is available at https://github.com/Verified-Intelligence/Affine-Image-SDP-CROWN.

1 Introduction

Bound propagation (Zhang et al., 2018; Singh et al., 2018, 2019b; Wang et al., 2018; Dvijotham et al., 2018; Hashemi et al., 2021; Xu et al., 2020, 2021) has emerged as a highly effective technique for the formal verification of neural networks against ℓ_∞ adversaries, yet it surprisingly struggles when faced with ℓ_2 adversaries. Addressing this disparity is increasingly critical. Modern robustness questions are moving beyond pixel-wise perturbations in image space toward semantic perturbations in latent representations (Wong and Kolter, 2021; Barrett et al., 2022), where Euclidean geometry is the natural model. Scalable ℓ_2 verification is therefore a necessary step toward certifying modern representation learners.

Existing bound propagation methods can incur two sources of looseness: geometric loss from approximating preactivation sets by larger, simpler domains, and functional loss from restricting the linear bounds constructed over those domains. For ℓ_2 perturbations, the reachable preactivation set is obtained by mapping the input Euclidean ball through the network. Methods based on elementwise

*Equal contribution

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

intervals, such as α-CROWN (Xu et al., 2021), construct their linear activation bounds over coordinatewise preactivation intervals. These bounds must therefore remain valid for combinations of preactivations outside the reachable set. For example, the elementwise box of an n-dimensional ball has a Euclidean radius $\sqrt{n}$ times that of the ball. Constructing activation bounds over such enlarged domains can introduce geometric loss and weaken the resulting certificates.

A significant breakthrough in reducing this geometric loss came from SDP-CROWN (Chiu et al., 2025), which integrated semidefinite programming (SDP) relaxations (Raghunathan et al., 2018; Brown et al., 2022) into scalable bound propagation. Although SDP relaxations have long provided tight certificates for ℓ_2 verification (Zhang, 2020; Chiu and Zhang, 2023), their computational cost has limited their scalability. SDP-CROWN addressed this bottleneck by deriving layerwise linear bounds from SDP relaxations and incorporating them into the backward pass, preserving the computational structure of bound propagation. The key insight is that, for preactivation domains modeled as Euclidean balls or axis-aligned ellipsoids, the SDP relaxation yields an explicit family of linear bounds. These bounds incorporate joint ℓ_2 constraints directly into bound propagation, avoiding the $\sqrt{n}$ increase in Euclidean radius that arises when a ball is enclosed in an elementwise box.

However, SDP-CROWN (Chiu et al., 2025) relies on the propagated axis-aligned ellipsoids remaining informative throughout the network. Multiplication by a weight matrix generally changes both the orientation of an ellipsoid. To maintain efficiency, SDP-CROWN encloses the resulting image using another axis-aligned ellipsoid. This approach works well on the Lipschitz-regularized benchmarks, where even naive spectral-norm propagation provides informative bounds. For non-Lipschitz-regularized networks, such enclosures can substantially enlarge the set along directions in which the image is narrow, introducing geometric loss. The resulting SDP constraints may then be too loose to yield tight certificates, limiting the effectiveness of SDP-CROWN on these networks.

This leaves a fundamental research question: *How can we achieve tight verification against ℓ_2 adversaries on non-Lipschitz-regularized networks while retaining the scalability of bound propagation?* To address this question, we develop affine-image SDP-CROWN, which augments elementwise interval and axis-aligned ellipsoid constraints with the affine image of a previous-layer outer approximation. For an ellipsoidal ℓ_2 input set, this representation captures the first preactivation domain exactly. In subsequent layers, it preserves the affine geometry of the previous-layer outer approximation, with approximation error inherited from earlier layers. We jointly optimize the slopes and offsets of the propagated linear bounds to exploit these richer domains.

The main technical challenge is to construct linear bounds over affine-image domains efficiently. The associated SDP relaxation introduces a dense positive semidefinite constraint, whose direct enforcement can be computationally expensive. We derive tractable sufficient conditions based on scaled diagonal-dominance and optimize the scaling jointly with the bound parameters. For convolutional layers, our GPU implementation exploits sparsity to evaluate the required terms and gradients without forming dense matrices. The resulting certificates incorporate affine-image constraints while preserving the linear bound propagation recursion, making stronger ℓ_2 verification practical for non-Lipschitz-regularized networks.

1.1 Contributions

We introduce affine-image SDP-CROWN for scalable ℓ_2 verification of non-Lipschitz-regularized ReLU networks. Our method augments interval and ellipsoidal constraints with affine images of previous-layer outer approximations and jointly optimizes the propagated linear bounds, addressing both geometric and functional loss. Scaled diagonal-dominance conditions and a GPU implementation that exploits convolutional sparsity make the resulting certificates tractable. The method preserves the bound propagation recursion, integrates with branch-and-bound, and substantially improves verified accuracy on the evaluated MNIST and CIFAR-10 networks.

1.2 Related work

Semidefinite programming (SDP) relaxation is a principled approach for verifying neural networks against ℓ_2 adversaries, due to its ability to capture the geometry induced by ℓ_2-norm. Consequently, SDP-based methods (Raghunathan et al., 2018; Fazlyab et al., 2019; Batten et al., 2021; Ma and Sojoudi, 2020; Fazlyab et al., 2020; Dvijotham et al., 2020) provide some of the tightest known verification against ℓ_2 adversaries. However, SDP relaxations suffer from poor scalability; even with

recent advances in solver design (Chiu and Zhang, 2023) and hardware acceleration (Dathathri et al., 2020b), they remain computationally prohibitive for networks with more than 10 thousand neurons. Moreover, their high computational cost prevents the propagation of intermediate preactivation bounds using the same SDP relaxation. As observed in (Raghunathan et al., 2018; Dathathri et al., 2020b; Chiu et al., 2025), tight intermediate bounds are critical for maintaining the effectiveness of SDP relaxations in deep networks.

Alternatively, bound propagation methods based on linear programming (LP) relaxations (Zhang et al., 2018; Singh et al., 2018, 2019b; Wang et al., 2018; Dvijotham et al., 2018; Hashemi et al., 2021; Xu et al., 2020, 2021) enable efficient and highly parallelizable computation of intermediate bounds. While these methods are primarily designed for ℓ_∞ adversaries and are generally looser than SDP-based approaches, they scale to networks with millions of neurons. To improve verification tightness, prior works combine bound propagation with branch-and-bound (Wang et al., 2021; De Palma et al., 2021; Ferrari et al., 2022; Shi et al., 2025), cutting planes (Zhang et al., 2022; Zhou et al., 2024), and domain clipping techniques (Zhou et al., 2025). Although these approaches can significantly tighten verification for small perturbation radii, their effectiveness typically deteriorates as the perturbation radius increases due to the rapidly growing number of unstable neurons and increasingly loose LP relaxations. To bridge this gap, several works tighten LP-based verification by introducing multi-neuron or group-wise constraints that capture dependencies among multiple activations (Singh et al., 2019a; Müller et al., 2022; Ferrari et al., 2022; Ma et al., 2024).

Recently, (Chiu et al., 2025) integrated SDP relaxation into bound propagation, leading to significantly tighter ℓ_2 verification with the scalability of bound propagation methods. In particular, (Chiu et al., 2025) improves the relaxation of intermediate preactivations by propagating ℓ_2-norm ball constraints through the network. However, its effectiveness can degrade significantly when the network is not Lipschitz-regularized, as illustrated in Table 1.

1.3 Notations

We use $\mathbb{S}^n$ to denote a set of symmetric $n \times n$ matrices. We use $\mathbb{R}^n_+$ and $\mathbb{R}^n_{++}$ to denote a set of nonnegative and positive length n vectors. We use x_i to denote the i-th entry of x. We use $\mathbf{1}$ to denote a column of ones. We use $\odot$ to denote elementwise multiplication. We use e_i to denote i-th standard basis vector. We use $X \succeq 0$ to denote X being positive semidefinite. We use $\operatorname{diag}(x) \in \mathbb{S}^n$ to denote a diagonal matrix whose diagonal is $x \in \mathbb{R}^n$ when applied to a vector, and use $\operatorname{diag}(X) \in \mathbb{R}^n$ to denote the diagonal entries of $X \in \mathbb{S}^n$ when applied to a matrix. We use $|\cdot|$ to denote the elementwise absolute value, and $\|\cdot\|_p$ to denote ℓ_p norm. We use $[\underline{x}, \overline{x}] = \{x \mid \underline{x}_i \le x_i \le \overline{x}_i \text{ for all } i\}$ to denote an elementwise interval between $\underline{x}$ and $\overline{x}$.

2 Background

2.1 Verification by bound propagation

Given an input uncertainty set $\mathcal{X}_0$, the verification problem is to compute a provable lower bound on

$$f_\star = \min_{x^{(0)} \in \mathcal{X}_0} \left\langle c^{(N)}, x^{(N)} \right\rangle \text{ s.t. } x^{(k)} = \operatorname{ReLU}(z^{(k)}),\ \ z^{(k)} = W^{(k)} x^{(k-1)} + b^{(k)} \text{ for } k = 1, \dots, N$$

where $W^{(k)}$ and $b^{(k)}$ are the weights and biases, and $c^{(N)}$ is an output specification vector. We focus on the input set being an ℓ_p-norm axis-aligned ellipsoid with center $\hat{x}$ and semiaxes $d > 0$:

$$\mathcal{X}_0 = \mathcal{E}_p(\hat{x}, d) = \{\hat{x} + \operatorname{diag}(d) u \mid \|u\|_p \le 1\}.$$

We focus on $p = 2$ in this work. Our method can be extended to any $1 \le p \le \infty$, and we describe these extensions in Appendix B. Bound propagation is effectively a dynamic programming algorithm for solving this problem. At layer k, it constructs a lower bound

$$\varphi_k(c) \le f_k(c) \coloneqq \min_{x \in \mathcal{X}_k} c^T x, \qquad \text{where } \mathcal{X}_k \coloneqq \{\operatorname{ReLU}(W^{(k)} x + b^{(k)}) \mid x \in \mathcal{X}_{k-1}\},$$

assuming that a lower bound $\varphi_{k-1}(c) \le f_{k-1}(c)$ has already been constructed for the previous layer. The base case is available in closed form

$$\varphi_0(c) \coloneqq f_0(c) = \min_{x \in \mathcal{X}_0} c^T x = c^T \hat{x} + \min_{\|u\|_p \le 1} c^T \operatorname{diag}(d) u = c^T \hat{x} - \|c \odot d\|_q$$

where $1/p + 1/q = 1$. By rolling out the induction, we obtain $\varphi_N(c^{(N)}) \le f_\star$ at the end.

2.2 α-CROWN

Given a valid lower bound $\varphi_{k-1}(\cdot) \le f_{k-1}(\cdot)$ for the previous layer, α-CROWN (Xu et al., 2021) constructs a valid lower bound $\varphi_k(c) \le f_k(c)$ in three steps. The first step is forward propagation. Using φ_{k-1}, we obtain interval bounds on the preactivation variable $z = W^{(k)}x + b^{(k)}$, namely

$$\underline{z}_i := \varphi_{k-1}(W^{(k)T} e_i) + b_i^{(k)}, \qquad \overline{z}_i := -\varphi_{k-1}(-W^{(k)T} e_i) + b_i^{(k)},$$

so that $z \in [\underline{z}, \overline{z}]$. The k-th layer subproblem is augmented with redundant interval bounds as

$$f_k(c) = \min_{x^+ \in \mathcal{X}_k} c^T x^+ = \min_{z \in [\underline{z}, \overline{z}]} \left\{ c^T \operatorname{ReLU}(z) : z = W^{(k)}x + b^{(k)},\ x \in \mathcal{X}_{k-1} \right\}.$$

The second step is to replace the ReLU activation by a linear lower bound. The key innovation of α-CROWN is to optimize over a family of lower bounds, parameterized by $\alpha \in [0,1]$, such that

$$g(\alpha)^T z + h(\alpha) \le c^T \operatorname{ReLU}(z) \quad \text{for all } z \in [\underline{z}, \overline{z}]. \tag{1}$$

The third step is back propagation. Substituting the linear lower bound gives

$$\begin{aligned} f_k(c) &\ge \min_{z \in [\underline{z}, \overline{z}]} \left\{ g(\alpha)^T z + h(\alpha) : z = W^{(k)}x + b^{(k)},\ x \in \mathcal{X}_{k-1} \right\} \\ &= \min_{x \in \mathcal{X}_{k-1}} \left\langle g(\alpha), W^{(k)}x + b^{(k)} \right\rangle + h(\alpha) = f_{k-1}(W^{(k)T} g(\alpha)) + b^{(k)T} g(\alpha) + h(\alpha) \\ &\ge \varphi_{k-1}(W^{(k)T} g(\alpha)) + b^{(k)T} g(\alpha) + h(\alpha). \end{aligned}$$

Optimizing over all such lower bounds gives

$$\varphi_k(c) := \max_{\alpha \in [0,1]} \ \varphi_{k-1}(W^{(k)T} g(\alpha)) + b^{(k)T} g(\alpha) + h(\alpha).$$

The validity $\varphi_k(c) \le f_k(c)$ follows from (1) and the induction hypothesis $\varphi_{k-1}(c) \le f_{k-1}(c)$.

2.3 Prior SDP-CROWN

The prior SDP-CROWN (Chiu et al., 2025) extends α-CROWN to work effectively for an axis-aligned ℓ_2-norm ellipsoid input set. At each layer k, it assumes an outer ellipsoid approximation $\mathcal{E}_2(\hat{x}, d_x) \supseteq \mathcal{X}_{k-1}$ for the previous layer. In the forward propagation step, it propagates the ellipsoid through the affine map, by defining a center $\hat{z} = W^{(k)}\hat{x} + b^{(k)}$ and semiaxes $d_z = \kappa a_z$ where

$$\kappa = \| \operatorname{diag}(a_z)^{-1} W^{(k)} \operatorname{diag}(d_x) \|_2, \quad a_z = \sqrt{(W^{(k)} \odot W^{(k)})(d_x \odot d_x)}, \tag{2}$$

and augmenting the k-th layer subproblem with both redundant interval and ellipsoid constraints. It then constructs a family of lower bounds, using the same parameters $\alpha \in [0,1]$ as α-CROWN, together with a new parameter $\lambda \ge 0$ for the ellipsoid, such that

$$g(\alpha)^T z + h(\alpha, \lambda) \le c^T \operatorname{ReLU}(z) \quad \text{for all } z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d_z). \tag{3}$$

The key idea is to shrink the uncertainty set on which the lower bound must hold, thereby making the bound tighter. Finally, it back propagates the bound in the same way:

$$\varphi_k(c) := \max_{\alpha \in [0,1], \lambda \ge 0} \ \varphi_{k-1}(W^{(k)T} g(\alpha)) + b^{(k)T} g(\alpha) + h(\alpha, \lambda).$$

Then $\varphi_k(c) \le f_k(c)$ follows from (3) and the induction hypothesis $\varphi_{k-1}(c) \le f_{k-1}(c)$.

3 Looseness in existing methods

The effectiveness of a verification method depends on the tightness of its certified lower bound $\varphi_N(c^{(N)}) \le f_\star$. In bound propagation, looseness arises from two distinct sources.

Geometric loss. At the k-th layer, the preactivation set $\mathcal{Z}_k$ is relaxed to a tractable outer approximation

$$\tilde{\mathcal{Z}}_k \supseteq \mathcal{Z}_k := \{ W^{(k)}x + b^{(k)} \mid x \in \mathcal{X}_{k-1} \}.$$

The relevant question is: How much larger is $\tilde{\mathcal{Z}}_k$ than the true set $\mathcal{Z}_k$?

Functional loss. The verifier then constructs affine lower bounds on the ReLU objective over this tractable set:

$$g^T z + h \le c^T \operatorname{ReLU}(z) \quad \text{for all } z \in \tilde{\mathcal{Z}}_k.$$

The relevant question is: How tightly can the nonlinear ReLU objective be lower bounded?

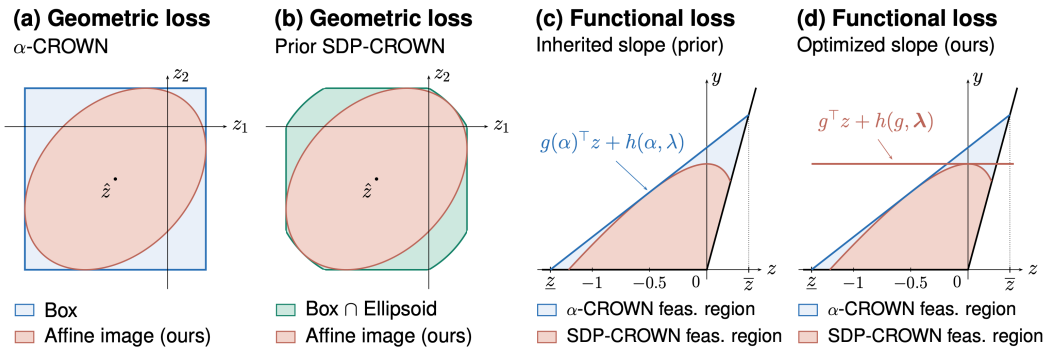


Figure 1: **Geometric and functional loss in bound propagation.** (a) α-CROWN incurs geometric loss by enclosing the affine image in a box. (b) Prior SDP-CROWN reduces this loss by intersecting the box with an axis-aligned ellipsoid. Our method retains the affine image, avoiding this source of geometric loss. (c) In this example, the linear upper bound with inherited slope $g(\alpha) = \overline{z}/(\overline{z} - \underline{z})$ cannot be tightened even though SDP-CROWN has a smaller feasible region. (d) We allow the slope g to be optimized, allowing it to exploit the smaller feasible region.

3.1 Looseness in α-CROWN

The bound propagation method α-CROWN is designed to eliminate the functional loss. It uses an interval as the tractable outer approximation

$$\tilde{\mathcal{Z}}_k = [\underline{z}, \overline{z}] \supseteq \mathcal{Z}_k,$$

so that the ReLU objective separates coordinatewise. This allows α-CROWN to construct an *exactly tight* family of affine lower bounds, $g(\alpha)^T z + h(\alpha)$, in the sense that

$$\max_{\alpha \in [0,1]} \min_{z \in [\underline{z}, \overline{z}]} g(\alpha)^T z + h(\alpha) = \min_{z \in [\underline{z}, \overline{z}]} c^T \operatorname{ReLU}(z).$$

Specifically, for every c, there exists an $\alpha \in [0, 1]$ that attains equality $c^T \operatorname{ReLU}(z) = g(\alpha)^T z + h(\alpha)$ for some $z \in [\underline{z}, \overline{z}]$ that minimizes $c^T \operatorname{ReLU}(z)$. Hence, α-CROWN incurs no functional loss.

However, the tradeoff is potentially significant geometric loss. The interval approximation $[\underline{z}, \overline{z}] \supseteq \mathcal{Z}_k$ can dramatically increase the size of the preactivation set, particularly for ℓ_2 adversaries. For example, if $\mathcal{Z}_k$ is an axis-aligned ellipsoid $\mathcal{E}_2(\hat{z}, d_z)$, then its elementwise interval is $\mathcal{E}_\infty(\hat{z}, d_z) = [\hat{z} - d_z, \hat{z} + d_z]$, whose Euclidean radius is larger by a factor of $\|d_z\|_2 / \|d_z\|_\infty$

$$\sup_{z \in \mathcal{E}_\infty(\hat{z}, d_z)} \|z - \hat{z}\|_2 = \frac{\|d_z\|_2}{\|d_z\|_\infty} \sup_{z \in \mathcal{E}_2(\hat{z}, d_z)} \|z - \hat{z}\|_2.$$

Thus, replacing the ellipsoid by its coordinate box increases the maximum Euclidean deviation by a factor of $\|d_z\|_2 / \|d_z\|_\infty$, reducing the tightness of α-CROWN.

3.2 Looseness in prior SDP-CROWN

Prior SDP-CROWN reduces the geometric loss in α-CROWN by choosing a more expressive tractable outer approximation, which is the intersection of an interval and an axis-aligned ellipsoid,

$$\tilde{\mathcal{Z}}_k = [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d_z) \supseteq \mathcal{Z}_k.$$

This directly targets the geometric loss caused by replacing an ℓ_2-shaped uncertainty set by an interval. When $\mathcal{Z}_k$ is itself an axis-aligned ellipsoid $\mathcal{E}_2(\hat{z}, d_z)$, prior SDP-CROWN can represent the preactivation set exactly, hence yielding a factor of $\|d_z\|_2 / \|d_z\|_\infty$ improvement over the interval relaxation $\mathcal{E}_\infty(\hat{z}, d_z)$ used by α-CROWN.

Unfortunately, this new tractable outer approximation is still not expressive enough to be broadly useful for ℓ_2 adversaries. In general, $\mathcal{Z}_k$ is an affine image under $\mathcal{X}_{k-1}$, for which an axis-aligned ellipsoid cannot closely capture. For instance, even when $\mathcal{X}_{k-1}$ is an axis-aligned ellipsoid $\mathcal{E}_2(\hat{x}, d_x)$,

its image under the affine transformation $z = W^{(k)}x + b^{(k)}$ is generally an ellipsoid that is not axis-aligned. The prior SDP-CROWN therefore encloses this ellipsoid in an axis-aligned ellipsoid $\mathcal{E}_2(\hat{z}, d_z)$, with center $\hat{z} = W^{(k)}\hat{x} + b^{(k)}$ and semiaxes $d_z = \kappa a_z$ controlled by the spectral norm $\kappa = \| \operatorname{diag}(a_z)^{-1} W^{(k)} \operatorname{diag}(d_x) \|_2$ and a_z defined in (2). This approximation can cause substantial enlargement along directions in which the original ellipsoid is narrow.

To better illustrate this point, write $W = \operatorname{diag}(d_z)^{-1} W^{(k)} \operatorname{diag}(d_x)$, and let $v_{\min}$ be a left singular vector associated with $\sigma_{\min}(W)$. Then the axis-aligned ellipsoid is larger than the true ellipsoid by a factor of $\operatorname{cond}(W) := \sigma_{\max}(W)/\sigma_{\min}(W)$ in the direction $v_{\min}$:

$$\sup_{z \in \mathcal{E}_2(\hat{z}, d_z)} \left| \left\langle v_{\min}, \operatorname{diag}(d_z)^{-1}(z - \hat{z}) \right\rangle \right| = \sup_{\|u\|_2 \le 1} |\langle v_{\min}, u \rangle| = \operatorname{cond}(W) \sup_{\|u\|_2 \le 1} |\langle v_{\min}, Wu \rangle|$$

$$= \operatorname{cond}(W) \sup_{z} \left\{ \left| \left\langle v_{\min}, \operatorname{diag}(d_z)^{-1}(z - \hat{z}) \right\rangle \right| : z = W^{(k)}x + b^{(k)},\ x \in \mathcal{E}_2(\hat{x}, d_x) \right\}.$$

The enclosing ellipsoid is therefore $\operatorname{cond}(W)$ times wider in this normalized direction. Hence, poor conditioning $\operatorname{cond}(W) \to \infty$ can cause substantial enlargement along directions in which the affine image is narrow, reducing the tightness of prior SDP-CROWN. We extend the prior SDP-CROWN to construct the outer approximation using an affine image, thereby reducing the geometric loss.

In addition to geometric loss, the prior SDP-CROWN also incurs functional loss in the sense that

$$\max_{\alpha \in [0,1], \lambda \ge 0} \min_{z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d_z)} g(\alpha)^T z + h(\alpha, \lambda) \le \min_{z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d_z)} c^T \operatorname{ReLU}(z).$$

For a zero-centered ellipsoid, i.e., $\hat{z} = 0$ and $-\underline{z} = \overline{z} = d_z$, Chiu et al. (2025) show that the bound used by the prior SDP-CROWN is exactly tight. Specifically, for every c, there exists an $\alpha \in [0, 1]$ and $\lambda \ge 0$ that attains equality $c^T \operatorname{ReLU}(z) = g(\alpha)^T z + h(\alpha, \lambda)$ for some $z \in [-d_z, d_z] \cap \mathcal{E}_2(0, d_z)$ that minimizes $c^T \operatorname{ReLU}(z)$. However, when the ellipsoid is not zero-centered, the prior SDP-CROWN can suffer from functional loss. The central weakness in the prior SDP-CROWN is to inherit the slope $g = g(\alpha)$ from α-CROWN, which limits how well the linear bound exploits the ellipsoidal domain even after optimizing α and λ, introducing functional loss as shown in Figure 1. We jointly optimize g and $h(g, \boldsymbol{\lambda})$ over domains strengthened by affine-image constraints, which substantially improve verification performance in our experiments.

4 This work: SDP-CROWN with affine-image propagation

In this paper, we improve the prior SDP-CROWN by augmenting ellipsoid constraints with *affine-image constraints*. Instead of approximating the preactivation set only by an intersection of an interval and an axis-aligned ellipsoid $[\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d_z)$, as in prior SDP-CROWN, we additionally intersect with an affine image of an ℓ_p-norm ellipsoid:

$$\tilde{\mathcal{Z}}_k = [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d_z) \cap \{W^{(k)}x + b^{(k)} \mid x \in \mathcal{E}_p(\hat{x}, d_x)\} \quad \text{with} \quad \hat{z} = W^{(k)}\hat{x} + b^{(k)}. \tag{4}$$

This seemingly small change in representation directly tackles the geometric loss identified in the previous section. The affine-image (4) also allows our method to naturally handle verification problems with general ℓ_p-norm input sets $\mathcal{X}_0$. In contrast, the ellipsoid propagation in prior SDP-CROWN is limited to ℓ_2-norm input sets. We focus on the ℓ_2 case, i.e., $p = 2$ in (4), in the main paper and defer the ℓ_p-norm generalization to Appendix B.

At the input layer, the improvement is immediate. If the input uncertainty set $\mathcal{X}_0 = \mathcal{E}_2(\hat{x}, d)$ is an ellipsoid, then the first preactivation set is exactly the affine image of that ellipsoid. Hence the representation (4) gives $\tilde{\mathcal{Z}}_1 = \mathcal{Z}_1$ and incurs no geometric loss at the input. This significantly improves upon α-CROWN and prior SDP-CROWN respectively, as previously described.

But the same representation is also much more useful in the intermediate layers. To make the accounting precise, let $\tilde{\mathcal{X}}_{k-1} \supseteq \mathcal{X}_{k-1}$ denote the outer approximation carried from the previous layer. Our relaxation uses

$$\tilde{\mathcal{Z}}_k = [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d_z) \cap \{W^{(k)}x + b^{(k)} \mid x \in \tilde{\mathcal{X}}_{k-1}\} \supseteq \{W^{(k)}x + b^{(k)} \mid x \in \mathcal{X}_{k-1}\} = \mathcal{Z}_k$$

to approximate the exact preactivation set $\mathcal{Z}_k$, where the affine map $W^{(k)}x + b^{(k)}$ itself is applied exactly; if $\tilde{\mathcal{X}}_{k-1} = \mathcal{X}_{k-1}$, then $\tilde{\mathcal{Z}}_k = \mathcal{Z}_k$, up to the redundant valid interval constraints. More

generally, the only geometric loss is the loss already present in the previous-layer relaxation $\tilde{\mathcal{X}}_{k-1} \supseteq \mathcal{X}_{k-1}$. This is the key distinction from prior methods, for which only our representation maintains the affine geometry and cross-neuron correlations encoded by $W^{(k)}$.

The remaining challenge is to keep functional loss small. Following the same semidefinite programming steps as (Chiu et al., 2025) to derive a lower bound on the affine image $\{Wx + b \mid x \in \mathcal{E}_2(\hat{x}, d_x)\}$ leads to semidefinite constraints of the form

$$\sigma I \succeq \operatorname{diag}(d_x) W^T \operatorname{diag}(\lambda) W \operatorname{diag}(d_x), \tag{5}$$

where $\lambda \geq 0$ and $\sigma \geq 0$ are Lagrange multipliers that must be optimized as part of the bound. Enforcing this constraint exactly would indeed yield very small functional loss, but would also require solving a large and fully dense SDP at every step. That is far too expensive for a scalable verifier.

The prior SDP-CROWN was able to work on a large scale with bound propagation because it replaced (5) with the spectral sufficient condition

$$\sigma \geq \|W \operatorname{diag}(d_x)\|_2^2 \|\lambda\|_\infty. \tag{6}$$

This crude sufficient condition (6) is sufficiently tight when W has small Lipschitz constant, which is why SDP-CROWN was able to achieve very small functional loss in Lipschitz-regularized networks. For more general ellipsoids, our key observation is that one can keep the affine image while replacing the full SDP constraint by a scaled diagonal-dominance surrogate. For example, a scaled diagonal-dominance condition takes the form

$$\sigma \geq \|\gamma(\operatorname{diag}(d_x) W^T \operatorname{diag}(\lambda) W \operatorname{diag}(d_x), l)\|_\infty \quad \text{for some } l > 0$$

where $\gamma : \mathbb{S}^n \times \mathbb{R}^n_{++} \to \mathbb{R}^n_+$ is the scaled Gershgorin operation

$$\gamma_i(M, l) = \max \left\{ 0, M_{ii} + \sum_{j \neq i} \frac{l_j}{l_i} |M_{ij}| \right\} \quad \text{for } i = 1, \ldots, n \tag{7}$$

We use scaled diagonal-dominance to obtain tractable sufficient conditions while retaining the affine-image constraints. It preserves the directions and correlations encoded by W, while remaining compatible with the back-propagation structure of the bound propagation framework. This approach can be particularly effective for convolutional layers, for which $W^T W$ could naturally exhibit diagonal-dominant structure, as illustrate in Figure 4. The resulting lower bound is stated below.

Theorem 1. *Given $c \in \mathbb{R}^m$, an affine-image $\{Wx + b \mid x \in \mathcal{E}_2(\hat{x}, d_x)\}$ with $W \in \mathbb{R}^{m \times n}$, an ellipsoid $\mathcal{E}_2(\hat{z}, d_z)$ with $\hat{z} = W\hat{x} + b$, and an interval $[\underline{z}, \overline{z}]$ with $\underline{z}, \overline{z} \in \mathbb{R}^m$. The following holds*

$$g^T z + h(g, \boldsymbol{\lambda}) \leq c^T \operatorname{ReLU}(z) \quad \textit{for all } z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d_z) \cap \{Wx + b \mid x \in \mathcal{E}_2(\hat{x}, d_x)\}$$

for any $g \in \mathbb{R}^m$, $\boldsymbol{\lambda} = (\lambda, \tau, l, \eta) \in \mathbb{R}^m_+ \times \mathbb{R}^m_+ \times \mathbb{R}^n_{++} \times \mathbb{R}_+$, and

$$h(g, \boldsymbol{\lambda}) = -\tau^T \overline{q} - \frac{1}{2} \left[\eta + \|\gamma(M(\lambda, \eta), l)\|_\infty + \sum_{j=1}^m \left(\frac{\phi_j^2}{\lambda_j} - \lambda_j \hat{z}_j^2 \right) \right]$$

where γ is the scaled Gershgorin operation defined in (7), and

$$M(\lambda, \eta) = \operatorname{diag}(d_x) W^T \left[\operatorname{diag}(\lambda) - \eta \operatorname{diag}(d_z)^{-2} \right] W \operatorname{diag}(d_x),$$
$$\phi_j = \min \left\{ c_j - g_j - \tau_j(\overline{p}_j - 1) - \lambda_j \hat{z}_j,\ g_j + \tau_j \overline{p}_j + \lambda_j \hat{z}_j,\ 0 \right\}$$

for $j = 1, \ldots, m$. Here, $\overline{p}_j, \overline{q}_j$ are the parameters of the linear upper bound $\operatorname{ReLU}(z_j) \leq \overline{p}_j z_j + \overline{q}_j$ for all $z_j \in [\underline{z}_j, \overline{z}_j]$, and are defined in (9).

4.1 Integration with bound propagation

Theorem 1 can be inserted into bound propagation. At layer k, we first propagate the interval bounds $[\underline{z}, \overline{z}]$ and the affine-image representation to construct $\tilde{\mathcal{Z}}_k$. Then, for a backward specification vector c, Theorem 1 gives a valid affine lower bound on $c^T \operatorname{ReLU}(z)$ over $\tilde{\mathcal{Z}}_k$. Substituting $z = W^{(k)} x + b^{(k)}$ gives the recursive bound

$$f_k(c) \geq f_{k-1}(W^{(k)T} g) + g^T b^{(k)} + h(g, \boldsymbol{\lambda}).$$

Thus, the propagated lower bound is

$$\varphi_k(c) = \max_{g,\boldsymbol{\lambda}} \left\{ \varphi_{k-1}(W^{(k)T} g) + g^T b^{(k)} + h(g, \boldsymbol{\lambda}) \right\},$$

where the maximization is over the admissible parameters g and $\boldsymbol{\lambda}$ in Theorem 1. We note that we allow g to be optimized freely in this work, as opposed to prior SDP-CROWN, which inherits $g = g(\alpha)$ from α-CROWN. The backward pass sends $W^{(k)T} g$ to the previous layer and accumulates the scalar offset $b^{(k)T} g + h(g, \boldsymbol{\lambda})$. Since the parameters $g, \boldsymbol{\lambda}$ are either unconstrained or nonnegative/positive, they can be optimized by projected gradient ascent similar to other bound propagation methods.

The only major additional computation is the diagonal-dominance surrogate $\|\gamma(M(\lambda, \eta), l)\|_\infty$. To illustrate the computation, consider $d_x = \mathbf{1}$, $d_z = \mathbf{1}$, $l = \mathbf{1}$ and $\eta = 0$ as they do not change the structure in W. When W corresponds to a convolutional layer, the sparse convolutional structure is preserved during the computation of $\|\gamma(M(\lambda, 0), \mathbf{1})\|_\infty = \|W^T \operatorname{diag}(\lambda) W\|_\infty$. In particular, two input neurons can have a nonzero interaction in $W^T \operatorname{diag}(\lambda) W$ only if they appear together in at least one convolution window. If W has kernel size $k_h \times k_w$, then the resulting interaction pattern has effective spatial support at most $(2k_h - 1) \times (2k_w - 1)$. Our GPU-optimized implementation exploits this sparsity directly to construct the matrix $W^T \operatorname{diag}(\lambda) W$, by enumerating the local interactions generated from each convolution window in parallel and accumulating only the nonzero terms.

As a result, our method preserves the computational structure of bound propagation while retaining more of the geometry of preactivation sets. Compared with prior SDP-CROWN, it retains affine-image constraints alongside axis-aligned enclosures, reducing geometric loss. Compared with full SDP verifiers, it avoids dense semidefinite optimization. This offers a practical balance between tightness and computational cost while retaining the recursive, GPU-accelerated backward pass of existing bound propagation methods.

4.2 Integration with branch-and-bound

Our method can also be integrated into the branch-and-bound framework of β-CROWN (Wang et al., 2021). We retain the α, β-CROWN branch-and-bound framework and replace only the bound computation for each subproblem with our approach. Each ReLU split introduces an additional dual variable β, which is then optimized jointly with the other multipliers of our bound. A subproblem inherits and is warm-started with the optimized multipliers of its parent, so a few optimization steps suffice per subproblem. As a design choice, we fix the affine-image multipliers λ, including the scaling vector l, at their optimized root values throughout branch-and-bound. This choice allows the affine-image term, the most expensive component of the bound, to be computed once rather than at every optimization step.

5 Experiments

Benchmarks. We evaluate our method on both Lipschitz-regularized networks and non-Lipschitz-regularized networks. The Lipschitz-regularized networks are adopted directly from the previous SDP-CROWN paper (Chiu et al., 2025), using the same perturbation radii and experimental settings. To further test scalability and robustness on more challenging models, we also evaluate on non-Lipschitz-regularized networks from prior works (Salman et al., 2019; Dathathri et al., 2020a). Details of the model architectures and estimated Lipschitz constants are provided in Appendix D.

Baselines & Implementations. We compare our method with several state-of-the-art verifiers for robustness under ℓ_2-norm perturbations. In particular, we consider two incomplete verifiers, namely α-CROWN (Xu et al., 2021) and the original SDP-CROWN (Chiu et al., 2025), which restricts the linear coefficient g to the α-CROWN parametrization. We also consider two stronger verifiers that incorporate branch-and-bound and cutting planes, β-CROWN (Wang et al., 2021) and BICCOS (Zhou et al., 2024). We intentionally omit GCP-CROWN in our comparison as its cut generation process cannot generate proper cuts for ℓ_2 perturbations. We also include LipNaive, a simple Lipschitz baseline used in previous work, which certifies robustness with an upper bound on the network's Lipschitz constant obtained by multiplying the Lipschitz constants of the individual weight matrices. On the Lipschitz-regularized networks, both the original SDP-CROWN and our affine-image SDP-CROWN use the ball propagated with the Lipschitz constants, as in the original SDP-CROWN paper,

Network	Upper Bound	Affine-Image SDP-CROWN + BaB	Affine-Image SDP-CROWN	Original SDP-CROWN	BICCOS	β-CROWN	α-CROWN	LipNaive
Lipschitz-regularized / Small Lipschitz-constant Networks								
MNIST Models								
MLP ($\rho = 1.0$)	50.5%	**49.5%** (8.36s)	44% (2.10s)	32.5% (1.93s)	40.5% (57.51s)	39% (48.87s)	1.5% (1.38s)	29% (0.02s)
ConvSmall ($\rho = 0.3$)	84.5%	**84.5%** (1.91s)	84% (1.81s)	81.5% (5.15s)	17% (128.72s)	16.5% (124.74s)	0% (2.39s)	77.5% (0.10s)
ConvLarge ($\rho = 0.3$)	84%	**82.5%** (7.36s)	**82.5%** (5.05s)	79.5% (32.71s)	0% (150.00s)	0% (150.00s)	0% (8.21s)	77% (1.00s)
CIFAR-10 Models								
CNN-A ($\rho = \frac{24}{255}$)	55.5%	**54%** (8.70s)	51.5% (2.85s)	49% (4.75s)	20% (97.59s)	20% (97.25s)	7.5% (2.20s)	39% (0.20s)
CNN-B ($\rho = \frac{24}{255}$)	59.5%	**58%** (10.99s)	52.5% (6.65s)	49.5% (9.02s)	3% (143.30s)	3% (142.71s)	0% (3.20s)	33% (0.30s)
CNN-C ($\rho = \frac{24}{255}$)	47%	**45.5%** (7.10s)	45% (3.30s)	42.5% (4.76s)	35.5% (39.98s)	35.5% (38.26s)	24.5% (1.91s)	36.5% (0.20s)
ConvSmall ($\rho = \frac{24}{255}$)	52.5%	**51%** (8.04s)	47.5% (3.58s)	43.5% (3.94s)	18% (102.03s)	18% (100.94s)	6% (2.30s)	33% (0.20s)
ConvDeep ($\rho = \frac{24}{255}$)	50%	**49.5%** (7.06s)	47.5% (5.73s)	45.5% (11.19s)	31.5% (59.39s)	30.5% (59.87s)	22.5% (2.53s)	39.5% (0.30s)
ConvLarge ($\rho = \frac{8}{255}$)	72.5%	**70.5%** (14.27s)	66.5% (7.72s)	63.5% (38.51s)	6% (139.77s)	6% (139.02s)	2.5% (11.08s)	47.5% (1.20s)
Non-Lipschitz regularized / Large Lipschitz-constant networks								
MNIST Models								
CNN-small NOR ($\rho = 0.15$)	98%	**86.5%** (26.67s)	69% (10.07s)	42% (8.39s)	30.5% (111.96s)	29.5% (108.78s)	3% (2.34s)	0% (0.40s)
CNN-small ADV ($\rho = 0.25$)	98.5%	**96.5%** (6.99s)	90.5% (4.39s)	33% (7.80s)	3.5% (146.96s)	2.5% (147.47s)	0% (2.46s)	0% (0.40s)
CIFAR-10 Models								
CNN-small NOR ($\rho = \frac{8}{255}$)	75.5%	**61.5%** (40.28s)	51.5% (21.05s)	34.5% (7.42s)	25% (103.88s)	24% (104.56s)	5% (2.33s)	0% (0.36s)
CNN-A-Adv ($\rho = \frac{32}{255}$)	63.5%	**49%** (49.05s)	39.5% (24.57s)	9% (8.04s)	13.5% (122.75s)	12.5% (122.15s)	0.5% (2.51s)	0% (0.40s)
CNN-A-Adv4 ($\rho = \frac{32}{255}$)	53%	**42%** (42.96s)	37.5% (20.90s)	14.5% (7.32s)	13.5% (116.38s)	12.5% (117.83s)	0.5% (2.54s)	0% (0.36s)

Table 1: **Verified accuracy and average runtime on benchmarks.** Bold entries indicate the best verified accuracy. The perturbation size is indicated next to the network name. We can see that our proposed method shows superior performance in all the benchmarks. Impressively, our affine-image SDP-CROWN without branch-and-bound already surpasses strong verifiers that incorporate branch-and-bound on these benchmarks.

since this ball is already nearly tight on these networks. On the non-Lipschitz-regularized networks, our affine-image SDP-CROWN additionally imposes the affine-image constraint on the first layer, with the scaling l optimized jointly, and the original SDP-CROWN is run with the elementwise interval constraint extension described in Theorem C.1 of the original paper (Chiu et al., 2025). We report our method both on its own and within branch-and-bound (Affine-Image SDP-CROWN + BaB). For all branch-and-bound-based verification methods, we use a 150-second timeout per instance.

5.1 Verification accuracy comparison

Table 1 reports the verified accuracy and average runtime of our method and the baseline verifiers under ℓ_2 norm perturbation. The "Upper Bound" column is computed using PGD attacks and upper bounds the maximum achievable verified accuracy. We first evaluate on the Lipschitz-regularized networks used in the previous SDP-CROWN paper. On these benchmarks, our incomplete verifier consistently improves over SDP-CROWN at comparable computational cost, increasing verified accuracy by 2–11.5% across all benchmarks. When integrated with branch-and-bound, our method achieves saturating performance on most benchmarks. Compared with other branch-and-bound-based verifiers such as BICCOS and β-CROWN, our method is often substantially faster while achieving higher certified accuracy, demonstrating a favorable accuracy-efficiency tradeoff.

We further evaluate on non-Lipschitz-regularized networks, which are more challenging for the Lipschitz-based method and also the original SDP-CROWN. On these models, the gap between our method and previous methods becomes more pronounced. Although the original SDP-CROWN can still significantly improve over α-CROWN, the performance is still not as competitive as in the Lipschitz-regularized case, and LipNaive fails completely. In contrast, our method certifies nontrivial robustness across all tested models, achieving close-to-saturating verified accuracy on the MNIST CNN-small NOR and ADV models. On CIFAR-10, our method with branch and bound certifies 61.5% on CNN-small NOR and improves over β-CROWN and BICCOS by more than 3 times on CNN-A-Adv4 and CNN-A-Adv. These results suggest that our method is substantially more robust

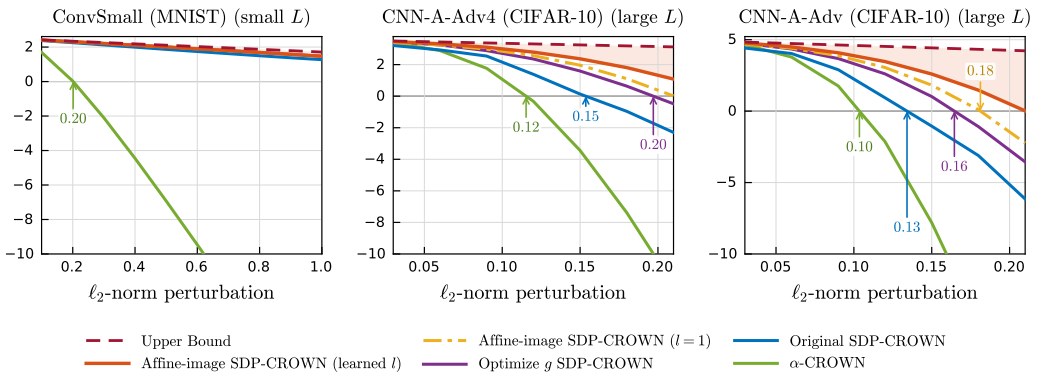


Figure 2: **Lower bounds on the classification margin against ℓ_2 perturbations.** We compare lower bounds on $c^T f(x)$ obtained using our full affine-image SDP-CROWN, affine-image SDP-CROWN with fixed scaling $l = 1$, the original SDP-CROWN with optimized g, the original SDP-CROWN that optimizes only the bias term, and α-CROWN, together with a PGD-based empirical upper bound. The lower bounds are averaged over 20 instances and all 9 incorrect specifications. The annotated numbers indicate where each lower-bound curve intersects zero, corresponding to the average perturbation radius certified by each method. Small L corresponds to Lipschitz-regularized networks, whereas large L corresponds to networks without Lipschitz regularization.

than previous bound-propagation methods on non-Lipschitz-regularized networks when considering l_2 perturbations, where the original SDP-CROWN relaxation becomes overly conservative.

5.2 Lower bound comparison

We further analyze the lower bounds produced by our method and baselines over a wide range of ℓ_2 perturbation radii, as reported in Figure 2. We also include an ablation that compares our full configuration with two reduced variants, one that optimizes the slope g without the affine image, and one that uses the affine image with the scaling fixed at $l = 1$. For each perturbation radius, we compute certified lower bounds on the classification margins $c^T f(x)$. To reduce randomness, we take the average lower bound over 20 input instances and all 9 incorrect-class specifications for each instance. For comparison, we also added an upper bound obtained from projected gradient descent (PGD). As shown in Figure 2, our method has a small advantage over the original SDP-CROWN in Lipschitz-regularized networks. And since the propagated ball is already nearly tight, the ablation variants of our methods essentially coincide. On non-Lipschitz-regularized networks, the advantage of our setting is more pronounced. Although much tighter than α-CROWN, the bounds produced by the original SDP-CROWN is still conservative. In contrast, our proposed method can consistently produce much tighter bounds than both the original SDP-CROWN and α-CROWN, as reflected by both the higher margin curves and the larger zero-crossing radii. In addition, these results reveal our hierarchy more clearly. Optimizing g in the original SDP-CROWN reduces the functional loss and tightens the bounds. The affine-image formulation further reduces the geometric loss, yielding even tighter bounds. Finally, jointly optimizing the scaling factor l within the affine-image formulation further reduces the geometric loss and yields the tightest bounds. Finally, the improvement over the baselines grows as the perturbation radius increases. While all methods are relatively close at small radii, the bounds from α-CROWN and original SDP-CROWN degrade much faster than those from our method. Overall, these results demonstrate that our method provides a significantly tighter relaxation and yields stronger certified robustness margins under ℓ_2 perturbations.

Acknowledgements and funding disclosures

Financial support for this work was provided by NSF CAREER Award ECCS-2047462, and ONR Award N00014-24-1-2671. Huan Zhang is supported in part by the AI2050 program at Schmidt Sciences (AI2050 Early Career Fellowship) and NSF (2525287, 2331967).

References

Ben Barrett, Alexander Camuto, Matthew Willetts, and Tom Rainforth. Certifiably robust variational autoencoders. In *International Conference on Artificial Intelligence and Statistics*, pages 3663–3683. PMLR, 2022.

Ben Batten, Panagiotis Kouvaros, Alessio Lomuscio, and Yang Zheng. Efficient neural network verification via layer-based semidefinite relaxations and linear cuts. In *IJCAI*, pages 2184–2190, 2021.

Robin A Brown, Edward Schmerling, Navid Azizan, and Marco Pavone. A unified view of sdp-based neural network verification through completely positive programming. In *International conference on artificial intelligence and statistics*, pages 9334–9355. PMLR, 2022.

Hong-Ming Chiu and Richard Y Zhang. Tight certification of adversarially trained neural networks via nonconvex low-rank semidefinite relaxations. In *International Conference on Machine Learning*, pages 5631–5660. PMLR, 2023.

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 *International Conference on Machine Learning*, pages 10449–10468. PMLR, 2025.

S. Dathathri, K. Dvijotham, A. Kurakin, A. Raghunathan, J. Uesato, R. R. Bunel, S. Shankar, J. Steinhardt, I. Goodfellow, P. S. Liang, et al. Enabling certification of verification-agnostic networks via memory-efficient semidefinite programming. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020a.

Sumanth Dathathri, Krishnamurthy Dvijotham, Alexey Kurakin, Aditi Raghunathan, Jonathan Uesato, Rudy Bunel, Shreya Shankar, Jacob Steinhardt, Ian Goodfellow, Percy Liang, et al. Enabling certification of verification-agnostic networks via memory-efficient semidefinite programming. In *Advances in Neural Information Processing Systems*, 2020b.

Alessandro De Palma, Harkirat S Behl, Rudy Bunel, Philip Torr, and M Pawan Kumar. Scaling the convex barrier with active sets. In *Proceedings of the ICLR 2021 Conference*. Open Review, 2021.

Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy A Mann, and Pushmeet Kohli. A dual approach to scalable verification of deep networks. In *UAI*, volume 1, page 2, 2018.

Krishnamurthy Dj Dvijotham, Robert Stanforth, Sven Gowal, Chongli Qin, Soham De, and Pushmeet Kohli. Efficient neural network verification with exactness characterization. In *Uncertainty in Artificial Intelligence*, pages 497–507. PMLR, 2020.

Ruediger Ehlers. Formal verification of piece-wise linear feed-forward neural networks. In *International Symposium on Automated Technology for Verification and Analysis*, pages 269–286. Springer, 2017.

Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. *Advances in neural information processing systems*, 32, 2019.

Mahyar Fazlyab, Manfred Morari, and George J Pappas. Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming. *IEEE Transactions on Automatic Control*, 67(1):1–15, 2020.

Claudio Ferrari, Mark Niklas Mueller, Nikola Jovanović, and Martin Vechev. Complete verification via multi-neuron relaxation guided branch-and-bound. In *International Conference on Learning Representations*, 2022.

Vahid Hashemi, Panagiotis Kouvaros, and Alessio Lomuscio. Osip: Tightened bound propagation for the verification of relu neural networks. In *International Conference on Software Engineering and Formal Methods*, pages 463–480. Springer, 2021.

Zhongkui Ma, Jiaying Li, and Guangdong Bai. Relu hull approximation. *Proceedings of the ACM on Programming Languages*, 8(POPL):2260–2287, 2024.

Ziye Ma and Somayeh Sojoudi. Strengthened sdp verification of neural network robustness via non-convex cuts. *arXiv preprint arXiv:2010.08603*, pages 715–727, 2020.

Mark Niklas Müller, Gleb Makarchuk, Gagandeep Singh, Markus Püschel, and Martin Vechev. Prima: general and precise neural network certification via scalable convex hull approximations. *Proceedings of the ACM on Programming Languages*, 6(POPL):1–33, 2022.

Aditi Raghunathan, Jacob Steinhardt, and Percy S Liang. Semidefinite relaxations for certifying robustness to adversarial examples. *Advances in Neural Information Processing Systems*, 31, 2018.

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.

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 *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2025.

Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. In *Advances in Neural Information Processing Systems*, pages 10802–10813, 2018.

Gagandeep Singh, Rupanshu Ganvir, Markus Püschel, and Martin Vechev. Beyond the single neuron convex barrier for neural network certification. *Advances in Neural Information Processing Systems*, 32, 2019a.

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

Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Efficient formal safety analysis of neural networks. *Advances in neural information processing systems*, 31, 2018.

Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. *Advances in Neural Information Processing Systems*, 34:29909–29921, 2021.

Eric Wong and J Zico Kolter. Learning perturbation sets for robust machine learning. In *International Conference on Learning Representations*, 2021.

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 Representation (ICLR)*, 2021.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. *Advances in neural information processing systems*, 31, 2018.

Huan Zhang, Shiqi Wang, Kaidi Xu, Linyi Li, Bo Li, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. General cutting planes for bound-propagation-based neural network verification. *Advances in Neural Information Processing Systems*, 2022.

Richard Y Zhang. On the tightness of semidefinite relaxations for certifying robustness to adversarial examples. In *Advances in Neural Information Processing Systems*, 2020.

Duo Zhou, Christopher Brix, Grani A Hanasusanto, and Huan Zhang. Scalable neural network verification with branch-and-bound inferred cutting planes. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Duo Zhou, Jorge Chavez, Hesun Chen, Grani A Hanasusanto, and Huan Zhang. Clip-and-verify: Linear constraint-driven domain clipping for accelerating neural network verification. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

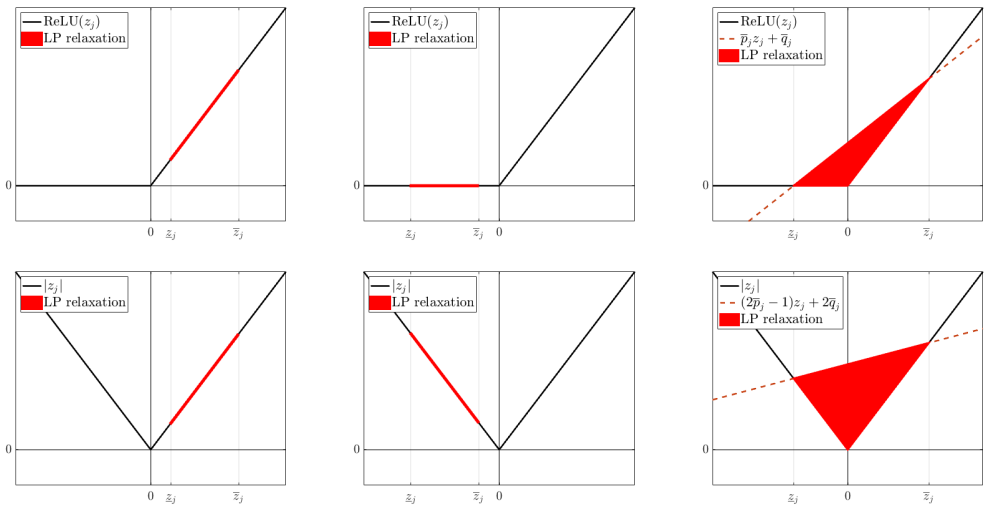


Figure 3: **Comparison of the triangular relaxation (Ehlers, 2017) and the absolute-split relaxation (8). (Left.)** $\underline{z}_j \geq 0$. In this case, $y_j = z_j$ and $r_j = z_j$. **(Middle.)** $\overline{z}_j \leq 0$. In this case, $y_j = 0$ while $r_j = -z_j$. **(Right.)** $\underline{z}_j \leq 0 \leq \overline{z}_j$. In this case, $\mathrm{ReLU}(z_j) \leq y_j \leq \overline{p}_j z_j + \overline{q}_j$ while $|z_j| \leq r_j \leq (2\overline{p}_j - 1)z_j + 2\overline{q}_j$.

A Proof of Theorem 1

We start by introducing our absolute-split SDP relaxation and then provide the proof of our main theorem in this section.

A.1 The absolute-split SDP relaxation

Our derivation of SDP relaxations for $y = \mathrm{ReLU}(z)$ is based on the absolute-split $y = \frac{1}{2}(z + |z|)$. Through this view, it naturally exposes the structure in (5). Given $\underline{z} \leq z \leq \overline{z}$ where $z \in \mathbb{R}^m$, we start by developing a linear relaxation by characterizing the convex hull of $r = |z|$:

$$y_j = \frac{1}{2}(z_j + r_j), \quad |z_j| \leq r_j \leq (2\overline{p}_j - 1)z_j + 2\overline{q}_j \quad \text{for } j = 1, \ldots, m \tag{8}$$

$$\text{where} \quad \overline{p}_j = \begin{cases} 1 & \text{if } \underline{z}_j \geq 0, \\ 0 & \text{if } \overline{z}_j \leq 0, \\ \frac{\overline{z}_j}{\overline{z}_j - \underline{z}_j} & \text{otherwise,} \end{cases} \quad \overline{q}_j = \begin{cases} 0 & \text{if } \underline{z}_j \geq 0, \\ 0 & \text{if } \overline{z}_j \leq 0, \\ -\frac{\overline{z}_j \underline{z}_j}{\overline{z}_j - \underline{z}_j} & \text{otherwise.} \end{cases} \tag{9}$$

We note that the linear relaxation derived from the absolute-split (8) is equivalent to the triangular relaxation $\mathrm{ReLU}(z_j) \leq y_j \leq \overline{p}_j z_j + \overline{q}_j$ of (Ehlers, 2017). We provide an illustration in Figure 3. The absolute-split SDP relaxation can be derived based on the observation that (8) becomes *exact* after adding a nonlinear inequality constraint $y = \mathrm{ReLU}(z) \iff (8), \quad r \odot r \leq z \odot z$.[2]

Given a weight $W \in \mathbb{R}^{m \times n}$ and bias $b \in \mathbb{R}^m$. Substituting $z = Wx + b$ and relaxing $xx^T \preceq X$ and $r_j \cdot r_j \leq R_j$ yield the following SDP relaxation

$$\begin{gathered} (8), \quad R \leq \mathrm{diag}(WXW^T) + 2z \odot b - b \odot b, \quad z = Wx + b, \\ \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0, \quad \begin{bmatrix} 1 & r_j \\ r_j & R_j \end{bmatrix} \succeq 0 \quad \text{for all } j = 1, \ldots, m. \end{gathered} \tag{10}$$

[2]Though the linear bound $r \leq (2\overline{p} - 1) \odot z + 2\overline{q}$ becomes redundant after adding $r \odot r \leq z \odot z$, it will ensure the subsequent absolute-split SDP relaxation will always be at least as tight, and potentially strictly tighter than the absolute-split LP relaxation (8). This is commonly refer as the strengthening constraints for SDP relaxations (Brown et al., 2022)

A.2 Proof of Theorem 1

Theorem 1 can be proved by deriving the SDP relaxation of the following

$$\min_{z\in\mathbb{R}^m} c^T \operatorname{ReLU}(z) \quad \text{s.t.} \quad z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d_z) \cap \{Wx + b \mid x \in \mathcal{E}_2(\hat{x}, d_x)\}.$$

We start by applying a variable substitution $x \equiv x + \hat{x}$ and $\hat{z} = W\hat{x} + b$ so that the constraints on z become

$$z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d_z) \cap \{Wx + \hat{z} \mid x \in \mathcal{E}_2(0, d_x)\}$$

and then applying the absolute split SDP relaxation (10) to yield

$$\begin{aligned} \min_{\substack{x\in\mathbb{R}^n, X\in\mathbb{S}^n \\ z,r,R\in\mathbb{R}^m}} \frac{1}{2}c^T(z+r) \quad \text{s.t.} \quad & |z| \le r \le (2\overline{p}-1)\odot z + 2\overline{q}, \quad z = Wx + \hat{z}, \\ & R \le \operatorname{diag}(WXW^T) + 2z\odot\hat{z} - \hat{z}\odot\hat{z}, \\ & \operatorname{tr}(\operatorname{diag}(d_x)^{-2}X) \le 1, \quad \operatorname{tr}(\operatorname{diag}(d_z)^{-2}WXW^T) \le 1, \\ & \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0, \quad \begin{bmatrix} 1 & r_j \\ r_j & R_j \end{bmatrix} \succeq 0 \quad \text{for } j = 1, \dots, m \end{aligned} \tag{11}$$

Notice that the SDP relaxations for $x \in \mathcal{E}_2(0, d_x) \iff \operatorname{tr}(\operatorname{diag}(d_x)^{-1}xx^T\operatorname{diag}(d_x)^{-1}) \le 1$ and $z \in \mathcal{E}_2(\hat{z}, d_z) \iff \operatorname{tr}(\operatorname{diag}(d_z)^{-1}Wxx^TW^T\operatorname{diag}(d_z)^{-1}) \le 1$ are obtained by relaxing $xx^T \preceq X$. The core insight to prove Theorem 1 is to fix any $x \in \mathcal{E}_2(0, d_x)$ and $z = Wx + \hat{z} \in \tilde{\mathcal{Z}}$ where $\tilde{\mathcal{Z}}$ is defined in (4) with $p = 2$, then take the dual to lower bound the above. In particular, the Lagrangian dual is given by:

$$\begin{aligned} \max_{\substack{\mu,s,t,\tau,\lambda\in\mathbb{R}^m_+ \\ \sigma,\eta\in\mathbb{R}_+}} \; & g^Tz - 2\tau^T\overline{q} + \frac{1}{2}x^TMx - \frac{1}{2}\left(\sigma + \eta - \hat{z}^T\operatorname{diag}(\lambda)\hat{z} + \mu^T\mathbf{1}\right) \\ \text{s.t.} \quad & g = \frac{1}{2}c + s - t - (2\overline{p}-1)\odot\tau - \lambda\odot\hat{z}, \quad \phi = \frac{1}{2}c - s - t + \tau, \\ & M = \sigma\operatorname{diag}(d_x)^{-2} - W^T\left[\operatorname{diag}(\lambda) - \eta\operatorname{diag}(d_z)^{-2}\right]W \succeq 0, \\ & \mu_j\lambda_j \ge \phi_j^2 \quad \text{for } j = 1, \dots, m. \end{aligned} \tag{12}$$

We defer the detailed derivation for the dual problem (12) to Appendix C. The dual variable M is fully dense, and therefore $M \succeq 0$ cannot be enforced efficiently. We propose to instead optimize over the scaled diagonal-dominance surrogate for M:

$$l > 0, \quad \sigma\mathbf{1} \ge \gamma\Big(\underbrace{\operatorname{diag}(d_x)W^T\left[\operatorname{diag}(\lambda) - \eta\operatorname{diag}(d_z)^{-2}\right]W\operatorname{diag}(d_x)}_{M(\lambda,\eta)}, l\Big) \implies M \succeq 0$$

where γ is the scaled Gershgorin operation defined in (7). Since setting $\sigma = \|\gamma(M(\lambda,\eta), l)\|_\infty$ ensures $M \succeq 0$, (12) can be relaxed by setting $\sigma = \|\gamma(M(\lambda,\eta), l)\|_\infty$ and observing that

$$\begin{aligned} \frac{1}{2}(x^TMx - \sigma) &= \frac{1}{2}\left(\operatorname{diag}(d_x)^{-1}x\right)^T\left(\sigma I - M(\lambda,\eta)\right)\left(\operatorname{diag}(d_x)^{-1}x\right) - \frac{1}{2}\sigma \\ &\ge -\frac{1}{2}\|\gamma(M(\lambda,\eta), l)\|_\infty. \end{aligned}$$

Hence, the diagonal-dominant surrogate yields the following relaxation for (12)

$$\begin{aligned} \max_{\substack{\mu,s,t,\tau,\lambda\in\mathbb{R}^m_+ \\ l\in\mathbb{R}^n_{++},\eta\in\mathbb{R}_+}} \; & g^Tz - 2\tau^T\overline{q} - \frac{1}{2}\left(\eta + \|\gamma(M(\lambda,\eta), l)\|_\infty - \hat{z}^T\operatorname{diag}(\lambda)\hat{z} + \mu^T\mathbf{1}\right) \\ \text{s.t.} \quad & g = \frac{1}{2}c + s - t - (2\overline{p}-1)\odot\tau - \lambda\odot\hat{z}, \quad \phi = \frac{1}{2}c - s - t + \tau, \\ & \mu_j\lambda_j \ge \phi_j^2 \quad \text{for } j = 1, \dots, m \end{aligned}$$

The variables $s, t \ge 0$ can be eliminated by applying the following variable transformation

$$s \ge 0, \quad t \ge 0 \quad \iff \quad \theta = s + t, \quad \omega = s - t, \quad \theta \ge |\omega|. \tag{13}$$

Notice that each θ_j in ϕ_j can be optimized in closed form as

$$\min_{\theta_j \ge |\omega_j|} \phi_j^2 = \min\{c_j/2 - |\omega_j| + \tau_j,\ 0\}^2 = \min\{c_j/2 - \omega_j + \tau_j,\ c_j/2 + \omega_j + \tau_j,\ 0\}^2$$

and ω_j can be eliminated by substituting $g_j = c_j/2 + \omega_j - \tau_j(2\overline{p}_j - 1) - \lambda_j \hat{z}_j$ into ϕ_j to yield

$$\phi_j = \min\{c_j - g_j - 2\tau_j(\overline{p}_j - 1) - \lambda_j \hat{z}_j,\ g_j + 2\tau_j \overline{p}_j + \lambda_j \hat{z}_j,\ 0\}.$$

Finally, applying the above two variable substitutions yields

$$\max_{\substack{\mu,\tau,\lambda\in\mathbb{R}^m_+, g\in\mathbb{R}^m \\ l\in\mathbb{R}^n_{++},\eta\in\mathbb{R}_+}} g^T z - 2\tau^T \overline{q} - \frac{1}{2}\left(\eta + \|\gamma(M(\lambda,\eta),l)\|_\infty - \hat{z}^T \operatorname{diag}(\lambda)\hat{z} + \mu^T \mathbf{1}\right)$$
$$\text{s.t.} \quad \phi_j = \min\{c_j - g_j - 2\tau_j(\overline{p}_j - 1) - \lambda_j \hat{z}_j,\ g_j + 2\tau_j \overline{p}_j + \lambda_j \hat{z}_j,\ 0\}, \quad \mu_j \lambda_j \geq \phi_j^2$$

for $j = 1, \ldots, m$. We are now ready to prove Theorem 1.

Proof. Rescaling $\tau \equiv \frac{1}{2}\tau$, the desired result in Theorem 1 follows from fixing any $\boldsymbol{\lambda} = (\lambda, \tau, l, \eta) \in \mathbb{R}^m_+ \times \mathbb{R}^m_+ \times \mathbb{R}^n_{++} \times \mathbb{R}_+$ and optimizing over μ

$$\max_{g,\boldsymbol{\lambda}} g^T z + h(g, \boldsymbol{\lambda})$$
$$\text{s.t.} \ \ h(g,\boldsymbol{\lambda}) = -\tau^T \overline{q} - \frac{1}{2} \cdot \left(\eta + \|\gamma(M(\lambda,\eta),l)\|_\infty + \sum_{j=1}^m \frac{\phi_j^2}{\lambda_j} - \lambda_j \hat{z}_j^2\right)$$
$$\phi_j = \min\{c_j - g_j - \tau_j(\overline{p}_j - 1) - \lambda_j \hat{z}_j,\ g_j + \tau_j \overline{p}_j + \lambda_j \hat{z}_j,\ 0\}.$$

for $j = 1, \ldots, m$. Since for every fixed $z \in \tilde{\mathcal{Z}}$ the above is a lower bound of $c^T \operatorname{ReLU}(z)$, it then follows that $g^T z + h(g, \boldsymbol{\lambda}) \leq c^T \operatorname{ReLU}(z)$ for all $z \in \tilde{\mathcal{Z}}$. □

B Extension for our method

In this section, we show that Theorem 1 can be extended to handle an affine image under ℓ_p-norm ellipsoid

$$\{Wx + b \mid x \in \mathcal{E}_p(\hat{x}, d_x)\}$$

for any $1 \leq p \leq \infty$. We note that when $p = \infty$, the ℓ_∞-norm ellipsoid is equivalent to an elementwise interval $\mathcal{E}_\infty(\hat{x}, d_x) = [\hat{x} - d_x, \hat{x} + d_x]$, and therefore it automatically captures the elementwise input set. In the following corollary, we develop a linear lower bound on (4) for $1 \leq p \leq \infty$.

Corollary 1. *Given $c \in \mathbb{R}^m$, an affine-image $\{Wx + b \mid x \in \mathcal{E}_p(\hat{x}, d_x)\}$ with $W \in \mathbb{R}^{m\times n}$ and $1 \leq p \leq \infty$, an ellipsoid $\mathcal{E}_2(\hat{z}, d_z)$ with $\hat{z} = W\hat{x} + b$, and an interval $[\underline{z}, \overline{z}]$ with $\underline{z}, \overline{z} \in \mathbb{R}^m$. The following holds*

$$g^T z + h(g, \boldsymbol{\lambda}) \leq c^T \operatorname{ReLU}(z) \quad \textit{for all } z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d_z) \cap \{Wx + b \mid x \in \mathcal{E}_p(\hat{x}, d_x)\}$$

for any $g \in \mathbb{R}^m$, $\boldsymbol{\lambda} = (\lambda, \tau, l, \eta) \in \mathbb{R}^m_+ \times \mathbb{R}^m_+ \times \mathbb{R}^n_{++} \times \mathbb{R}_+$, and

$$h(g,\boldsymbol{\lambda}) = -\tau^T \overline{q} - \frac{1}{2}\left[\eta + \|\gamma(M(\lambda,\eta),l)\|_q + \sum_{j=1}^m \left(\frac{\phi_j^2}{\lambda_j} - \lambda_j \hat{z}_j^2\right)\right], \quad \frac{1}{\max\{1, p/2\}} + \frac{1}{q} = 1$$

where γ is the scaled Gershgorin operation defined in (7), $\overline{q}$ is defined in (9), $M(\lambda, \eta)$ and ϕ are defined in Theorem 1.

B.1 Proof of Corollary 1

The proof of Corollary 1 follows the same structure as the proof of Theorem 1. We start by deriving the SDP relaxation of the following

$$\min_{z\in\mathbb{R}^m} c^T \operatorname{ReLU}(z) \quad \text{s.t.} \quad z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d_z) \cap \{Wx + b \mid x \in \mathcal{E}_p(\hat{x}, d_x)\}.$$

The SDP relaxation of the above is given by replacing the constraint $\operatorname{tr}(\operatorname{diag}(d_x)^{-1} X \operatorname{diag}(d_x)^{-1}) \leq 1$ in the primal problem (11) with

$$\begin{cases} \| \operatorname{diag}\left(\operatorname{diag}(d_x)^{-1} X \operatorname{diag}(d_x)^{-1}\right) \|_{p/2} \leq 1 & \text{if } 2 \leq p \leq \infty \\ \| \operatorname{diag}\left(\operatorname{diag}(d_x)^{-1} X \operatorname{diag}(d_x)^{-1}\right) \|_1 \leq 1 & \text{if } 1 \leq p < 2. \end{cases}$$

To maintain convexity, we use the ℓ_1 unit ball when $1 \le p < 2$, which is the convex hull of $\ell_{p/2}$ quasi-norm unit ball for $1 \le p < 2$. The proof of Corollary 1 follows from fixing any $x \in \mathcal{E}_p(0, d_x)$ and $z = Wx + \hat{z} \in \tilde{\mathcal{Z}}$ where $\tilde{\mathcal{Z}}$ is defined in (4), and taking the dual to lower bound the primal problem. To see how its dual differ from (12), let $\sigma \in \mathbb{R}^n$ be the dual variable of $v = \operatorname{diag}\left(\operatorname{diag}(d_x)^{-1} X \operatorname{diag}(d_x)^{-1}\right)$, the Lagrangian is

$$\left\langle \sigma, \operatorname{diag}\left(\operatorname{diag}(d_x)^{-1} X \operatorname{diag}(d_x)^{-1}\right) - v \right\rangle = \left\langle X, \operatorname{diag}(d_x)^{-1} \operatorname{diag}(\sigma) \operatorname{diag}(d_x)^{-1} \right\rangle - \sigma^T v.$$

Following the same derivation in Appendix C, the first term in the Lagrangian contributes to the PSD constraints

$$\operatorname{diag}(\sigma) \operatorname{diag}(d_x)^{-2} - W^T \left[\operatorname{diag}(\lambda) - \eta \operatorname{diag}(d_z)^{-2}\right] W \succeq 0,$$

where η is the dual variable associated with the constraint $\operatorname{tr}(\operatorname{diag}(d_z)^{-2} W X W^T) \le 1$ in (11).

For the second term $-\sigma^T v$ in the Lagrangian, notice that $\|v\|_{\max\{1,p/2\}} \le 1$, taking the infimum over $\|v\|_{\max\{1,p/2\}} \le 1$ yields

$$\inf_{\|v\|_{\max\{1,p/2\}} \le 1} -\sigma^T v = -\|\sigma\|_q, \quad \text{where} \quad \frac{1}{\max\{1, p/2\}} + \frac{1}{q} = 1$$

Rescaling $\sigma \equiv \frac{1}{2}\sigma$ and $\eta \equiv \frac{1}{2}\eta$, the Lagrangian dual is given by:

$$\begin{aligned} \max_{\substack{\mu,s,t,\tau,\lambda \in \mathbb{R}^m_+ \\ \sigma \in \mathbb{R}^n, \eta \in \mathbb{R}_+}} \; & g^T z - 2\tau^T \overline{q} + \frac{1}{2} x^T M x - \frac{1}{2}\left(\|\sigma\|_q + \eta - \hat{z}^T \operatorname{diag}(\lambda) \hat{z} + \mu^T \mathbf{1}\right) \\ \text{s.t.} \quad & g = \frac{1}{2} c + s - t - (2\overline{p} - 1) \odot \tau - \lambda \odot \hat{z}, \quad \phi = \frac{1}{2} c - s - t + \tau, \\ & M = \operatorname{diag}(\sigma) \operatorname{diag}(d_x)^{-2} - W^T \left[\operatorname{diag}(\lambda) - \eta \operatorname{diag}(d_z)^{-2}\right] W \succeq 0, \\ & \mu_j \lambda_j \ge \phi_j^2 \quad \text{for } j = 1, \ldots, m. \end{aligned} \tag{14}$$

Again, the dual variable M is fully dense, and therefore $M \succeq 0$ cannot be enforced efficiently. We propose to instead optimize over the scaled diagonal-dominance surrogate for M

$$l > 0, \quad \sigma \ge \gamma\Big(\underbrace{\operatorname{diag}(d_x) W^T \left[\operatorname{diag}(\lambda) - \eta \operatorname{diag}(d_z)^{-2}\right] W \operatorname{diag}(d_x)}_{M(\lambda,\eta)}, l\Big) \implies M \succeq 0$$

where γ is the scaled Gershgorin operation defined in (7). Since setting $\sigma = \gamma(M(\lambda, \eta), l)$ ensures $M \succeq 0$, (14) can be relaxed by setting $\sigma = \gamma(M(\lambda, \eta), l)$ and observe that

$$\begin{aligned} \frac{1}{2}(x^T M x - \|\sigma\|_q) &= \frac{1}{2}\left(\operatorname{diag}(d_x)^{-1} x\right)^T \left(\operatorname{diag}(\sigma) - M(\lambda, \eta)\right) \left(\operatorname{diag}(d_x)^{-1} x\right) - \frac{1}{2}\|\sigma\|_q \\ &\ge -\frac{1}{2}\|\gamma(M(\lambda, \eta), l)\|_q. \end{aligned}$$

Hence, the diagonal-dominant surrogate yields the following relaxation for (14)

$$\begin{aligned} \max_{\substack{\mu,s,t,\tau,\lambda \in \mathbb{R}^m_+ \\ l \in \mathbb{R}^n_{++}, \eta \in \mathbb{R}_+}} \; & g^T z - 2\tau^T \overline{q} - \frac{1}{2}\left(\eta + \|\gamma(M(\lambda, \eta), l)\|_q - \hat{z}^T \operatorname{diag}(\lambda) \hat{z} + \mu^T \mathbf{1}\right) \\ \text{s.t.} \quad & g = \frac{1}{2} c + s - t - (2\overline{p} - 1) \odot \tau - \lambda \odot \hat{z}, \quad \phi = \frac{1}{2} c - s - t + \tau, \\ & \mu_j \lambda_j \ge \phi_j^2 \quad \text{for } j = 1, \ldots, m. \end{aligned}$$

The desired result in Corollary 1 follows by applying (13) to eliminate s and t, and then optimize over μ.

C Derivation for the dual problem

We describe how to derive the dual problem (12). Recall that we have the primal problem

$$\min_{\substack{x\in\mathbb{R}^n, X\in\mathbb{S}^n \\ z,r,R\in\mathbb{R}^m}} \frac{1}{2}c^T(z+r) \quad \text{s.t.} \quad \begin{array}{c} |z| \le r \le (2\overline{p}-1)\odot z + 2\overline{q}, \quad z = Wx+\hat{z}, \\ R \le \operatorname{diag}(WXW^T) + 2z\odot\hat{z} - \hat{z}\odot\hat{z}, \\ \operatorname{tr}(\operatorname{diag}(d_x)^{-2}X) \le 1, \quad \operatorname{tr}(\operatorname{diag}(d_z)^{-2}WXW^T) \le 1, \\ \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0, \quad \begin{bmatrix} 1 & r_j \\ r_j & R_j \end{bmatrix} \succeq 0 \quad \text{for } j=1,\dots,m \end{array}$$

Let $s,t,\tau \in \mathbb{R}^m_+$ denote the dual variables for $z \le r$, $-z \le r$ and $r \le (2\overline{p}-1)\odot z + 2\overline{q}$, respectively. Let $\lambda \in \mathbb{R}^m_+$ denote the dual variable of $R \le \operatorname{diag}(WXW^T) + 2z\odot\hat{z} - \hat{z}\odot\hat{z}$. Let $\sigma,\eta \ge 0$ denote the dual variables for $\operatorname{tr}(\operatorname{diag}(d_x)^{-2}X) \le 1$ and $\operatorname{tr}(\operatorname{diag}(d_z)^{-2}WXW^T) \le 1$, respectively. Finally, let $\begin{bmatrix} \nu & y^T \\ y & \tilde{X} \end{bmatrix} \succeq 0$ and $\begin{bmatrix} \mu_i & \tilde{r}_i^T \\ \tilde{r}_i & \tilde{R}_i \end{bmatrix} \succeq 0$ denote the dual variables for the PSD blocks $\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0$, $\begin{bmatrix} 1 & r_i \\ r_i & R_i \end{bmatrix} \succeq 0$, respectively. The Lagrangian is given by:

$$\begin{aligned} &-2\tau^T\overline{q} - \sigma - \eta + \hat{z}^T\operatorname{diag}(\lambda)\hat{z} - \mu^T\mathbf{1} - \nu \\ &+\left\langle Wx+\hat{z}, \underbrace{\frac{1}{2}c + s - t - \tau\odot(2\overline{p}-1) - 2\lambda\odot\hat{z}}_{g} \right\rangle - \langle 2y, x\rangle \\ &+\left\langle X, \sigma\operatorname{diag}(d_x)^{-2} - W^T\left[\operatorname{diag}(\lambda) - \eta\operatorname{diag}(d_z)^{-2}\right]W - \tilde{X}\right\rangle \\ &+\left\langle r, \underbrace{\frac{1}{2}c - s - t + \tau}_{\phi} \right\rangle - \langle r, 2\tilde{r}\rangle + \left\langle R, \lambda - \tilde{R}\right\rangle. \end{aligned}$$

Fixing x and $z = Wx+\hat{z}$, and taking Lagrangian dual yield

$$\phi - 2\tilde{r} = 0, \quad \lambda - \tilde{R} = 0, \quad \sigma\operatorname{diag}(d_x)^{-2} - W^T\left[\operatorname{diag}(\lambda) - \eta\operatorname{diag}(d_z)^{-2}\right]W - \tilde{X} = 0,$$

which gives

$$\begin{aligned} \max_{\substack{\mu,s,t,\tau,\lambda\in\mathbb{R}^m_+ \\ y\in\mathbb{R}^n,\sigma,\eta,\nu\in\mathbb{R}_+}} \quad & g^Tz - 2y^Tx - 2\tau^T\overline{q} - \sigma - \eta + \hat{z}^T\operatorname{diag}(\lambda)\hat{z} - \mu^T\mathbf{1} - \nu \\ \text{s.t.} \quad & g = \frac{1}{2}c + s - t - (2\overline{p}-1)\odot\tau - 2\lambda\odot\hat{z}, \quad \phi = \frac{1}{2}c - s - t + \tau, \\ & M = \sigma\operatorname{diag}(d_x)^{-2} - W^T\left[\operatorname{diag}(\lambda) - \eta\operatorname{diag}(d_z)^{-2}\right]W, \\ & \begin{bmatrix} \nu & y^T \\ y & M \end{bmatrix} \succeq 0, \quad \begin{bmatrix} \mu_j & \phi_j/2 \\ \phi_j/2 & \lambda_j \end{bmatrix} \succeq 0 \quad \text{for } j=1,\dots,m. \end{aligned}$$

It then follows from the identity that for any $M \succeq 0$

$$x^TMx = \max_{\nu,y} \; -2y^Tx - \nu \quad \text{s.t.} \quad \begin{bmatrix} \nu & y^T \\ y & M \end{bmatrix} \succeq 0$$

that ν and y can be eliminated to yield

$$\begin{aligned} \max_{\substack{\mu,s,t,\tau,\lambda\in\mathbb{R}^m_+ \\ \sigma,\eta\in\mathbb{R}_+}} \quad & g^Tz + x^TMx - 2\tau^T\overline{q} - \sigma - \eta + \hat{z}^T\operatorname{diag}(\lambda)\hat{z} - \mu^T\mathbf{1} \\ \text{s.t.} \quad & g = \frac{1}{2}c + s - t - (2\overline{p}-1)\odot\tau - 2\lambda\odot\hat{z}, \quad \phi = \frac{1}{2}c - s - t + \tau, \\ & M = \sigma\operatorname{diag}(d_x)^{-2} - W^T\left[\operatorname{diag}(\lambda) - \eta\operatorname{diag}(d_z)^{-2}\right]W \succeq 0, \\ & \begin{bmatrix} \mu_j & \phi_j/2 \\ \phi_j/2 & \lambda_j \end{bmatrix} \succeq 0 \quad \text{for } j=1,\dots,m. \end{aligned}$$

Rescaling $\sigma \equiv \frac{1}{2}\sigma$, $\eta \equiv \frac{1}{2}\eta$, $\lambda \equiv \frac{1}{2}\lambda$ and $\mu \equiv \frac{1}{2}\mu$ to yield

$$\begin{aligned}
\max_{\substack{\mu,s,t,\tau,\lambda\in\mathbb{R}^m_+ \\ \sigma,\eta\in\mathbb{R}_+}} \; & g^T z - 2\tau^T \overline{q} + \frac{1}{2} x^T M x - \frac{1}{2}\left(\sigma + \eta - \hat{z}^T \operatorname{diag}(\lambda)\hat{z} + \mu^T \mathbf{1}\right) \\
\text{s.t.} \quad & g = \frac{1}{2}c + s - t - (2\overline{p} - 1) \odot \tau - \lambda \odot \hat{z}, \quad \phi = \frac{1}{2}c - s - t + \tau, \\
& M = \sigma \operatorname{diag}(d_x)^{-2} - W^T \left[\operatorname{diag}(\lambda) - \eta \operatorname{diag}(d_z)^{-2}\right] W \succeq 0, \\
& \begin{bmatrix} \mu_j & \phi_j \\ \phi_j & \lambda_j \end{bmatrix} \succeq 0 \quad \text{for } j = 1, \ldots, m.
\end{aligned}$$

Notice that for a symmetric 2×2 matrix, $X \succeq 0$ if and only if $\det(X) \geq 0$ and $\operatorname{tr}(X) \geq 0$. Applying this insight yields the desired dual in (12)

$$\begin{aligned}
\max_{\substack{\mu,s,t,\tau,\lambda\in\mathbb{R}^m_+ \\ \sigma,\eta\in\mathbb{R}_+}} \; & g^T z - 2\tau^T \overline{q} + \frac{1}{2} x^T M x - \frac{1}{2}\left(\sigma + \eta - \hat{z}^T \operatorname{diag}(\lambda)\hat{z} + \mu^T \mathbf{1}\right) \\
\text{s.t.} \quad & g = \frac{1}{2}c + s - t - (2\overline{p} - 1) \odot \tau - \lambda \odot \hat{z}, \quad \phi = \frac{1}{2}c - s - t + \tau, \\
& M = \sigma \operatorname{diag}(d_x)^{-2} - W^T \left[\operatorname{diag}(\lambda) - \eta \operatorname{diag}(d_z)^{-2}\right] W \succeq 0, \\
& \mu_j \lambda_j \geq \phi_j^2 \quad \text{for } j = 1, \ldots, m.
\end{aligned}$$

D Experimental details

D.1 Computational resources

All experiments were run on a server equipped with one AMD engineering-sample CPU (32 physical cores, 64 threads, 128 MB L3 cache) and four NVIDIA RTX 4090 GPUs.

D.2 Model architecture

The Lipschitz-regularized networks are taken directly from the previous SDP-CROWN (Chiu et al., 2025), whereas the non-Lipschitz-regularized networks are taken from (Salman et al., 2019; Dathathri et al., 2020a). Detailed model architectures are listed in Table 2. We also provided naive Lipschitz constant estimation by multiplying the spectral norm of each weight matrix. Several convolutional layers in Table 2 exhibit a diagonal-dominant structure.

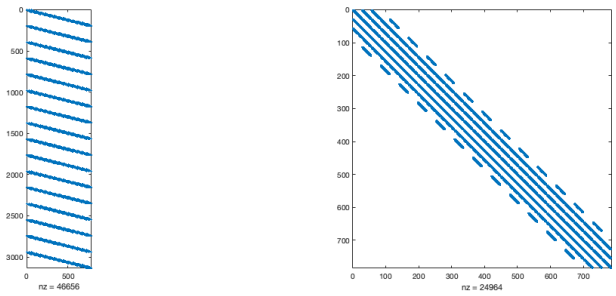


Figure 4: **Diagonal-dominant structure in convolutional layers. (Left.)** The sparsity pattern of the weight matrix W for Conv(1, 16, 4, 2, 1), which is the first layer of CNN-small NOR (MNIST) and CNN-small ADV (MNIST) in Table 2. **(Right.)** The sparsity pattern of $W^T W$ exhibits a diagonal-dominant structure, with nonzero entries concentrated near the diagonal. The matrix $M(\lambda, \eta)$ inherits this banded sparsity structure.

Table 2: **Models used in our experiments.** Conv(c_{in}, c_{out}, k, s, p) denotes a convolutional layer with c_{in} input channels, c_{out} output channels, kernel size $k \times k$, stride s, and padding p. Linear(d_{in}, d_{out}) denotes a fully connected layer with d_{in} input features and d_{out} output features. ZeroPad2d(p_l, p_r, p_t, p_b) adds p_l, p_r, p_t, and p_b pixels of zero padding to the left, right, top, and bottom of each input channel, respectively. There are no max-pooling or average-pooling layers in these models, and ReLU activations are applied between consecutive layers, except after ZeroPad2d.

Model name	Model structure	Accuracy	Lipschitz
Small Lipschitz-constant networks			
MNIST Models			
MLP (MNIST)	Linear(784, 100) - Linear(100, 100) - Linear(100, 10)	79%	0.98
ConvSmall (MNIST)	Conv(1, 16, 4, 2, 1) - Conv(16, 32, 4, 2, 1) - Linear(1568, 100) - Linear(100, 10)	87.5%	1.14
ConvLarge (MNIST)	Conv(1, 32, 3, 1, 1) - Conv(32, 32, 4, 2, 1) - Conv(32, 64, 3, 1, 1) - Conv(64, 64, 4, 2, 1) - Linear(3136, 512) - Linear(512, 512) - Linear(512, 10)	89.5%	1.21
CIFAR-10 Models			
CNN-A (CIFAR-10)	Conv(3, 16, 4, 2, 1) - Conv(16, 32, 4, 2, 1) - Linear(2048, 100) - Linear(100, 10)	62%	0.99
CNN-B (CIFAR-10)	ZeroPad2d(1, 2, 1, 2) - Conv(3, 32, 5, 2, 0) - Conv(32, 128, 4, 2, 1) - Linear(8192, 250) - Linear(250, 10)	66%	1.53
CNN-C (CIFAR-10)	Conv(3, 8, 4, 2, 0) - Conv(8, 16, 4, 2, 0) - Linear(576, 128) - Linear(128, 64) - Linear(64, 10)	51%	1.06
ConvSmall (CIFAR-10)	Conv(3, 16, 4, 2, 0) - Conv(16, 32, 4, 2, 0) - Linear(1152, 100) - Linear(100, 10)	60.5%	1.52
ConvDeep (CIFAR-10)	Conv(3, 8, 4, 2, 1) - Conv(8, 8, 3, 1, 1) - Conv(8, 8, 3, 1, 1) - Conv(8, 8, 4, 2, 1) - Linear(512, 100) - Linear(100, 10)	53%	0.79
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)	74%	5.13
Large Lipschitz-constant networks			
MNIST Models			
CNN-small NOR (MNIST)	Conv(1, 16, 4, 2, 1) - Conv(16, 32, 4, 2, 1) - Linear(1568, 100) - Linear(100, 10)	98.5%	1132.08
CNN-small ADV (MNIST)	Conv(1, 16, 4, 2, 1) - Conv(16, 32, 4, 2, 1) - Linear(1568, 100) - Linear(100, 10)	98.5%	513.95
CIFAR-10 Models			
CNN-small NOR (CIFAR-10)	Conv(3, 16, 4, 2, 1) - Conv(16, 32, 4, 2, 1) - Linear(2048, 100) - Linear(100, 10)	80.5%	325.31
CNN-A-Adv (CIFAR-10)	Conv(3, 16, 4, 2, 1) - Conv(16, 32, 4, 2, 1) - Linear(2048, 100) - Linear(100, 10)	70.5%	632.41
CNN-A-Adv4 (CIFAR-10)	Conv(3, 16, 4, 2, 1) - Conv(16, 32, 4, 2, 1) - Linear(2048, 100) - Linear(100, 10)	60.5%	445.97

E Limitations

Our method currently supports only ReLU networks. Extending it to general activations and architectures remains future work. Although Theorem 1 permits general affine-image enclosures, our implementation uses each layer's weight matrix as W and inherits approximation error from the preceding layer. Efficiently constructing and certifying tighter enclosures using information from multiple layers remains a future direction.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly stated our contirbution in the abstract and introduction. The contribution of this paper is to develop novel bound propagation method for neural network verification.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Limitations are discussed in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide a complete proof for all the Theorem in this paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We clearly state all the experimental setting in the paepr.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We plan to release the full implementation and experimental code upon acceptance of the paper. We are also willing to provide additional implementation details, experimental settings, and clarification regarding the evaluation protocol should the reviewers request them.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: All the experimental setting are specified.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: There are no error bars to be reported for our main experiments.

Guidelines: All the information about the statistical significance of our experiments are correct defined.

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide detailed information on the CPU/GPUs being used in our study in the Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed. The study is very theoretical, and we do not foresee any potential harm to society.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the assets used in the paper are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.