SEMIDEFINITE BOUND PROPAGATION FOR NEURAL NETWORK VERIFICATION WITH GENERAL ACTIVATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Linear bound propagation enables scalable neural network verification by propagating linear bounds through activation functions, but these bounds can lose tightness for general activations. The semidefinite bound propagation method SDP-CROWN strengthens these bounds while retaining efficient linear bound propagation, but its construction relies on a quadratic outer approximation derived from ReLU's piecewise linear structure. We extend semidefinite bound propagation to general activations using quadratic outer approximations derived from sector bounds and quadratic bounds. Our construction converts these constraints into optimizable linear bounds over ellipsoidal domains, which we refine recursively using the same semidefinite bound-propagation framework. The resulting linear bounds integrate directly into the standard bound-propagation recursion, enabling scalable verification against both ℓ_∞ and ℓ_2 adversaries. Our experiments demonstrate substantial improvements in verification performance across all benchmarks with general activations. On representative benchmarks, our method improves verified accuracy over GenBaB from 12% to 57% on MNIST under ℓ_2 perturbations and from 23% to 87% on CIFAR-10 under ℓ_∞ perturbations.

1 INTRODUCTION

Verifying modern neural networks requires methods that support nonlinear operations beyond ReLU. Transformers apply softmax to scaled query-key inner products to compute attention weights and use LayerNorm to normalize hidden representations (Vaswani et al., 2017). Gated recurrent networks combine sigmoid gates with tanh activations (Hochreiter & Schmidhuber, 1997), while neural controllers for physical systems commonly use tanh to obtain smooth control laws (Li et al., 2026).

However, advances in scalable neural network verification have largely focused on ReLU networks, whose structure favors both linear relaxation and branching. In particular, over an interval, the per-neuron convex hull of ReLU admits an exact linear outer approximation (Ehlers, 2017), which can be represented with finitely many linear inequalities. Moreover, splitting at zero fixes the activation phase, making ReLU linear on each branch. *Linear bound propagation* (Xu et al., 2021) exploits this structure by propagating optimized linear bounds derived from the per-neuron convex hull. Combined with branching on activation phases (Wang et al., 2021), this approach enables scalable verification of large ReLU networks and has driven strong performance in verification competitions (Brix et al., 2023).

Linear bound propagation and branch-and-bound have been extended to general activations (Zhang et al., 2018; Shi et al., 2025), but the structural properties that make these methods effective for ReLU are not generally shared by general activations. For instance, the per-neuron convex hull of smooth activations has curved boundaries that no finite collection of linear inequalities can represent exactly, introducing an additional source of relaxation error. Branching can reduce this error, but activations without a piecewise linear structure remain nonlinear after splitting. Consequently, linear bound propagation is weakened for general activations, due to the inability to capture activation curvature through linear outer approximation, motivating richer activation relaxations that capture curvature while preserving the computational efficiency of bound propagation.

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

Recently, linear bound propagation has been extended to *semidefinite bound propagation* (Chiu et al., 2025). The key insight is that ReLU's quadratic outer approximation, $y = \mathrm{ReLU}(z) \iff y \geq 0,\ y \geq z,\ y(y-z) \leq 0$, can be used to derive linear bounds over an ellipsoidal domain through a semidefinite programming relaxation. This strengthens the propagated linear bounds while retaining the scalability of bound propagation. While semidefinite bound propagation demonstrates promising improvement in verification, the construction in Chiu et al. (2025) still relies on a ReLU-specific quadratic outer approximation and cannot be directly extended to general activations.

In this work, we develop semidefinite bound propagation for general activations using two complementary families of quadratic constraints: sector bounds and quadratic bounds. Our key insight is that the quadratic outer approximation for ReLU is already a special case: it combines the sector bound $y(y-z) \leq 0$ with two quadratic lower bounds of zero curvature, $y \geq 0$ and $y \geq z$. Sector bounds enclose the activation in two opposite sectors formed by two lines intersecting on its curve, capturing piecewise nonlinear behavior in activations. Quadratic bounds enclose the activation between upper and lower quadratic functions, capturing the curvature in the activation. Together, these families provide quadratic outer approximations beyond ReLU's piecewise linear structure.

Our semidefinite bound propagation method demonstrates substantial improvements over baseline methods for networks with general activations against both ℓ_∞ and ℓ_2 adversaries. Our method outperforms α-CROWN and GenBaB on all MNIST and CIFAR-10 benchmarks with general activations under a shared 300-second per-instance timeout.

Contributions. Our central contribution is a semidefinite relaxation that combines sector and quadratic bounds to derive optimizable linear bounds over ellipsoidal domains. These bounds can be integrated directly into the standard bound-propagation recursion, strengthening verification through richer quadratic relaxations without requiring dense SDP solves during propagation. We further use the same framework to recursively refine the ellipsoidal domain used to compute the linear bound. Our experiments demonstrate substantial improvements in verification performance over baseline methods against both ℓ_∞ and ℓ_2 adversaries.

Related work. Linear bound propagation methods propagate linear bounds through activation functions (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), enabling efficient and highly parallelizable computation. These bounds can be interpreted through the dual of a linear programming (LP) relaxation (Ehlers, 2017; Wong & Kolter, 2018; Salman et al., 2019). Their tightness can be further improved through 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 (Zhou et al., 2025). A complementary line of work strengthens LP relaxations using multi-neuron or groupwise constraints that capture dependencies among activations (Singh et al., 2019a; Müller et al., 2022; Ferrari et al., 2022; Ma et al., 2024). For general activations, CROWN (Zhang et al., 2018) and DeepPoly (Singh et al., 2019b) provide bound propagation, PRIMA (Müller et al., 2022) constructs groupwise relaxations, and GenBaB (Shi et al., 2025) supports branch-and-bound.

Semidefinite programming (SDP) relaxations provide strong verification bounds (Raghunathan et al., 2018; Fazlyab et al., 2019; Batten et al., 2021; Ma & Sojoudi, 2020; Fazlyab et al., 2020; Dvijotham et al., 2020). Some formulations already support general activations by encoding activation properties as quadratic constraints (Fazlyab et al., 2020). However, they remain computationally expensive for large networks despite substantial efforts to improve their scalability (Zhang, 2020; Dathathri et al., 2020; Batten et al., 2021; Newton & Papachristodoulou, 2021; Chiu & Zhang, 2023). SDP-CROWN (Chiu et al., 2025) addresses this challenge by incorporating semidefinite programming into bound propagation. However, (Chiu et al., 2025) relies on ReLU-specific quadratic constraints and remains limited to ReLU activations.

Notations We use $\mathbb{R}_{+}^{n\times m}$ and $\mathbb{R}_{++}^{n\times m}$ to denote the sets of $n \times m$ matrices with nonnegative and positive entries, respectively. We use x_i to denote the i-th element 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 $\mathrm{diag}(x)$ to denote a diagonal matrix whose diagonal entries equal to $x \in \mathbb{R}^n$. We use $\|\cdot\|_p$ to denote the vector ℓ_p norm. We use $[\underline{x}, \overline{x}] = \{x \mid \underline{x}_i \leq x_i \leq \overline{x}_i \text{ for all } i\}$ to denote an elementwise interval between $\underline{x}$ and $\overline{x}$.

2 VERIFICATION BY BOUND PROPAGATION

Given an input uncertainty set $\mathcal{X}_0$ and an output specification vector $c^{(N)}$, neural network verification seeks to certify that $f_\star \geq 0$, where

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

where σ is the activation function, and $W^{(\ell)}$ and $b^{(\ell)}$ are the weights and biases of layer ℓ, respectively. We consider the input set that is an axis-aligned ℓ_p ellipsoid with center $\hat{x}$ and semiaxes $d > 0$:

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

We focus on $p = 2$ and $p = \infty$ in the main paper. Our work can be extended to any $p \geq 1$. We note that $\mathcal{E}_\infty(\hat{x}, d)$ is equivalent to the elementwise interval set $[\underline{x}, \overline{x}]$ with $\underline{x} = \hat{x} - d$ and $\overline{x} = \hat{x} + d$.

Bound propagation (Xu et al., 2020) efficiently computes a valid lower bound on $f_\star$ by exploiting the network's layerwise structure. Let

$$\mathcal{X}_\ell = \{\sigma(z) \mid z \in \mathcal{Z}_\ell\}, \quad \mathcal{Z}_\ell = \{W^{(\ell)} x + b^{(\ell)} \mid x \in \mathcal{X}_{\ell-1}\} \quad \text{for } \ell = 1, \ldots, N$$

denote the exact reachable sets of the preactivations $z^{(\ell)}$ and postactivations $x^{(\ell)}$, and associate each layer with the value function

$$f_\ell(c) \coloneqq \min_{x \in \mathcal{X}_\ell} c^T x = \min_{z \in \mathcal{Z}_\ell} c^T \sigma(z) \quad \text{for } \ell = 1, \ldots, N$$

so that $f_\star = f_N(c^{(N)})$. Bound propagation repeatedly replaces a linear objective on the current layer with a linear objective on the preceding layer. Its central subproblem is to construct a linear lower bound on $c^\top \sigma(z)$.

Definition 1 (Linear bounding oracle)**.** *Given $c \in \mathbb{R}^n$ and a reachable set $\mathcal{Z}$, a linear bounding oracle with parameter $\lambda \in [\underline{\lambda}, \overline{\lambda}]$ with a domain $\widetilde{\mathcal{Z}} \supseteq \mathcal{Z}$ is defined such that*

$$\textit{for any } \lambda \in [\underline{\lambda}, \overline{\lambda}], \quad g(\lambda)^T z + h(\lambda) \leq c^T \sigma(z) \quad \textit{holds for all } z \in \widetilde{\mathcal{Z}} \supseteq \mathcal{Z}.$$

Given a lower bound $\varphi_{\ell-1}(\cdot) \leq f_{\ell-1}(\cdot)$, bound propagation constructs a lower bound $\varphi_\ell(\cdot) \leq f_\ell(\cdot)$ at the current layer through the linear bounding oracle:

$$f_\ell(c) \geq f_{\ell-1}(W^{(\ell)T} g(\lambda)) + g(\lambda)^T b^{(\ell)} + h(\lambda) \geq \varphi_{\ell-1}(W^{(\ell)T} g(\lambda)) + g(\lambda)^T b^{(\ell)} + h(\lambda).$$

The strongest lower bound within the chosen oracle is obtained by maximizing over $\lambda \in [\underline{\lambda}, \overline{\lambda}]$:

$$f_\ell(c) \geq \varphi_\ell(c) \coloneqq \max_{\lambda \in [\underline{\lambda}, \overline{\lambda}]} \left\{ \varphi_{\ell-1}\big((W^{(\ell)})^\top g(\lambda)\big) + g(\lambda)^\top b^{(\ell)} + h(\lambda) \right\}.$$

Crucially, the interval constraint $\lambda \in [\underline{\lambda}, \overline{\lambda}]$ permits inexpensive coordinatewise projection, making the maximization amenable to GPU-accelerated projected gradient methods.

This recurrence expresses $\varphi_\ell(\cdot) \leq f_\ell(\cdot)$ in terms of the lower bound $\varphi_{\ell-1}(\cdot) \leq f_{\ell-1}(\cdot)$ at the previous layer. Repeated backward substitution reaches the base case $\varphi_0(\cdot) = f_0(\cdot)$, which is available in closed form

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

where $1/p + 1/q = 1$. Applying the recurrence inductively yields $\varphi_N(c^{(N)}) \leq f_\star$.

2.1 CROWN: LINEAR BOUND PROPAGATION

Linear bound propagation relies on constructing a linear bounding oracle over an interval domain $\tilde{\mathcal{Z}} = [\underline{z}, \overline{z}]$. In particular, CROWN (Zhang et al., 2018) constructs a linear outer approximation for each $\sigma(z_i)$ over its interval

$$\underline{p}_i z_i + \underline{q}_i \leq \sigma(z_i) \leq \overline{p}_i z_i + \overline{q}_i \quad \text{for all } z_i \in [\underline{z}_i, \overline{z}_i].$$

Aggregating these bounds according to the sign of c_i gives the linear bounding oracle

$$g^T z + h \leq c^T \sigma(z) \quad \text{for all } z \in [\underline{z}, \overline{z}] \quad \text{where } \begin{cases} g_i = \underline{p}_i \max\{c_i, 0\} + \overline{p}_i \min\{c_i, 0\} \\ h = \sum_i \underline{q}_i \max\{c_i, 0\} + \overline{q}_i \min\{c_i, 0\}. \end{cases} \tag{1}$$

This construction applies to any activation for which valid linear bounds are available. For ReLU, its piecewise-linear structure provides a particularly simple and tight choice.

Linear bounding oracle for ReLU. Suppose $0 \in [\underline{z}, \overline{z}]$. Ehlers (2017) showed that the convex hull of $y = \mathrm{ReLU}(z)$ over $[\underline{z}, \overline{z}]$ is characterized by $\max\{z_i, 0\} \le y_i \le \frac{\overline{z}_i}{\overline{z}_i - \underline{z}_i}(z_i - \underline{z}_i)$ for all i. The linear bound propagation method α-CROWN (Xu et al., 2021) exploits this structure by constructing a parameterized linear outer approximation

$$\alpha_i \in [0,1], \quad \alpha_i z_i \le y_i \le \frac{\overline{z}_i}{\overline{z}_i - \underline{z}_i}(z_i - \underline{z}_i), \quad \text{for all } z_i \in [\underline{z}_i, \overline{z}_i].$$

Aggregating these bounds yields a linear bounding oracle with parameter $\alpha \in [0,1]$

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

The linear outer approximation for ReLU exactly describes the convex hull of an individual ReLU over its interval. For general activations, a pair of linear bounds need not describe the corresponding convex hull as tightly, leading to a looser bounding oracle (1). This motivates the semidefinite bound propagation developed in this work, which strengthens the linear bounding oracle using a quadratic outer approximation.

2.2 SDP-CROWN: Semidefinite Bound Propagation

Semidefinite bound propagation constructs the linear bounding oracle over a richer ellipsoid domain $\tilde{Z} = [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d)$. Suppose $0 \in \tilde{Z}$. Raghunathan et al. (2018) showed that $y_i = \mathrm{ReLU}(z_i)$ can be equivalently written as $y_i \ge z_i$, $y_i \ge 0$, $y_i(y_i - z_i) \le 0$ for all i due to its piecewise linear structure. The semidefinite bound propagation method SDP-CROWN (Chiu et al., 2025) exploits this structure by constructing the equivalent quadratic outer approximation over the ellipsoid domain

$$y_i \ge 0, \quad y_i \ge z_i, \quad y_i(y_i - z_i) \le 0 \quad \text{for all } z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d).$$

SDP-CROWN takes the quadratic outer approximation and derives the linear bounding oracle with parameters $g \in \mathbb{R}^n$, $\lambda \ge 0$

$$g^T z + h(g, \lambda) \le c^T \sigma(z) \text{ for all } z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d).$$

Unlike linear bound propagation, ReLU's quadratic outer approximation does not directly transfer to general activations; for instance, smooth activations, such as tanh or sigmoid, do not admit a similar piecewise-linear structure. Despite a substantial gain in verification performance, SDP-CROWN remains limited to ReLU networks.

3 From ReLU to General Activations

Prior work (Chiu et al., 2025) exploits the quadratic outer approximation of $y = \mathrm{ReLU}(z) \iff y \ge z,\ y \ge 0,\ y(y - z) \le 0$ to tighten the linear bounding oracle. Our key insight is that this quadratic outer approximation combines two classes of quadratic constraints: sector bounds and quadratic bounds:

$$y(y-z) \le 0, \quad y \ge 0, \quad y \ge z \iff \underbrace{0 \le \frac{y}{z} \le 1}_{\text{Sector bounds}}, \quad \underbrace{y \ge 0, \quad y \ge 0 \cdot z^2 + z}_{\text{Quadratic bounds}}.$$

Sector bounds confine a nonlinear function $\sigma(z)$ within two opposite sectors formed by two lines with slopes $\underline{a} \le \overline{a}$ that intersect at an anchor $\tilde{z}$, capturing the piecewise nonlinear structure at $\tilde{z}$. Quadratic bounds confine $\sigma(z)$ between two quadratic functions, capturing its curvature directly. The general forms of sector bounds and quadratic bounds are given as

$$\text{Sector bound: } \underline{a} \le \frac{\sigma(z) - \sigma(\tilde{z})}{z - \tilde{z}} \le \overline{a}. \quad \text{Quadratic bound: } \underline{c}z^2 + \underline{p}z + \underline{q} \le \sigma(z) \le \overline{c}z^2 + \overline{p}z + \overline{q}.$$

Together, sector bounds and quadratic bounds encode a powerful quadratic outer approximation. We illustrate this in Figure 1. We note that while $y \ge 0$ and $y \ge z$ are linear bounds, they can be captured by quadratic bounds with zero curvature. For ReLU, these zero-curvature bounds suffice when combined with the sector bound, which captures its piecewise linear structure.

In this section, we construct tractable linear bounding oracles for sector bounds and quadratic bounds. Each oracle converts the corresponding quadratic outer approximation into an optimizable linear lower bound over an ellipsoidal domain, which can be integrated directly into our proposed semidefinite bound propagation framework in Section 4.

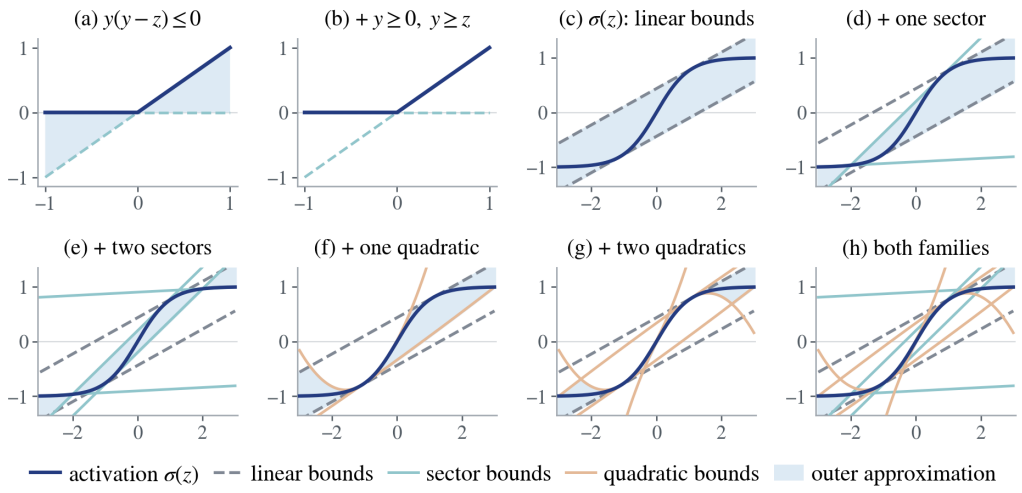

Figure 1: Quadratic outer approximations for ReLU and smooth activations $\sigma(z)$. (a) Sector bound for ReLU. (b) Adding the zero-curvature quadratic bounds $y \geq 0$ and $y \geq z$ makes ReLU's quadratic outer approximation exact. (c) Linear outer approximation of $\sigma(z)$. (d) Sector bound for $\sigma(z)$ anchored at $z = -2$. (e) Intersection of two sector bounds anchored at $z = 2$ and $z = -2$. (f) One pair of quadratic bounds for $\sigma(z)$. (g) Intersection of two pairs of quadratic bounds. (h) Intersecting sector and quadratic bounds further tightens the outer approximation of $\sigma(z)$. In (d) – (h), the constraints are combined with the linear bounds in (c), which can themselves be viewed as quadratic bounds with zero curvature.

3.1 Sector bounds

Sector bounds provide a simple way to construct quadratic outer approximations by controlling how $\sigma(z_i)$ changes within an interval $[\underline{z}_i, \overline{z}_i]$ relative to a chosen anchor. In particular, for each neuron z_i, we consider s sector bounds with anchors $\tilde{Z}_{ij} \in [\underline{z}_i, \overline{z}_i]$ and slopes $\underline{A}_{ij} \leq \overline{A}_{ij}$ for $j = 1, \ldots, s$

$$y_i = \sigma(z_i) \quad \Longrightarrow \quad \underline{A}_{ij} \leq \frac{y_i - \sigma(\tilde{Z}_{ij})}{z_i - \tilde{Z}_{ij}} \leq \overline{A}_{ij} \quad \text{for all } z_i \in [\underline{z}_i, \overline{z}_i]. \tag{2}$$

We note that in bound propagation, the interval $[\underline{z}_i, \overline{z}_i]$ is available, and therefore it is sufficient for the sector bounds to be valid within the interval. In practice, the interval can be computed by treating z_i as the output of bound propagation. We discuss how to compute it in Section 4. These sector bounds give a quadratic outer approximation for $y = \sigma(z)$ within an ellipsoidal domain.

$$\left[y_i - \sigma(\tilde{Z}_{ij}) - \underline{A}_{ij}(z_i - \tilde{Z}_{ij})\right]\left[y_i - \sigma(\tilde{Z}_{ij}) - \overline{A}_{ij}(z_i - \tilde{Z}_{ij})\right] \leq 0 \quad \text{for all } z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d)$$

In Lemma 1, we derive an optimizable linear bounding oracle using the quadratic outer approximation from sector bounds.

Lemma 1. *Given $c \in \mathbb{R}^n$, an interval $[\underline{z}, \overline{z}]$, and an ellipsoid $\mathcal{E}_2(\hat{z}, d)$. Suppose $\sigma(\cdot)$ is sector bounded (2), then the following holds*

$$g_s(\boldsymbol{\lambda})^T z + h_s(\boldsymbol{\lambda}) - \frac{1}{2}\eta \leq c^T \sigma(z) \quad \textit{for all } z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d)$$

for any $\eta \in \mathbb{R}_+$, $\boldsymbol{\lambda} = (\Lambda, G, \Omega) \in \mathbb{R}_+^{n\times s} \times \mathbb{R}^{n\times s} \times \mathbb{R}^{n\times s}$ that satisfy $c = \Omega \mathbf{1}$ and $\eta d_i^{-2} \geq m_{s,i}(\boldsymbol{\lambda})$ for all $i = 1, \ldots, n$, where $m_s(\boldsymbol{\lambda}) = \Lambda \mathbf{1}$ and

$$g_s(\boldsymbol{\lambda}) = (G + \Lambda \odot \tilde{Z})\mathbf{1} - \hat{z} \odot m_s(\boldsymbol{\lambda}), \quad \Phi_{ij} = \max\left\{|\Omega_{ij}\underline{A}_{ij} - G_{ij}|, |\Omega_{ij}\overline{A}_{ij} - G_{ij}|\right\},$$

$$h_s(\boldsymbol{\lambda}) = \sum_{i=1}^{n}\sum_{j=1}^{s}\left[\Omega_{ij}\sigma(\tilde{Z}_{ij}) - G_{ij}\tilde{Z}_{ij} - \frac{\Phi_{ij}^2 + (\Lambda_{ij}\tilde{Z}_{ij})^2}{2\Lambda_{ij}}\right] + \frac{1}{2}\hat{z}^T \operatorname{diag}(m_s(\boldsymbol{\lambda}))\hat{z}.$$

The linear bounding oracle in Lemma 1 can be efficiently optimized by eliminating the first column of Ω using $\Omega e_1 = c - \sum_{j=2}^{s} \Omega e_j$ and eliminating η using $\eta = \max\left\{0, \max_{1\le i\le n} d_i^2 m_{s,i}(\boldsymbol{\lambda})\right\}$. The remaining parameters become either unconstrained or nonnegative.

3.2 Quadratic bounds

Quadratic bounds provide a direct extension of the linear outer approximation used in bound propagation. Their additional quadratic terms allow the bounding functions to follow activation curvature more closely and can reduce approximation error. We consider q pairs of quadratic bounds with coefficients $\underline{C}_{ij}, \underline{P}_{ij}, \underline{Q}_{ij}, \overline{C}_{ij}, \overline{P}_{ij}, \overline{Q}_{ij}$ for $j = 1, \dots, q$

$$y_i = \sigma(z_i) \implies \frac{1}{2}\underline{C}_{ij} z_i^2 + \underline{P}_{ij} z_i + \underline{Q}_{ij} \le y_i \le \frac{1}{2}\overline{C}_{ij} z_i^2 + \overline{P}_{ij} z_i + \overline{Q}_{ij} \quad \text{for all } z_i \in [\underline{z}_i, \overline{z}_i]. \tag{3}$$

We note that the linear outer approximation used in CROWN is covered under (3), so our method can automatically incorporate it. The quadratic bounds provide quadratic outer approximations complementary to the sector bounds. In Lemma 2, we derive a linear bounding oracle using the quadratic outer approximation from quadratic bounds.

Lemma 2. *Given $c \in \mathbb{R}^n$, an interval $[\underline{z}, \overline{z}]$, and an ellipsoid $\mathcal{E}_2(\hat{z}, d)$. Suppose $\sigma(\cdot)$ is quadratically bounded (3), then the following holds*

$$g_q(\boldsymbol{\tau})^T z + h_q(\boldsymbol{\tau}) - \frac{1}{2}\eta \le c^T \sigma(z) \quad \textit{for all } z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d)$$

for any $\eta \in \mathbb{R}_+$, $\boldsymbol{\tau} = (\underline{\Gamma}, \overline{\Gamma}) \in \mathbb{R}_+^{n\times q} \times \mathbb{R}_+^{n\times q}$ that satisfy $c = (\underline{\Gamma} - \overline{\Gamma})\mathbf{1}$ and $\eta d_i^{-2} \ge m_{q,i}(\boldsymbol{\tau})$ for all $i = 1, \dots, n$, where $m_q(\boldsymbol{\tau}) = (\overline{C} \odot \overline{\Gamma} - \underline{C} \odot \underline{\Gamma})\mathbf{1}$ and

$$g_q(\boldsymbol{\tau}) = (\underline{P} \odot \underline{\Gamma} - \overline{P} \odot \overline{\Gamma})\mathbf{1} - \hat{z} \odot m_q(\boldsymbol{\tau}), \quad h_q(\boldsymbol{\tau}) = \left\langle \underline{Q}, \underline{\Gamma} \right\rangle - \left\langle \overline{Q}, \overline{\Gamma} \right\rangle + \frac{1}{2}\hat{z}^T \operatorname{diag}(m_q(\boldsymbol{\tau}))\hat{z}.$$

The linear lower bounding oracle in Lemma 2 can also be efficiently optimized by eliminating η using $\eta = \max\left\{0, \max_{1\le i\le n} d_i^2 m_{q,i}(\boldsymbol{\tau})\right\}$ and eliminating the first columns, $\underline{\Gamma} e_1$ and $\overline{\Gamma} e_1$, via

$$\underline{\Gamma} e_1 = \max\{0, \tilde{c}\} + \gamma, \quad \overline{\Gamma} e_1 = \max\{0, -\tilde{c}\} + \gamma, \quad \gamma \in \mathbb{R}_+^n \tag{4}$$

with $\tilde{c} = c - \sum_{j=2}^{q} (\underline{\Gamma} - \overline{\Gamma}) e_j$. The remaining parameters become nonnegative.

Due to space constraints, we defer the proofs of Lemma 1 and Lemma 2 to Appendix A, and the details of constructing sector bounds (2) and quadratic bounds (3) to Appendix B.

Comparison with Quadratic Bounds in CROWN (Zhang et al., 2018) Although CROWN-Quad (Zhang et al., 2018) also considers quadratic activation bounds, the curvature signs must be chosen to ensure convexity of the resulting quadratic optimization problem. Specifically, $\overline{C}_{ij} \le 0$ and $\underline{C}_{ij} \ge 0$, and such a construction is extremely limited. For instance, the quadratic upper bound demonstrated in Figure 1 (f) is prohibited in CROWN-Quad. In addition, CROWN-Quad can only choose one pair of quadratic bounds per neuron, while our method combines multiple pairs into a linear bounding oracle that can propagate through successive layers. Furthermore, our semidefinite bound propagation framework converts quadratic bounds into linear bounding oracles in Lemma 2, preserving the linear backward recursion through successive layers. CROWN-Quad instead retains a quadratic objective and solves a quadratic optimization problem at the preceding layer, which cannot directly handle more than two layers, as quadratics are not closed under composition.

4 Semidefinite Bound Propagation for General Activations

In this section, we combine the bounding oracles from sector and quadratic bounds in Section 3 into a single bounding oracle. We then describe our semidefinite bound propagation framework.

4.1 Combining Linear Bounding Oracles from Sector and Quadratic Bounds

A high-level intuition of how the bounding oracles in Lemmas 1 and 2 are constructed is that they are constructed through a semidefinite programming relaxation (SDP) of their respective quadratic outer approximations. In particular, their dual SDP can be expressed as a quadratic lower bound over the ellipsoid domain

$$\eta \geq 0, \quad \eta d_i^{-2} \geq m_i, \quad g^\top z + h - \frac{1}{2}\sum_{i=1}^{n} m_i(z_i - \hat{z}_i)^2 \leq c^T\sigma(z) \quad \text{for all } z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d) \tag{5}$$

We defer the derivation details of (5) to Appendix A. Here, g, h, m are equal to $g_s(\boldsymbol{\lambda}), h_s(\boldsymbol{\lambda}), m_s(\boldsymbol{\lambda})$ for sector bounds (Lemma 1) and $g_q(\boldsymbol{\tau}), h_q(\boldsymbol{\tau}), m_q(\boldsymbol{\tau})$ for quadratic bounds (Lemma 2). The ellipsoid domain $\mathcal{E}_2(\hat{z}, d)$ then controls the quadratic term in (5). In particular, since $\eta \geq 0$ and $m_i \leq \eta d_i^{-2}$ for every i, the quadratic term can be upper bounded as:

$$\sum_{i=1}^{n} m_i(z_i - \hat{z}_i)^2 \leq \eta \sum_{i=1}^{n} \frac{(z_i - \hat{z}_i)^2}{d_i^2} \leq \eta \qquad \text{for } z \in \mathcal{E}_2(\hat{z}, d).$$

Substituting this inequality into the quadratic representation gives the linear bounding oracle $g^T z + h - \frac{1}{2}\eta$ in both lemmas. Since both lemmas yield the quadratic lower bound in the same form as (5), combining the two lemmas gives the quadratic lower bound (5) with

$$g = g_s(\boldsymbol{\lambda}) + g_q(\boldsymbol{\tau}), \qquad h = h_s(\boldsymbol{\lambda}) + h_q(\boldsymbol{\tau}), \qquad m = m_s(\boldsymbol{\lambda}) + m_q(\boldsymbol{\tau}).$$

In Theorem 1, we derive the linear bounding oracle using the quadratic outer approximations for both sector bounds (2) and quadratic bounds (3). The bounding oracle in Theorem 1 is additionally extended to an even more general ellipsoidal domain given by an intersection of K ellipsoids $\mathcal{E}_2(\hat{z}_k, d_k)$ for $k = 1, \ldots, K$.

Theorem 1. *Given $c \in \mathbb{R}^n$, an interval $[\underline{z}, \overline{z}]$, and ellipsoids $\mathcal{E}_2(\hat{z}_k, d_k)$ for $k = 1, \ldots, K$. Suppose $\sigma(\cdot)$ is sector bounded (2) and quadratically bounded (3), then the following holds*

$$g(\eta, \boldsymbol{\lambda}, \boldsymbol{\tau})^T z + h(\eta, \boldsymbol{\lambda}, \boldsymbol{\tau}) \leq c^T\sigma(z) \quad \textit{for all } z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}_1, d_1) \cap \ldots \cap \mathcal{E}_2(\hat{z}_K, d_K)$$

for any $\eta \in \mathbb{R}_+^K$, $\boldsymbol{\lambda} = (\Lambda, G, \Omega) \in \mathbb{R}_+^{n\times s} \times \mathbb{R}^{n\times s} \times \mathbb{R}^{n\times s}$, $\boldsymbol{\tau} = (\underline{\Gamma}, \overline{\Gamma}) \in \mathbb{R}_+^{n\times q} \times \mathbb{R}_+^{n\times q}$ that satisfy $c = \Omega\mathbf{1} + (\underline{\Gamma} - \overline{\Gamma})\mathbf{1}$ and $\sum_{k=1}^K \eta_k d_{k,i}^{-2} \geq m_{s,i}(\boldsymbol{\lambda}) + m_{q,i}(\boldsymbol{\tau})$ for all $i = 1, \ldots, n$, where

$$g(\eta, \boldsymbol{\lambda}, \boldsymbol{\tau}) = g_s(\boldsymbol{\lambda}) + g_q(\boldsymbol{\tau}), \quad \hat{z}_i(\eta) = \frac{\sum_{k=1}^K \hat{z}_{k,i}\eta_k d_{k,i}^{-2}}{\sum_{k=1}^K \eta_k d_{k,i}^{-2}}$$

$$h(\eta, \boldsymbol{\lambda}, \boldsymbol{\tau}) = h_s(\boldsymbol{\lambda}) + h_q(\boldsymbol{\tau}) - \frac{1}{2}\sum_{k=1}^{K} \eta_k \left[1 - \| \operatorname{diag}(d_k)^{-1} (\hat{z}_k - \hat{z}(\eta)) \|_2^2\right].$$

Here, $g_s(\boldsymbol{\lambda}), h_s(\boldsymbol{\lambda}), g_q(\boldsymbol{\tau}), h_q(\boldsymbol{\tau})$ are evaluated at the center $\hat{z} = \hat{z}(\eta)$.

The linear lower bounding oracle in Theorem 1 can be efficiently optimized by reparameterizing the first element of η via

$$\eta_1 = \max\left\{0, \max_{1\leq i\leq n} d_{1,i}^2 \left(m_i - \sum_{k=2}^{K} \frac{\eta_k}{d_{k,i}^2}\right)\right\} + t, \quad t \geq 0$$

with $m_i = m_{s,i}(\boldsymbol{\lambda}) + m_{q,i}(\boldsymbol{\tau})$, and eliminating the first columns, $\underline{\Gamma}e_1$ and $\overline{\Gamma}e_1$, using (4) with $\tilde{c} = c - \Omega\mathbf{1} - \sum_{j=2}^q (\underline{\Gamma} - \overline{\Gamma})e_j$. The remaining parameters become either unconstrained or nonnegative.

Comparison with SDP-CROWN (Chiu et al., 2025) for ReLU activations SDP-CROWN (Chiu et al., 2025) is a special case of Theorem 1. In particular, SDP-CROWN (Chiu et al., 2025) is recovered by picking the sector $0 \leq \frac{y_i}{z_i} \leq 1$ in (2) and the quadratic bounds $y_i \geq 0$, $y_i \geq z_i$ in (3). We provide the details of this connection in Appendix D.

4.2 SEMIDEFINITE BOUND PROPAGATION

The linear bounding oracle in Theorem 1 can be seamlessly integrated into the bound propagation framework that efficiently computes a lower bound $\varphi_\ell(c) \le f_\ell(c)$. We start by describing how to construct the domain $\tilde{\mathcal{Z}}_\ell \supseteq \mathcal{Z}_\ell$ of our linear bounding oracle, which consists of an interval $[\underline{z}, \overline{z}]$ and ellipsoids $\mathcal{E}_2(\hat{z}_k, d_k)$ for $k = 1, \dots, K$.

Interval Domain. We construct an interval $[\underline{z}, \overline{z}] \supseteq \mathcal{Z}_\ell$ by bounding each preactivation coordinate using $\varphi_{\ell-1}$.

$$[\underline{z}, \overline{z}] \supseteq \mathcal{Z}_\ell \quad \text{where} \quad \underline{z}_i = \varphi_{\ell-1}(W^{(\ell)T} e_i) + b_i^{(\ell)}, \quad \overline{z}_i = -\varphi_{\ell-1}(-W^{(\ell)T} e_i) + b_i^{(\ell)}.$$

This construction of $[\underline{z}, \overline{z}]$ is a standard step in bound propagation (Zhang et al., 2018).

Ellipsoid Domain. We recursively construct ellipsoids $\mathcal{E}_2(\hat{z}_k, d_k) \supseteq \mathcal{Z}_\ell$ for $k = 1, \dots, K$ using semidefinite bound propagation. For a center $\hat{z}$ and semiaxis ratios a, we associate a value function

$$h_\ell(\hat{z}, a) \coloneqq \max_{z \in \mathcal{Z}_\ell} \sum_i \frac{(z_i - \hat{z}_i)^2}{a_i^2}$$

This value function corresponds to a verification problem obtained by subtracting $\hat{z}$ from $z^{(\ell)}$, applying the square activation $\sigma(z_i) = z_i^2$, and using the output specification $c_i = a_i^{-2}$. We note that the square activation is represented exactly by the quadratic bounds (3). Picking any $\hat{z}_k$ and a_k, we apply semidefinite bound propagation to provide an upper bound $\kappa_\ell(\hat{z}, a) \ge h_\ell(\hat{z}, a)$, which determines the ellipsoid semiaxes

$$\mathcal{E}_2(\hat{z}_k, d_k) \supseteq \mathcal{Z}_\ell \quad \text{where} \quad (\hat{z}_k, d_k) \quad \text{satisfy} \quad d_k = a_k \sqrt{\kappa_\ell(\hat{z}_k, a_k)}.$$

The ellipsoids can also be constructed using the propagation rules in Appendix C, which we adopted during initialization.

Semidefinite Bound Propagation. We construct our linear bounding oracle using Theorem 1 with the domain $[\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}_1, d_1) \cap \ldots \cap \mathcal{E}_2(\hat{z}_K, d_K) \supseteq \mathcal{Z}_\ell$ and propagate the bounding oracle backward

$$\begin{aligned} f_\ell(c) &= \min_{z \in \mathcal{Z}_\ell} c^T \sigma(z) \ge \max_{\eta, \boldsymbol{\lambda}, \boldsymbol{\tau}} \min_{z \in \mathcal{Z}_\ell} g(\eta, \boldsymbol{\lambda}, \boldsymbol{\tau})^T z + h(\eta, \boldsymbol{\lambda}, \boldsymbol{\tau}) \\ &= \max_{\eta, \boldsymbol{\lambda}, \boldsymbol{\tau}} f_{\ell-1}(W^{(\ell)T} g(\eta, \boldsymbol{\lambda}, \boldsymbol{\tau})) + b^{(\ell)T} g(\eta, \boldsymbol{\lambda}, \boldsymbol{\tau}) + h(\eta, \boldsymbol{\lambda}, \boldsymbol{\tau}) \\ &\ge \max_{\eta, \boldsymbol{\lambda}, \boldsymbol{\tau}} \varphi_{\ell-1}(W^{(\ell)T} g(\eta, \boldsymbol{\lambda}, \boldsymbol{\tau})) + b^{(\ell)T} g(\eta, \boldsymbol{\lambda}, \boldsymbol{\tau}) + h(\eta, \boldsymbol{\lambda}, \boldsymbol{\tau}) \coloneqq \varphi_\ell(c). \end{aligned}$$

Since the parameters $\eta, \boldsymbol{\lambda}, \boldsymbol{\tau}$ are either unconstrained or nonnegative after reparameterization, our semidefinite bound propagation framework permits inexpensive coordinatewise projection, making the maximization amenable to GPU-accelerated projected gradient methods.

5 EXPERIMENTS

Baselines. We compare against the two most widely used verifiers for networks with general activations: α-CROWN (Xu et al., 2021), which optimizes linear bound propagation, and GenBaB (Shi et al., 2025), which further adds branch-and-bound for general nonlinearities.

Models. We use the networks from GenBaB (Shi et al., 2025). As our method does not yet support general computational graphs, we evaluate all their feedforward networks, which use sigmoid, tanh, sine, and GeLU activations on MNIST and CIFAR-10. Although our method handles any ℓ_p perturbation with $p \ge 1$, we focus on the two most common threat models, ℓ_2 and ℓ_∞ (Madry et al., 2018; Carlini & Wagner, 2017; Wong & Kolter, 2018; Zhang et al., 2018; Chiu et al., 2025). For ℓ_∞, we use the original instances of GenBaB; for ℓ_2, we use the same images with a per-network radius ρ chosen so that most instances are robust but nontrivial to verify. Each benchmark has 100 instances, and each instance has a 300-second timeout per the original GenBaB settings.

Table 1: Verified accuracy (%) under ℓ_2 perturbations of radius ρ. "Upper Bound" is the percentage of instances on which PGD (Madry et al., 2018) finds no adversarial example. Best in bold.

Activation	Model	Radius ρ	Upper Bound (%)	Verified (%)			
				Ours	Ours w/o BaB	α-CROWN	GenBaB
MNIST							
Sigmoid	6×100	0.2	89	**76**	68	62	68
	6×200	0.2	98	**80**	75	64	68
	9×100	0.225	89	**57**	53	41	44
	ConvSmall	0.1	98	**57**	39	6	12
Tanh	6×100	0.075	97	**84**	78	77	81
	6×200	0.05	99	**58**	53	49	56
	9×100	0.1	98	**63**	58	54	59
	ConvSmall	0.05	98	**19**	13	0	3
CIFAR-10							
Sigmoid	4×100	0.2	74	**16**	3	0	2
	4×500	0.175	98	**60**	43	5	12
	6×100	0.175	88	**41**	27	3	28
	6×200	0.175	91	**53**	45	13	31
Tanh	4×100	0.15	99	**64**	50	20	47
	6×100	0.1	100	**81**	71	48	64
Sine	4×100	0.2	68	**56**	44	0	16
	4×200	0.2	78	**61**	54	0	0
	4×500	0.2	85	**82**	77	0	0
GeLU	4×100	0.3	60	**15**	6	3	10
	4×200	0.2	95	**46**	30	15	30
	4×500	0.2	95	**31**	23	10	22

Table 2: Verified accuracy (%) under ℓ_∞ perturbations with $\epsilon = 1/255$ on CIFAR-10. Best in bold.

Method	Sigmoid Networks				Tanh Networks		Sine Networks			GeLU Networks		
	4×100	4×500	6×100	6×200	4×100	6×100	4×100	4×200	4×500	4×100	4×200	4×500
α-CROWN	28	16	43	39	25	6	4	2	4	49	40	33
GenBaB	61	25	64	50	54	19	59	35	23	81	67	40
Ours w/o BaB	51	53	53	55	49	30	**81**	82	85	72	62	42
Ours	**73**	**67**	**75**	**69**	**68**	**37**	**81**	**84**	**87**	**89**	**76**	**49**

ℓ_2 Verification. As shown in Table 1, our method achieves the highest verified accuracy on all 20 networks, verifying 2 to 45 more instances than GenBaB on MNIST and 5 to 82 more on CIFAR-10, where it more than doubles the verified instances of GenBaB on 5 of the 12 networks. The gains are most pronounced where linear relaxations are loose, especially on the sine networks: on the 4×200 and 4×500 networks, neither baseline verifies any instance, whereas our method verifies 61 and 82, close to the PGD upper bounds of 78 and 85. Our method is equally effective on the convolutional ConvSmall networks, verifying 57 and 19 instances for sigmoid and tanh versus 12 and 3 for GenBaB, and it remains the strongest method on the deeper 9-layer MLPs. The benefit of tighter bounds is evident. Our method verifies more instances than α-CROWN on every network and more than GenBaB, which uses branch-and-bound, on 13 of the 20 networks.

ℓ_∞ Verification. Prior semidefinite bound propagation methods (Chiu et al., 2025) have demonstrated their benefits only for ℓ_2 perturbations, but our method applies equally to ℓ_∞. Table 2 reports results on all the CIFAR-10 feedforward benchmarks from Shi et al. (2025), using their original instances. Similarly, we compare with GenBaB with the default branching setting. Our method again achieves the highest verified accuracy on all 12 networks, verifying 8 to 64 more instances than GenBaB, with the largest gains also on the sine networks (81, 84, and 87 versus 59, 35, and 23). These gains are consistent with the ability of quadratic bounds to better capture the curvature of sine activations, allowing substantially tighter relaxations on intervals where linear bounds are loose. Similarly, without branch-and-bound, our method verifies more instances than α-CROWN on every network and more than GenBaB on 7 of the 12 networks. Adding branch-and-bound verifies 7 to 22 more instances on the sigmoid and tanh networks, showing that our bounds combine effectively with branching for general activations.

AI USE STATEMENT

We used Codex to help implement our method and to help polish the writing of this paper. All mathematical derivations, research ideas, and experimental choices are our own. We checked every result and claim in the paper and take full responsibility for its content.

REFERENCES

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

Christopher Brix, Mark Niklas Müller, Stanley Bak, Taylor T Johnson, and Changliu Liu. First three years of the international verification of neural networks competition (vnn-comp). *International Journal on Software Tools for Technology Transfer*, 25(3):329–339, 2023.

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

Hong-Ming Chiu and Richard Y Zhang. Tight certification of adversarially trained neural networks via nonconvex low-rank semidefinite relaxations. In *International Conference on Machine Learning*, pp. 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*, pp. 10449–10468. PMLR, 2025.

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*, 2020.

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, pp. 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*, pp. 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*, pp. 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*, pp. 463–480. Springer, 2021.

Sepp Hochreiter and J"urgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Haoyu Li, Xiangru Zhong, Bin Hu, and Huan Zhang. Two-stage learning of stabilizing neural controllers via zubov sampling and iterative domain expansion. *Advances in Neural Information Processing Systems*, 38:73980–74009, 2026.

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*, pp. 715–727, 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

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.

Matthew Newton and Antonis Papachristodoulou. Exploiting sparsity for neural network verification. In *Learning for dynamics and control*, pp. 715–727. PMLR, 2021.

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*, pp. 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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 Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pp. 5286–5295, 2018.

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

Kaidi Xu, Huan Zhang, Shiqi Wang, Yihan Wang, Suman Jana, Xue Lin, and Cho-Jui Hsieh. Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. In *International Conference on Learning 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.

A Proof of Main Theorem

In this section, we first establish the proofs for Lemma 1 and Lemma 2. The proof of Theorem 1 follows directly from both lemmas.

A.1 Proof of Lemma 1

We start by describing our SDP relaxation for the sector bounded activations

$$y_i = \sigma(z_i) \quad \Longrightarrow \quad \underline{a}_i \le \frac{y_i - \sigma(\tilde{z}_i)}{z_i - \tilde{z}_i} \le \overline{a}_i.$$

Absolute-split SDP Relaxation for Sector Bounded Activations We propose to derive a SDP relaxation for a sector bounded activation $y = \sigma(z)$ from the absolute-split point of view. Our key observation is that the quadratic outer approximation for sector bound

$$[y_i - \sigma(\tilde{z}_i) - \underline{a}_i(z_i - \tilde{z}_i)]\,[y_i - \sigma(\tilde{z}_i) - \overline{a}_i(z_i - \tilde{z}_i)] \le 0 \quad \text{for all } z_i \in [\underline{z}_i, \overline{z}_i]$$

can be equivalently written as

$$\frac{\overline{a}_i + \underline{a}_i}{2}(z_i - \tilde{z}_i) - \frac{\overline{a}_i - \underline{a}_i}{2}|z_i - \tilde{z}_i| \le y_i - \sigma(\tilde{z}_i) \le \frac{\overline{a}_i + \underline{a}_i}{2}(z_i - \tilde{z}_i) + \frac{\overline{a}_i - \underline{a}_i}{2}|z_i - \tilde{z}_i|.$$

Replacing $r = |z - \tilde{z}|$ with

$$|z - \tilde{z}| \le r, \quad r \odot r \le (z - \tilde{z}) \odot (z - \tilde{z}),$$

and relaxing $z_i^2 \le Z_i$ and $r_i^2 \le R_i$ yield the following SDP relaxation

$$\begin{aligned} \frac{\overline{a} + \underline{a}}{2} \odot (z - \tilde{z}) - \frac{\overline{a} - \underline{a}}{2} \odot r \le y - \sigma(\tilde{z}) \le \frac{\overline{a} + \underline{a}}{2} \odot (z - \tilde{z}) + \frac{\overline{a} - \underline{a}}{2} \odot r, \\ |z - \tilde{z}| \le r, \quad R \le Z - 2z \odot \tilde{z} + \tilde{z} \odot \tilde{z}, , \\ \begin{bmatrix} 1 & z_i \\ z_i & Z_i \end{bmatrix} \succeq 0, \quad \begin{bmatrix} 1 & r_i \\ r_i & R_i \end{bmatrix} \succeq 0 \quad \text{for all } i. \end{aligned} \tag{6}$$

Proof of Lemma 1 Lemma 1 can be proved by deriving the SDP relaxation of the following

$$\min_z \; c^T \sigma(z) \quad \text{s.t.} \quad (2), \quad z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d).$$

We start by applying a variable substitution $z = x + \hat{z}$, and let $\hat{B}_{ij} = \hat{z}_i - \tilde{Z}_{ij}$. Applying the absolute split SDP relaxation (6) to yield

$$\begin{aligned} \min_{\substack{y,x,X \in \mathbb{R}^n \\ Z,r,R \in \mathbb{R}^{n \times s}}} \; c^T y \text{ s.t.} \quad & \tfrac{\overline{A}_{ij} + \underline{A}_{ij}}{2} Z_{ij} - \tfrac{\overline{A}_{ij} - \underline{A}_{ij}}{2} r_{ij} \le y_i - \sigma(\tilde{Z}_{ij}) \le \tfrac{\overline{A}_{ij} + \underline{A}_{ij}}{2} Z_{ij} + \tfrac{\overline{A}_{ij} - \underline{A}_{ij}}{2} r_{ij}, \\ & |Z_{ij}| \le r_{ij}, \quad R_{ij} \le X_i + 2Z_{ij}\hat{B}_{ij} - \hat{B}_{ij}^2, \quad Z_{ij} = x_i + \hat{B}_{ij}, \\ & \textstyle\sum_{i=1}^n \frac{X_i}{d_i^2} \le 1, \quad \begin{bmatrix} 1 & x_i \\ x_i & X_i \end{bmatrix} \succeq 0, \quad \begin{bmatrix} 1 & r_{ij} \\ r_{ij} & R_{ij} \end{bmatrix} \succeq 0 \end{aligned} \tag{7}$$

for $i = 1, \ldots, n$, $j = 1, \ldots, s$. The core insight to prove Lemma 1 is to fix any $z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d)$, and then take the dual to lower bound (7). We defer the detail of deriving the dual problem into Appendix E.

Let $\eta \in \mathbb{R}_+$, $\boldsymbol{\lambda} = (\Lambda, G, \Omega) \in \times \mathbb{R}_+^{n \times s} \times \mathbb{R}^{n \times s} \times \mathbb{R}^{n \times s}$ be the dual variables, the Lagrangian dual of (7) is given by

$$\begin{aligned} \max_{\eta, \boldsymbol{\lambda}} \; & g_s(\boldsymbol{\lambda})^T z + h_s(\boldsymbol{\lambda}) + \frac{1}{2}\left[(z - \hat{z})^T M (z - \hat{z}) - \eta\right] \\ \text{s.t. } & c = \Omega \mathbf{1}, \quad M = \eta \,\mathrm{diag}(d)^{-2} - \mathrm{diag}(\Lambda \mathbf{1}) \succeq 0. \end{aligned} \tag{8}$$

Set $m_s(\boldsymbol{\lambda}) = \Lambda \mathbf{1}$ and choose η such that $\eta d_i^{-2} \ge m_{s,i}(\boldsymbol{\lambda})$ for all i, then $M \succeq 0$. For every $z \in \mathcal{E}_2(\hat{z}, d)$, the desired result follows from $(z - \hat{z})^T M (z - \hat{z}) - \eta \ge -\eta$.

A.2 Proof of Lemma 2

The proof of Lemma 2 follows similar structure as in Lemma 2 We start by describing the SDP relaxation for the quadratic bounded activations

$$y_i = \sigma(z_i) \quad \Longrightarrow \quad \frac{1}{2}\underline{c}_i z_i^2 + \underline{p}_i z_i + \underline{q}_i \le y_i \le \frac{1}{2}\overline{c}_i z_i^2 + \overline{p}_i z_i + \overline{q}_i.$$

SDP Relaxation for Quadratic Bounded Activations The SDP relaxation for a quadratically bounded activation $y = \sigma(z)$ can be derived by relaxing $z_i^2 \le Z_i$

$$\frac{1}{2}\underline{c}_i Z_i + \underline{p}_i z_i + \underline{q}_i \le y_i \le \frac{1}{2}\overline{c}_i Z_i + \overline{p}_i z_i + \overline{q}_i, \quad \begin{bmatrix} 1 & z_i \\ z_i & Z_i \end{bmatrix} \succeq 0, \quad \text{for all } i. \tag{9}$$

Proof of Lemma 2 Lemma 2 can be proved by deriving the SDP relaxation of the following

$$\min_z \; c^T\sigma(z) \quad \text{s.t.} \quad (3), \quad z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d).$$

We start by applying a variable substitution $z = x + \hat{z}$, and then applying SDP relaxation (9) to yield

$$\min_{\substack{y,x,X\in\mathbb{R}^n \\ z,Z\in\mathbb{R}^n}} c^T y \;\; \text{s.t.} \;\; \begin{array}{c} \frac{1}{2}\underline{C}_{ij} Z_i + \underline{P}_{ij} z_i + \underline{Q}_{ij} \le y_i \le \frac{1}{2}\overline{C}_{ij} Z_i + \overline{P}_{ij} z_i + \overline{Q}_{ij}, \quad z = x + \hat{z} \\ Z_i = X_i + 2z_i\hat{z}_i - \hat{z}_i^2, \quad \sum_{i=1}^n \frac{X_i}{d_i^2} \le 1, \quad \begin{bmatrix} 1 & x_i \\ x_i & X_i \end{bmatrix} \succeq 0 \end{array} \tag{10}$$

for $i = 1, \ldots, n$, $j = 1, \ldots, q$. Similar to the proof in Lemma 2, we fix any $z \in [\underline{z}, \overline{z}] \cap \mathcal{E}_2(\hat{z}, d)$, and then take the dual to lower bound (10). We defer the derivation for the dual problem to Appendix E. Let $\eta \in \mathbb{R}_+$, $\boldsymbol{\tau} = (\underline{\Gamma}, \overline{\Gamma}) \in \mathbb{R}_+^{n\times q} \times \mathbb{R}_+^{n\times q}$ be the dual variables, the Lagrangian dual is given by

$$\begin{array}{l} \max\limits_{\eta,\boldsymbol{\tau}} \; g_q(\boldsymbol{\tau})^T z + h_q(\boldsymbol{\tau}) + \dfrac{1}{2}\left[(z-\hat{z})^T M (z-\hat{z}) - \eta\right] \\ \text{s.t.} \;\; c = (\underline{\Gamma} - \overline{\Gamma})\mathbf{1}, \quad M = \eta \,\mathrm{diag}(d)^{-2} - \mathrm{diag}((\overline{C} \odot \overline{\Gamma} - \underline{C} \odot \underline{\Gamma})\mathbf{1}) \succeq 0. \end{array} \tag{11}$$

Set $m_q(\boldsymbol{\tau}) = (\overline{C} \odot \overline{\Gamma} - \underline{C} \odot \underline{\Gamma})\mathbf{1}$ and choose η such that $\eta d_i^{-2} \ge m_{q,i}(\boldsymbol{\tau})$ for all i, then $M \succeq 0$. For every $z \in \mathcal{E}_2(\hat{z}, d)$, the desired result follows from $x^T M x - \eta \ge -\eta$.

A.3 Proof of Theorem 1

Theorem 1 additionally consider the intersection of K ellipsoid

$$\mathcal{E}_2(\hat{z}_1, d_1) \cap \ldots \cap \mathcal{E}_2(\hat{z}_K, d_K).$$

We first study how it would change the dual in (8) and (11).

Dual of Ellipsoids Domain. Let $\hat{z}$ be any reference center and let $\Delta_k = \hat{z} - \hat{z}_k$ and $x = z - \hat{z}$, we can rewrite each ellipsoid as $\mathcal{E}_2(\hat{z} - \Delta_k, d_k)$. Now, consider the following feasibility problem of $\mathcal{E}_2(\hat{z} - \Delta_1, d_1) \cap \ldots \cap \mathcal{E}_2(\hat{z} - \Delta_K, d_K)$

$$\min_{x,X\in\mathbb{R}^n} 0 \quad \text{s.t.} \quad \sum_{i=1}^n \frac{X_i + 2\Delta_{k,i}(z_i - \hat{z}_i) + \Delta_{k,i}^2}{d_{k,i}^2} \le 1, \quad \begin{bmatrix} 1 & x_i \\ x_i & X_i \end{bmatrix} \succeq 0 \tag{12}$$

Let $\eta_k \geq 0$ denote the dual variables associated with the k-th ellipsoids, and let $\begin{bmatrix} \nu_i & \tilde{x}_i \\ \tilde{x}_i & \tilde{X}_i \end{bmatrix} \succeq 0$ denote the dual variables for the PSD blocks $\begin{bmatrix} 1 & x_i \\ x_i & X_i \end{bmatrix} \succeq 0$. The Lagrangian is given by

$$\begin{aligned}
&\sum_{k=1}^{K} \eta_k \left[\sum_{i=1}^{n} \frac{X_i + 2\Delta_{k,i} x_i + \Delta_{k,i}^2}{d_{k,i}^2} - 1 \right] - \sum_{i=1}^{n} \left\langle \begin{bmatrix} \nu_i & \tilde{x}_i \\ \tilde{x}_i & \tilde{X}_i \end{bmatrix}, \begin{bmatrix} 1 & x_i \\ x_i & X_i \end{bmatrix} \right\rangle \\
&= -\eta^T \mathbf{1} + \left\langle \sum_{k=1}^{K} \eta_k \operatorname{diag}(d_k)^{-2}, \operatorname{diag}(X) \right\rangle + \underbrace{2 \sum_{i=1}^{n} \left\langle x_i, \sum_{k=1}^{K} \eta_k \Delta_{k,i} d_{k,i}^{-2} \right\rangle}_{(A)} \\
&\quad + \underbrace{\sum_{i=1}^{n} \sum_{k=1}^{K} \eta_k \Delta_{k,i}^2 d_{k,i}^{-2}}_{(B)} - \sum_{i=1}^{n} \left\langle \begin{bmatrix} \nu_i & \tilde{x}_i \\ \tilde{x}_i & \tilde{X}_i \end{bmatrix}, \begin{bmatrix} 1 & x_i \\ x_i & X_i \end{bmatrix} \right\rangle.
\end{aligned}$$

Since the reference center $\hat{z}$ can be chosen freely, we select the one that cancel the term in x_i, i.e. the term (A) in the Lagrangian. Specifically, we select $\hat{z}_i$ such that

$$0 = \sum_{k=1}^{K} \eta_k \Delta_{k,i} d_{k,i}^{-2} = \sum_{k=1}^{K} \eta_k (\hat{z}_i - \hat{z}_{k,i}) d_{k,i}^{-2} = \hat{z}_i \sum_{k=1}^{K} \eta_k d_{k,i}^{-2} - \sum_{k=1}^{K} \hat{z}_{k,i} \eta_k d_{k,i}^{-2}.$$

Notice the above is satisfies under the following choice $\hat{z}_i(\eta)$ in Theorem 1

$$\hat{z}_i(\eta) = \frac{\sum_{k=1}^{K} \hat{z}_{k,i} \eta_k d_{k,i}^{-2}}{\sum_{k=1}^{K} \eta_k d_{k,i}^{-2}}$$

In addition, term (B) in the Lagrangian can be expressed as

$$\sum_{i=1}^{n} \sum_{k=1}^{K} \eta_k \Delta_{k,i}^2 d_{k,i}^{-2} = \sum_{i=1}^{n} \sum_{k=1}^{K} \eta_k (\hat{z}_i(\eta) - \hat{z}_{k,i})^2 d_{k,i}^{-2} = \sum_{k=1}^{K} \eta_k \| \operatorname{diag}(d_k)^{-1} (\hat{z}_k - \hat{z}(\eta)) \|_2^2.$$

With the simplification of the two terms above, we fix any $z \in \mathcal{E}_2(\hat{z}_1, d_1) \cap \ldots \cap \mathcal{E}_2(\hat{z}_K, d_K)$, and the the Lagrangian dual of (12) to yield

$$\begin{aligned}
&\max_{\nu, \tilde{x}, \eta} \; -2\tilde{x}^T x - \nu^T \mathbf{1} - \sum_{k=1}^{K} \eta_k \left[1 - \| \operatorname{diag}(d_k)^{-1} (\hat{z}_k - \hat{z}(\eta)) \|_2^2 \right] \\
&\text{s.t.} \quad M = \sum_{k=1}^{K} \eta_k \operatorname{diag}(d_k)^{-2}, \quad \begin{bmatrix} \nu_i & \tilde{x}_i \\ \tilde{x}_i & M_{ii} \end{bmatrix} \succeq 0
\end{aligned}$$

It then again follows from the identity

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

that ν and $\tilde{x}$ can be eliminated to yield

$$\begin{aligned}
&\max_{\eta} \; x^T M x - \sum_{k=1}^{K} \eta_k \left[1 - \| \operatorname{diag}(d_k)^{-1} (\hat{z}_k - \hat{z}(\eta)) \|_2^2 \right] \\
&\text{s.t.} \; M = \sum_{k=1}^{K} \eta_k \operatorname{diag}(d_k)^{-2} \succeq 0.
\end{aligned}$$

Finally, rescaling $\eta \equiv \frac{1}{2}\eta$, and substituting $x = z - \hat{z}(\eta)$ to yield

$$\begin{aligned}
&\max_{\eta} \; \frac{1}{2} \left[x^T M x - \sum_{k=1}^{K} \eta_k \left[1 - \| \operatorname{diag}(d_k)^{-1} (\hat{z}_k - \hat{z}(\eta)) \|_2^2 \right] \right] \\
&\text{s.t.} \; M = \sum_{k=1}^{K} \eta_k \operatorname{diag}(d_k)^{-2} \succeq 0.
\end{aligned} \tag{13}$$

Proof of Theorem 1 We have shown that the dual feasibility problem for the ellipsoid domain (13) can be expressed in the same quadratic form as in (8) in Lemma 1 and (11) in Lemma 2. Therefore, the associated dual problem for Theorem 1 can you written as

$$\max_{\eta,\boldsymbol{\lambda},\boldsymbol{\tau}} \; g(\eta,\boldsymbol{\lambda},\boldsymbol{\tau})^T z + h(\boldsymbol{\lambda},\boldsymbol{\tau}) + \frac{1}{2}\left[x^T M x - \sum_{k=1}^{K} \eta_k \left[1 - \| \operatorname{diag}(d_k)^{-1}(\hat{z}_k - \hat{z}(\eta))\|_2^2\right]\right]$$
$$\text{s.t. } g(\eta,\boldsymbol{\lambda},\boldsymbol{\tau}) = g_s(\boldsymbol{\lambda}) + g_q(\boldsymbol{\tau}), \quad h(\boldsymbol{\lambda},\boldsymbol{\tau}) = h_s(\boldsymbol{\lambda}) + h_q(\boldsymbol{\tau}), \quad c = \Omega\mathbf{1} + (\underline{\Gamma} - \overline{\Gamma})\mathbf{1},$$
$$M = \sum_{k=1}^{K} \eta_k \operatorname{diag}(d_k)^{-2} - \operatorname{diag}(m_s(\boldsymbol{\lambda}) + m_q(\boldsymbol{\tau})) \succeq 0.$$

Here, $g_s(\boldsymbol{\lambda}), h_s(\boldsymbol{\lambda}), g_q(\boldsymbol{\tau}), h_q(\boldsymbol{\tau})$ are evaluated at the center $\hat{z} = \hat{z}(\eta)$. Recall that $x = z - \hat{z}(\eta)$, it then follows form choosing η such that $\sum_{k=1}^{K} \eta_k d_{k,i}^{-2} \geq m_{s,i}(\boldsymbol{\lambda}) + m_{q,i}(\boldsymbol{\tau})$ for all i, then $M \succeq 0$. For every $z \in \mathcal{E}_2(\hat{z}_1, d_1) \cap \ldots \cap \mathcal{E}_2(\hat{z}_K, d_K)$, the desired result follows from

$$(z - \hat{z}(\eta))^T M (z - \hat{z}(\eta)) - \sum_{k=1}^{K} \eta_k \left[1 - \| \operatorname{diag}(d_k)^{-1}(\hat{z}_k - \hat{z}(\eta))\right] \|_2^2$$
$$\geq - \sum_{k=1}^{K} \eta_k \left[1 - \| \operatorname{diag}(d_k)^{-1}(\hat{z}_k - \hat{z}(\eta))\right] \|_2^2$$

and then setting $h(\eta,\boldsymbol{\lambda},\boldsymbol{\tau}) = h(\boldsymbol{\lambda},\boldsymbol{\tau}) - \frac{1}{2}\sum_{k=1}^{K} \eta_k \left[1 - \| \operatorname{diag}(d_k)^{-1}(\hat{z}_k - \hat{z}(\eta))\right] \|_2^2$.

B Construction of Sector and Quadratic Bounds

We construct sector and quadratic bounds for a scalar activation $y = \sigma(z)$ on an interval $[\underline{z}, \overline{z}]$. We assume that $\sigma(z)$ is Lipschitz continuous on the interval $[\underline{z}, \overline{z}]$, which ensures that sector slope $\underline{A}, \overline{A}$ and quadratic coefficient $\underline{Q}, \overline{Q}$ are finite. We note that the Lipschitz continuous assumption covers common activation functions, such as tanh, sigmoid, ReLU, sin, and GeLU.

B.1 Sector Bounds

To construct a sector bound

$$\underline{A} \leq \frac{y - \sigma(\tilde{z})}{z - \tilde{z}} \leq \overline{A} \quad \text{for all } z \in [\underline{z}, \overline{z}]$$

for an anchor $\tilde{z} \in [\underline{z}, \overline{z}]$, we select

$$\underline{A} = \inf_{z \in [\underline{z}, \overline{z}] \setminus \tilde{z}} \frac{\sigma(z) - \sigma(\tilde{z})}{z - \tilde{z}}, \qquad \overline{A} = \sup_{z \in [\underline{z}, \overline{z}] \setminus \tilde{z}} \frac{\sigma(z) - \sigma(\tilde{z})}{z - \tilde{z}}. \tag{14}$$

Under the assumption that $\sigma(z)$ is Lipschitz continuous on the interval $[\underline{z}, \overline{z}]$, both $\underline{A}$ and $\overline{A}$ are finite. By construction, the slopes in (14) define the narrowest sector with anchor $\tilde{z}$ such that the activation $y = \sigma(z)$ satisfies the following quadratic outer approximation

$$[y - \sigma(\tilde{z}) - \underline{A}(z - \tilde{z})]\left[y - \sigma(\tilde{z}) - \overline{A}(z - \tilde{z})\right] \leq 0 \quad \text{for all } z \in [\underline{z}, \overline{z}].$$

B.2 Quadratic Bounds

To construct a quadratic bound

$$\frac{1}{2}\underline{C} z^2 + \underline{P} z + \underline{Q} \leq \sigma(z) \leq \frac{1}{2}\overline{C} z^2 + \overline{P} z + \overline{Q} \quad \text{for all } z \in [\underline{z}, \overline{z}],$$

for a choice of coefficients $\underline{C}, \underline{P}, \overline{C}, \overline{P}$, we select

$$\underline{Q} = \inf_{z \in [\underline{z}, \overline{z}]} \left[\sigma(z) - \frac{1}{2}\underline{C} z^2 - \underline{P} z\right], \qquad \overline{Q} = \sup_{z \in [\underline{z}, \overline{z}]} \left[\sigma(z) - \frac{1}{2}\overline{C} z^2 - \overline{P} z\right]. \tag{15}$$

Under the assumption that $\sigma(z)$ is Lipschitz continuous on the interval $[\underline{z}, \overline{z}]$, both $\underline{Q}$ and $\overline{Q}$ are finite. By construction, (15) best quadratic bounds for the choice of $\underline{C}, \underline{P}, \overline{C}, \overline{P}$.

For twice differentiable activations $\sigma(z)$, we choose any $\tilde{z}$ and construct the following quadratic bounds within $[\underline{z}, \overline{z}]$

$$\frac{1}{2}\underline{C}(z-\tilde{z})^2 + \sigma'(\tilde{z})(z-\tilde{z}) + \sigma(\tilde{z}) \le \sigma(z) \le \sigma(\tilde{z}) + \sigma'(\tilde{z})(z-\tilde{z}) + \frac{1}{2}\overline{C}(z-\tilde{z})^2$$

where the curvature is chosen to be

$$\frac{1}{2}\underline{C} = \min_{z\in[\underline{z},\overline{z}]} k(z), \quad \frac{1}{2}\overline{C} = \max_{z\in[\underline{z},\overline{z}]} k(z) \quad \text{where} \quad k(z) = \frac{\sigma(z) - \sigma(\tilde{z}) - \sigma'(\tilde{z})(z-\tilde{z})}{(z-\tilde{z})^2}.$$

The intuition of this construction is to pick a tangent line $\sigma(\tilde{z}) + \sigma'(\tilde{z})(z-\tilde{z})$ at $z = \tilde{z}$, and bend it by adding the quadratic terms $\frac{1}{2}\underline{C}(z-\tilde{z})^2$, $\frac{1}{2}\overline{C}(z-\tilde{z})^2$ to obtain a valid quadratic lower and upper bound, respectively.

B.3 Multivariate Activations

Most multivariate activations, such as inner products, softmax, can be reduced to affine operations of univariate activations. For example, inner product $a^T b = \frac{1}{4}\sum_i \left[(a_i+b_i)^2 - (a_i-b_i)^2\right]$ and softmax activation $\mathrm{softmax}(z)_i = \exp\Big(z_i - \log\sum_j \exp(z_j)\Big)$. Our quadratic bound propagation framework remain application for those multivariate activations by bounding their univariate components.

C Ellipsoid Propagation

In this section, we describe how to construct ℓ_2 ellipsoids to overapproximate the uncertainty set of preactivations

$$\mathcal{E}_2(\hat{z}^{(\ell)}, d_z^{(\ell)}) = \{\hat{z}^{(\ell)} + \mathrm{diag}(d_z^{(\ell)})v \mid \|v\|_2 \le 1\} \supseteq \mathcal{Z}_\ell = \{W^{(\ell)}x + b^{(\ell)} \mid x \in \mathcal{X}_{\ell-1}\}$$

and postactivations

$$\mathcal{E}_2(\hat{x}^{(\ell)}, d_x^{(\ell)}) = \{\hat{x}^{(\ell)} + \mathrm{diag}(d_x^{(\ell)})u \mid \|u\|_2 \le 1\} \supseteq \mathcal{X}_\ell = \{\sigma(z) \mid z \in \mathcal{Z}_\ell\}.$$

Without loss of generality, we assume $d_x^{(\ell-1)} > 0$ and $d_z^{(\ell)} > 0$ for all $\ell = 1\ldots, N$. Specifically, if $d_{x,i}^{(\ell-1)} = 0$, then $x_i^{(\ell-1)} = \hat{x}_i^{(\ell-1)}$ is a constant, which can be removed by absorbing its contribution into the bias $b^{(\ell)} \leftarrow \hat{x}_i^{(\ell-1)}(W^{(\ell)}e_i) + b^{(\ell)}$ as in

$$z^{(\ell)} = \sum_j \left[x_j^{(\ell-1)}(W^{(\ell)}e_j)\right] + b^{(\ell)} = \sum_{j\neq i} \left[x_j^{(\ell-1)}(W^{(\ell)}e_j)\right] + \hat{x}_i^{(\ell-1)}(W^{(\ell)}e_i) + b^{(\ell)}. \tag{16}$$

After removing the constant component in $x^{(\ell-1)}$, we have $d_x^{(\ell-1)} > 0$. Similarly, if $d_{z,i}^{(\ell)} = 0$, then $z_i^{(\ell)} = \hat{z}_i^{(\ell)}$ is a constant, which can be removed by absorbing its contribution into the bias $b^{(\ell+1)} \leftarrow \sigma(\hat{z}_i^{(\ell)})(W^{(\ell+1)}e_i) + b^{(\ell+1)}$ similar to (16).

C.1 Propagating through linear layer

We want to construct an ellipsoid outer approximation of the preactivation uncertainty set

$$\mathcal{E}_2(\hat{z}, d_z) \supseteq \mathcal{Z} = \{Wx + b \mid x \in \mathcal{X}\} \tag{17}$$

where the input set $\mathcal{X} = \mathcal{E}_p(\hat{x}, d_x)$ is an ℓ_p ellipsoid with $p = 1$, $p = 2$, or $p = \infty$. In particular, choosing $\hat{z} = W\hat{x} + b$ and substituting the input parametrization gives

$$\mathcal{Z} = \{W\hat{x} + b + W\,\mathrm{diag}(d_x)u \mid \|u\|_p \le 1\} = \{\hat{z} + W\,\mathrm{diag}(d_x)u \mid \|u\|_p \le 1\}.$$

The containment (17) holds if and only if for every $\|u\|_p \leq 1$, there exists a $\|v\|_2 \leq 1$ such that $\hat{z} + \operatorname{diag}(d_z)v = \hat{z} + W \operatorname{diag}(d_x)u$. Since $d_z > 0$, this equality uniquely determines v. Thus, any ellipsoid $\mathcal{E}_2(\hat{z}, d_z)$ with semiaxes d_z satisfying

$$1 \geq \max_{\|u\|_p \leq 1} \| \operatorname{diag}(d_z)^{-1} W \operatorname{diag}(d_x) u\|_2$$

is an outer approximation of $\mathcal{Z}$. This condition provides a systematic way to construct d_z. Given a positive vector a_z specifying the desired semiaxis ratios, the desired $d_z = \kappa a_z$ can be obtained by applying a scaling κ satisfies $\kappa \geq \max_{\|u\|_p \leq 1} \| \operatorname{diag}(a_z)^{-1} W \operatorname{diag}(d_x) u\|_2$.

We choose a_z as the ratio of the width of $\mathcal{Z}$

$$a_{z,i} = \max_{z \in \mathcal{Z}} |z_i - \hat{z}_i| = \max_{\|u\|_p \leq 1} |e_i^T W \operatorname{diag}(d_x) u| = \|e_i^T W \operatorname{diag}(d_x)\|_q$$

where $1/p + 1/q = 1$. We summarize how to choose a_z and κ as follow.

ℓ_1 ellipsoid input set Let $\mathcal{X} = \mathcal{E}_1(\hat{x}, d_x)$, a simple choice of $\mathcal{E}_2(\hat{z}, d_z)$ is

$$d_z = \kappa a_z, \quad \kappa = \max_j \| \operatorname{diag}(a_z)^{-1} W \operatorname{diag}(d_x) e_j\|_2, \quad a_{z,i} = \max_j |W_{ij}| d_{x,j}.$$

Here, κ is chosen based on the equality $\kappa = \max_{\|u\|_1 \leq 1} \|Au\|_2 = \max_j \|Ae_j\|_2$ where $A = \operatorname{diag}(a_z)^{-1} W \operatorname{diag}(d_x)$.

ℓ_2 ellipsoid input set Let $\mathcal{X} = \mathcal{E}_2(\hat{x}, d_x)$, a simple choice of $\mathcal{E}_2(\hat{z}, d_z)$ is

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

Here, κ is chosen based on the equality $\kappa = \max_{\|u\|_2 \leq 1} \|Au\|_2 = \|A\|_2$ where $\|A\|_2$ denotes the spectral norm of the matrix A, and $A = \operatorname{diag}(a_z)^{-1} W \operatorname{diag}(d_x)$.

ℓ_∞ ellipsoid input set Let $\mathcal{X} = \mathcal{E}_\infty(\hat{x}, d_x)$, a simple choice of $\mathcal{E}_2(\hat{z}, d_z)$ is

$$d_z = \kappa a_z, \quad \kappa = \sqrt{d_x^T |W^T \operatorname{diag}(a_z)^{-2} W| d_x}, \quad a_z = |W| d_x.$$

In this case, κ is chosen via the upper bound $\kappa^2 = \mathbf{1}^T |A^T A| \mathbf{1} \geq \max_{\|u\|_\infty \leq 1} \|Au\|_2^2$ where $A = \operatorname{diag}(a_z)^{-1} W \operatorname{diag}(d_x)$.

C.2 Propagating through nonlinear layer

We consider constructing an ℓ_2 ellipsoid outer approximation of the postactivation uncertainty set

$$\mathcal{E}_2(\hat{x}, d_x) \supseteq \mathcal{X} = \{\sigma(z) \mid z \in \mathcal{Z}\} \tag{18}$$

when $\mathcal{Z} = \mathcal{E}_2(\hat{z}, d_z)$ is an ellipsoid. In particular, we construct ellipsoid outer approximation (18) via the following simple choice

$$\hat{x} = \sigma(\hat{z}), \quad d_x = d_z \odot \operatorname{Lip}(\sigma)$$

where the i-th element of $\operatorname{Lip}_i(\sigma)$ denotes the local Lipschitz constant of $\sigma(\cdot)$ centered at $\hat{z}_i$. In particular, when the preactivation bounds $z_i \in [\underline{z}_i, \overline{z}_i]$ are available, $\operatorname{Lip}_i(\sigma)$ is defined as the average derivative between $\hat{z}_i$ and $z_i \in [\underline{z}_i, \overline{z}_i]$

$$\operatorname{Lip}_i(\sigma) = \sup_{z \in [\underline{z}_i, \overline{z}_i] \setminus \{\hat{z}_i\}} \frac{|\sigma(z) - \sigma(\hat{z}_i)|}{|z - \hat{z}_i|}.$$

If the preactivation bounds are not available, $\operatorname{Lip}_i(\sigma)$ can be replaced with the global Lipschitz constant of $\sigma(\cdot)$.

D Connection to Chiu et al. (2025)

The linear bounding oracle in Chiu et al. (2025, Theorem 4.1) can by recovered by choosing the sector bound

$$\underline{A}_i = 0, \quad \overline{A}_i = 1, \quad \tilde{Z}_i = 0,$$

and quadratic bounds

$$(\underline{C}_{i1}, \underline{P}_{i1}, \underline{Q}_{i1}) = (0, 0, 0), \quad (\overline{C}_{i2}, \overline{P}_{i2}, \overline{Q}_{i2}) = (0, 1, 0),$$

which together give the quadratic outer approximation of ReLU

$$y_i(y_i - z_i) \le 0, \quad y_i \ge 0, \quad y_i \ge z_i.$$

The bounding oracle in Chiu et al. (2025, Theorem 4.1) consider a single ball domain $\mathcal{E}_2(\hat{z}, \rho\mathbf{1})$ with radius $\rho > 0$. To simplify notation, let $\alpha_i = \underline{\Gamma}_{i1}$ and $\beta_i = \underline{\Gamma}_{i2}$, and suppress the sector index. Substituting the above bounds into Theorem 1 gives $m_s = \Lambda$, $m_q = 0$, and $\Omega = c - \alpha - \beta$. The resulting oracle is $g^\top z + h \le c^\top \mathrm{ReLU}(z)$, where

$$g_i = G_i + \beta_i - \Lambda_i \hat{z}_i, \quad h = -\frac{1}{2}\left[\eta - \sum_i \Lambda_i \hat{z}_i^2 + \sum_i \frac{\max\{G_i^2, (c_i - \alpha_i - \beta_i - G_i)^2\}}{\Lambda_i}\right],$$

with $\alpha, \beta \ge 0$, $\Lambda_i > 0$, and $\eta \ge \rho^2 \max_i \Lambda_i$. We note that G and Λ are vectors as there is only one sector in this case.

To recover SDP-CROWN, set $\Lambda_i = \lambda > 0$ and $\eta = \lambda\rho^2$, and therefore $G_i = g_i + \lambda\hat{z}_i - \beta_i$. It then follows form the identity $\min_{u,v\ge 0} \max\{(a-u)^2, (b-v)^2\} = \min\{a, b, 0\}^2$ that setting $a = c_i - g_i - \lambda\hat{z}_i$ and $b = g_i + \lambda\hat{z}_i$ eliminates $\alpha_i \ge 0$ and $\beta_i \ge 0$, which yields

$$g^T z - \frac{1}{2}\left[\lambda(\rho^2 - \|\hat{z}\|_2^2) + \frac{1}{\lambda}\|\min\{c - g - \lambda\hat{z},\ g + \lambda\hat{z},\ 0\}\|_2^2\right] \le c^T \mathrm{ReLU}(z) \quad \text{for all } \|z - \hat{z}\|_2 \le \rho.$$

This recovers the linear bounding oracle in Chiu et al. (2025, Theorem 4.1).

E Derivation for Dual Problems

E.1 Dual problem for Lemma 1

Recall that we have the primal problem

$$\min_{\substack{y,x,X\in\mathbb{R}^n \\ Z,r,R\in\mathbb{R}^{n\times s}}} c^T y \quad \text{s.t.} \quad \begin{array}{c} \frac{\overline{A}_{ij}+\underline{A}_{ij}}{2} Z_{ij} - \frac{\overline{A}_{ij}-\underline{A}_{ij}}{2} r_{ij} \le y_i - \sigma(\tilde{Z}_{ij}) \le \frac{\overline{A}_{ij}+\underline{A}_{ij}}{2} Z_{ij} + \frac{\overline{A}_{ij}-\underline{A}_{ij}}{2} r_{ij}, \\ |Z_{ij}| \le r_{ij}, \quad R_{ij} \le X_i + 2Z_{ij}\hat{B}_{ij} - \hat{B}_{ij}^2, \quad Z_{ij} = x_i + \hat{B}_{ij}, \\ \sum_{i=1}^n \frac{X_i}{d_i^2} \le 1, \quad \begin{bmatrix} 1 & x_i \\ x_i & X_i \end{bmatrix} \succeq 0, \quad \begin{bmatrix} 1 & r_{ij} \\ r_{ij} & R_{ij} \end{bmatrix} \succeq 0 \end{array}$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, s$. Here $Z_{ij} = x_i + \hat{B}_{ij}$, $\hat{B}_{ij} = \hat{z}_i - \tilde{Z}_{ij}$ and $x = z - \hat{z}$.

Let $S, T, \hat{S}, \hat{T}, \Lambda \in \mathbb{R}_+^{n\times s}$ denote the dual variable associated with the first five inequality constraints, respectively. Let $\eta \ge 0$ denote the dual variables for $\sum_{i=1}^n \frac{X_i}{d_i^2}$. Finally, let $\begin{bmatrix} \nu_i & \tilde{x}_i \\ \tilde{x}_i & \tilde{X}_i \end{bmatrix} \succeq 0$ and $\begin{bmatrix} \mu_{ij} & \tilde{r}_{ij} \\ \tilde{r}_{ij} & \tilde{R}_{ij} \end{bmatrix} \succeq 0$ denote the dual variables for the PSD blocks $\begin{bmatrix} 1 & x_i \\ x_i & X_i \end{bmatrix} \succeq 0$, $\begin{bmatrix} 1 & r_{ij} \\ r_{ij} & R_{ij} \end{bmatrix} \succeq 0$,

respectively. The Lagrangian is given by:

$$\underbrace{\sum_{i=1}^{n}\sum_{j=1}^{s}\left[\Lambda_{ij}\hat{B}_{ij}^2+\sigma(\tilde{Z}_{ij})(S_{ij}-T_{ij})\right]}_{h}-\eta+\langle c-(S-T)\mathbf{1},y\rangle$$
$$+\left\langle Z,\underbrace{(S-T)\odot\frac{\overline{A}+\underline{A}}{2}+(\hat{S}-\hat{T})-2\hat{B}\odot\Lambda}_{G}\right\rangle+\left\langle \eta\,\mathrm{diag}(d)^{-2}-\mathrm{diag}(\Lambda\mathbf{1}),\mathrm{diag}(X)\right\rangle$$
$$+\left\langle r,\underbrace{-(S+T)\odot\frac{\overline{A}-\underline{A}}{2}-(\hat{S}+\hat{T})}_{\Phi}\right\rangle+\langle\Lambda,R\rangle$$
$$-\sum_{i=1}^{n}\left\langle\begin{bmatrix}\nu_i & \tilde{x}_i\\ \tilde{x}_i & \tilde{X}_i\end{bmatrix},\begin{bmatrix}1 & x_i\\ x_i & X_i\end{bmatrix}\right\rangle-\sum_{i=1}^{n}\sum_{j=1}^{s}\left\langle\begin{bmatrix}\mu_{ij} & \tilde{r}_{ij}\\ \tilde{r}_{ij} & \tilde{R}_{ij}\end{bmatrix},\begin{bmatrix}1 & r_{ij}\\ r_{ij} & R_{ij}\end{bmatrix}\right\rangle$$

Fixing z, and taking Lagrangian dual yield

$$c=(S-T)\mathbf{1},\quad \Phi-2\tilde{r}=0,\quad \Lambda-\tilde{R}=0,\quad \eta\,\mathrm{diag}(d)^{-2}-\mathrm{diag}(\Lambda\mathbf{1})-\mathrm{diag}(\tilde{X})=0,$$

which gives

$$\begin{aligned}\max_{G,h,\Lambda,\eta,\tilde{x},\nu,\mu}\ &\langle G,Z\rangle+h-2\tilde{x}^Tx-\nu^T\mathbf{1}-\mathbf{1}^T\mu\mathbf{1}-\eta\\ \text{s.t.}\quad &c=(S-T)\mathbf{1},\quad M=\eta\,\mathrm{diag}(d)^{-2}-\mathrm{diag}(\Lambda\mathbf{1}),\\ &\begin{bmatrix}\nu_i & \tilde{x}_i\\ \tilde{x}_i & M_{ii}\end{bmatrix}\succeq 0,\quad \begin{bmatrix}\mu_{ij} & \Phi_{ij}/2\\ \Phi_{ij}/2 & \Lambda_{ij}\end{bmatrix}\succeq 0\end{aligned}$$

where G,h,Φ are defined above. It then follows from the identity

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

that ν and $\tilde{x}$ can be eliminated to yield

$$\begin{aligned}\max_{G,h,\Lambda,\eta,\mu}\ &\langle G,Z\rangle+h-\mathbf{1}^T\mu\mathbf{1}+x^TMx-\eta\\ \text{s.t.}\quad &c=(S-T)\mathbf{1},\quad M=\eta\,\mathrm{diag}(d)^{-2}-\mathrm{diag}(\Lambda\mathbf{1})\succeq 0,\quad\begin{bmatrix}\mu_{ij} & \Phi_{ij}/2\\ \Phi_{ij}/2 & \Lambda_{ij}\end{bmatrix}\succeq 0\end{aligned}$$

The variables $S,T,\hat{S},\hat{T},\in\mathbb{R}_+^{n\times s}$ can be eliminated by optimizing over Φ. The can be seen by first applying the following variable transformation

$$\begin{aligned}S\geq 0,\quad T\geq 0\quad&\Longleftrightarrow\quad \Theta=S+T,\quad \Omega=S-T,\quad \Theta\geq|\Omega|,\\ \hat{S}\geq 0,\quad \hat{T}\geq 0\quad&\Longleftrightarrow\quad \hat{\Theta}=\hat{S}+\hat{T},\quad \hat{\Omega}=\hat{S}-\hat{T},\quad \hat{\Theta}\geq|\hat{\Omega}|.\end{aligned}$$

Now, notice that each Θ_{ij} and $\hat{\Theta}_{ij}$ in Φ_{ij} can be optimized in closed-form as

$$\begin{aligned}\min_{\Theta_{ij}\geq|\Omega_{ij}|,\hat{\Theta}_{ij}\geq|\hat{\Omega}_{ij}|}\Phi_{ij}^2&=\min_{\Theta_{ij}\geq|\Omega_{ij}|,\hat{\Theta}_{ij}\geq|\hat{\Omega}_{ij}|}\left(-\Theta_{ij}\frac{\overline{A}_{ij}-\underline{A}_{ij}}{2}-\hat{\Theta}_{ij}\right)^2\\ &=\min\left\{-|\Omega_{ij}|\frac{\overline{A}_{ij}-\underline{A}_{ij}}{2}-|\hat{\Omega}_{ij}|,\ 0\right\}^2=\left(|\Omega_{ij}|\frac{\overline{A}_{ij}-\underline{A}_{ij}}{2}+|\hat{\Omega}_{ij}|\right)^2\\ &=\max\left\{\left|\Omega_{ij}\frac{\overline{A}_{ij}-\underline{A}_{ij}}{2}+\hat{\Omega}_{ij}\right|,\left|\Omega_{ij}\frac{\overline{A}_{ij}-\underline{A}_{ij}}{2}-\hat{\Omega}_{ij}\right|\right\}^2.\end{aligned}$$

where the second and third equality follows from $\frac{\overline{A}_{ij}-\underline{A}_{ij}}{2}\geq 0$ since $\overline{A}_{ij}\geq\underline{A}_{ij}$ by definition of sector bounds. The last equality follows from the identity $(|a|+|b|)^2=\max\{|a+b|,|a-b|\}^2$ for any $a,b\in\mathbb{R}$.

Substituting $G = \Omega \odot \frac{\overline{A}+\underline{A}}{2} + \hat{\Omega} - 2\hat{B} \odot \Lambda$ into the above expression yield

$$\Phi_{ij} = \max\left\{\left|\Omega_{ij}\overline{A}_{ij} - G_{ij} - 2\hat{B}_{ij}\Lambda_{ij}\right|, \left|\Omega_{ij}\underline{A}_{ij} - G_{ij} - 2\hat{B}_{ij}\Lambda_{ij}\right|\right\}$$

Rescaling $\Lambda \equiv \frac{1}{2}\Lambda$, $\mu \equiv \frac{1}{2}\mu$, $\eta \equiv \frac{1}{2}\eta$, the Lagrangian dual becomes

$$\begin{aligned}
\max_{G,\Omega,\Lambda,\eta,\mu}\ & \langle G, Z\rangle + h + \frac{1}{2}(x^T M x - \eta) \\
\text{s.t.}\quad & h = \sum_{i=1}^{n}\sum_{j=1}^{s}\left[\sigma(\tilde{Z}_{ij})\Omega_{ij} + \frac{\Lambda_{ij}\hat{B}_{ij}^2 - \mu_{ij}}{2}\right] \\
& \Phi_{ij} = \max\left\{\left|\Omega_{ij}\overline{A}_{ij} - G_{ij} - \hat{B}_{ij}\Lambda_{ij}\right|, \left|\Omega_{ij}\underline{A}_{ij} - G_{ij} - \hat{B}_{ij}\Lambda_{ij}\right|\right\} \\
& c = \Omega\mathbf{1}, \quad M = \eta\,\mathrm{diag}(d)^{-2} - \mathrm{diag}(\Lambda\mathbf{1}) \succeq 0, \quad \begin{bmatrix}\mu_{ij} & \Phi_{ij}\\ \Phi_{ij} & \Lambda_{ij}\end{bmatrix} \succeq 0
\end{aligned}$$

Recall that $Z_{ij} = x_i + \hat{B}_{ij}$, $\hat{B}_{ij} = \hat{z}_i - \tilde{Z}_{ij}$ and $x = z - \hat{z}$. Notice that for 2×2 matrix, $X \succeq 0$ if and only if $\det(X) \geq 0$ and $\mathrm{tr}(X) \geq 0$; therefore, the optimal μ_{ij} satisfies $\mu_{ij}\Lambda_{ij} = \Phi_{ij}^2$. Substituting them into the above and apply a reparameterization $G_{ij} \equiv G_{ij} + \hat{B}_{ij}\Lambda_{ij}$ yield the desired dual problem

$$\begin{aligned}
\max_{\eta,\boldsymbol{\lambda}}\ & g_s(\boldsymbol{\lambda})^T z + h_s(\boldsymbol{\lambda}) + \frac{1}{2}((z-\hat{z})^T M(z-\hat{z}) - \eta) \\
\text{s.t.}\quad & g_s(\boldsymbol{\lambda}) = (G + \Lambda \odot \tilde{Z})\mathbf{1} - \hat{z} \odot m_s(\boldsymbol{\lambda}), \quad \Phi_{ij} = \max\left\{|\Omega_{ij}\underline{A}_{ij} - G_{ij}|, |\Omega_{ij}\overline{A}_{ij} - G_{ij}|\right\} \\
& h_s(\boldsymbol{\lambda}) = \sum_{i=1}^{n}\sum_{j=1}^{s}\left[\Omega_{ij}\sigma(\tilde{Z}_{ij}) - G_{ij}\tilde{Z}_{ij} - \frac{\Phi_{ij}^2 + (\Lambda_{ij}\tilde{Z}_{ij})^2}{2\Lambda_{ij}}\right] + \frac{1}{2}\hat{z}^T\,\mathrm{diag}(m_s(\boldsymbol{\lambda}))\hat{z}. \\
& c = \Omega\mathbf{1}, \quad m_s(\boldsymbol{\lambda}) = \Lambda\mathbf{1}, \quad M = \eta\,\mathrm{diag}(d)^{-2} - \mathrm{diag}(m_s(\boldsymbol{\lambda})) \succeq 0.
\end{aligned}$$

with $\boldsymbol{\lambda} = (\Lambda, G, \Omega)$.

E.2 Dual problem for Lemma 2

Recall that we have the primal problem

$$\min_{\substack{y,x,X\in\mathbb{R}^n\\ z,Z\in\mathbb{R}^n}} c^T y \ \text{ s.t. } \begin{array}{c} \frac{1}{2}\underline{C}_{ij}Z_i + \underline{P}_{ij}z_i + \underline{Q}_{ij} \leq y_i \leq \frac{1}{2}\overline{C}_{ij}Z_i + \overline{P}_{ij}z_i + \overline{Q}_{ij}, \quad z = x + \hat{z} \\ Z_i = X_i + 2z_i\hat{z}_i - \hat{z}_i^2, \quad \sum_{i=1}^n \frac{X_i}{d_i^2} \leq 1, \quad \begin{bmatrix}1 & x_i\\ x_i & X_i\end{bmatrix} \succeq 0 \end{array}$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, q$.

Let $\underline{\Gamma}, \overline{\Gamma} \in \mathbb{R}_+^{n\times q}$ denote the dual variable associated with the first two inequality constraints, respectively. Let $\eta \geq 0$ denote the dual variables for $\sum_{i=1}^n \frac{X_i}{d_i^2}$. Finally, let $\begin{bmatrix}\nu_i & \tilde{x}_i\\ \tilde{x}_i & \tilde{X}_i\end{bmatrix} \succeq 0$ denote the dual variables for the PSD blocks $\begin{bmatrix}1 & x_i\\ x_i & X_i\end{bmatrix} \succeq 0$. The Lagrangian is given by:

$$\begin{aligned}
& \underbrace{\langle \underline{Q}, \underline{\Gamma}\rangle - \langle \overline{Q}, \overline{\Gamma}\rangle + \frac{1}{2}\left\langle \hat{z}\odot\hat{z}, (\overline{C}\odot\overline{\Gamma} - \underline{C}\odot\underline{\Gamma})\mathbf{1}\right\rangle}_{h} - \eta + \left\langle c - (\underline{\Gamma} - \overline{\Gamma})\mathbf{1}, y\right\rangle \\
& + \left\langle z, \underbrace{(\underline{P}\odot\underline{\Gamma} - \overline{P}\odot\overline{\Gamma})\mathbf{1} - \hat{z}\odot(\overline{C}\odot\overline{\Gamma} - \underline{C}\odot\underline{\Gamma})\mathbf{1}}_{g}\right\rangle \\
& + \left\langle \eta\,\mathrm{diag}(d)^{-2} - \frac{1}{2}\mathrm{diag}((\overline{C}\odot\overline{\Gamma} - \underline{C}\odot\underline{\Gamma})\mathbf{1}), \mathrm{diag}(X)\right\rangle \\
& - \sum_{i=1}^n \left\langle \begin{bmatrix}\nu_i & \tilde{x}_i\\ \tilde{x}_i & \tilde{X}_i\end{bmatrix}, \begin{bmatrix}1 & x_i\\ x_i & X_i\end{bmatrix}\right\rangle
\end{aligned}$$

Fixing z, and taking Lagrangian dual yield

$$c = (\underline{\Gamma} - \overline{\Gamma})\mathbf{1}, \quad \eta \operatorname{diag}(d)^{-2} - \frac{1}{2}\operatorname{diag}((\overline{C} \odot \overline{\Gamma} - \underline{C} \odot \underline{\Gamma})\mathbf{1}) - \operatorname{diag}(\tilde{X}) = 0,$$

which gives

$$\max_{g,h,\underline{\Gamma},\overline{\Gamma},\eta,\tilde{x},\nu} \langle g, z\rangle + h - 2\tilde{x}^T x - \nu^T \mathbf{1} - \eta$$
$$\text{s.t.} \quad c = (\underline{\Gamma} - \overline{\Gamma})\mathbf{1}, \quad M = \eta \operatorname{diag}(d)^{-2} - \frac{1}{2}\operatorname{diag}((\overline{C} \odot \overline{\Gamma} - \underline{C} \odot \underline{\Gamma})\mathbf{1}), \quad \begin{bmatrix} \nu_i & \tilde{x}_i \\ \tilde{x}_i & M_{ii} \end{bmatrix} \succeq 0$$

It then again follows from the identity

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

that ν and $\tilde{x}$ can be eliminated to yield

$$\max_{g,h,\underline{\Gamma},\overline{\Gamma},\eta} \langle g, z\rangle + h + x^T M x - \eta$$
$$\text{s.t.} \quad c = (\underline{\Gamma} - \overline{\Gamma})\mathbf{1}, \quad M = \eta \operatorname{diag}(d)^{-2} - \frac{1}{2}\operatorname{diag}((\overline{C} \odot \overline{\Gamma} - \underline{C} \odot \underline{\Gamma})\mathbf{1}) \succeq 0$$

Rescaling $\eta \equiv \frac{1}{2}\eta$, and substituting $x = z - \hat{z}$ to yield the desired result

$$\begin{aligned} \max_{\eta,\boldsymbol{\tau}} \; & g_q(\boldsymbol{\tau})^T z + h_q(\boldsymbol{\tau}) + \frac{1}{2}((z-\hat{z})^T M (z-\hat{z}) - \eta) \\ \text{s.t.} \; & g_q(\boldsymbol{\tau}) = (\underline{P} \odot \underline{\Gamma} - \overline{P} \odot \overline{\Gamma})\mathbf{1} - \hat{z} \odot m_q(\boldsymbol{\tau}), \\ & h_q(\boldsymbol{\tau}) = \langle \underline{Q}, \underline{\Gamma}\rangle - \langle \overline{Q}, \overline{\Gamma}\rangle + \frac{1}{2}\hat{z}^T \operatorname{diag}(m_q(\boldsymbol{\tau}))\hat{z}. \\ & c = (\underline{\Gamma} - \overline{\Gamma})\mathbf{1}, \quad m_q(\boldsymbol{\tau}) = (\overline{C} \odot \overline{\Gamma} - \underline{C} \odot \underline{\Gamma})\mathbf{1}, \\ & M = \eta \operatorname{diag}(d)^{-2} - \operatorname{diag}(m_q(\boldsymbol{\tau})) \succeq 0. \end{aligned}$$

with $\boldsymbol{\tau} = (\underline{\Gamma}, \overline{\Gamma})$.