Hierarchical End-to-End Taylor Bounds for Complete Neural Network Verification

Taha Entesari TENTESA1@JHU.EDU
Mahyar Fazlyab MAHYARFAZLYAB@JHU.EDU
Johns Hopkins University

Editors: G. Sukhatme, L. Lindemann, S. Tu, A. Wierman, N. Atanasov

Abstract

Reachability analysis of neural networks, which seeks to compute or bound the set of outputs attainable over a given input domain, is central to certifying safety and robustness in learning-enabled physical systems. Since exact reachable set computation is generally intractable, existing methods typically rely on tractable overapproximations. Examining the state of the art for smooth, twice-differentiable networks, we observe that existing approaches exploit at most second-order information and do not systematically leverage higher-order information. In this work, we introduce HITAB, a novel verification framework that exploits second-order smoothness through both the Hessian, $\nabla^2 f$, and its Lipschitz constant, $L_{\nabla^2 f}$. We further develop a unified hierarchy of zeroth-, first-, and second-order bounds, together with precise conditions under which higher-order approximations yield provable improvements. Our main technical contribution is a compositional procedure for efficiently bounding $L_{\nabla^2 f}$ in deep neural networks via layerwise propagation of curvature bounds. We extend the framework to both ℓ_2- and ℓ_∞-constrained input sets and show how it can be integrated into branch-and-bound verification pipelines. To our knowledge, this is the first practical reachability analysis framework for smooth neural networks that systematically exploits Lipschitz continuity of curvature, leading to tighter and more informative safety certificates.

Keywords: Neural network reachability, Formal verification, Higher-order smoothness, Hessian Lipschitz continuity, Taylor models

1. Introduction

Ensuring the reliability of neural network-based systems is essential for their deployment in safety-critical applications, including autonomous driving Ibrahum et al. (2024), medical diagnosis Javed et al. (2024), and control Tambon et al. (2022). A central verification question in this setting is the following: given a neural network and a bounded set of admissible inputs, can we efficiently compute or tightly bound the set of possible outputs? This problem arises in many settings, ranging from certifying robustness against adversarial perturbations Wang et al. (2021) to verifying the safety of neural network controllers in closed-loop dynamical systems Entesari et al. (2023).

In this work, we study this question through the lens of reachability analysis. Given a neural network-represented function $f : \mathbb{R}^n \to \mathbb{R}$ and an input set $\mathcal{X}$, we consider the problem of upper bounding

$$\max_{x \in \mathcal{X}} f(x).$$

Such bounds provide certificates on the reachable output range of the network over $\mathcal{X}$ and form a core primitive in safety verification and robustness analysis.

Although this problem is generally nonconvex and computationally intractable, substantial progress has been made in developing tractable overapproximation methods that provide provable upper

bounds on the optimal value. Such bounds enable sound, though conservative, certification of neural network behavior and underpin many modern formal verification frameworks Zhang et al. (2018); Xu et al. (2020); Wang et al. (2021); Entesari et al. (2023).

Despite the extensive literature on verification and reachability analysis for non-smooth networks, particularly ReLU networks, comparatively less attention has been given to smooth and differentiable architectures. Existing methods specialized to differentiable networks include the branch-and-bound framework GenBaB Shi et al. (2025) and several approaches that exploit derivative information to obtain certified bounds Zhang et al. (2019); Singla and Feizi (2020); Entesari et al. (2024); Sharifi and Fazlyab (2024). However, these approaches rely on at most local first- or second-derivative information and do not systematically exploit higher-order smoothness to control approximation error.

Contributions In this work, we develop a practical framework for reachability analysis of smooth neural networks that systematically exploits higher-order smoothness. Our main contributions are:

- **A compositional method for bounding the Hessian Lipschitz constant.** We develop a novel layerwise procedure for bounding the Lipschitz constant of the Hessian, $L_{\nabla^2 f}$, in scalar-valued feedforward neural networks. This provides a practical way to quantify higher-order smoothness at the network level.
- **An end-to-end second-order Taylor model with certified cubic remainder bounds.** We derive an end-to-end second-order Taylor approximation of the network that preserves the local gradient and Hessian of the network, and we bound the approximation error using the Lipschitz continuity of the Hessian. This yields a certified local upper bound whose remainder scales cubically with the input perturbation size.
- **A hierarchy of reachability bounds.** We develop a unified hierarchy of zeroth-, first-, and second-order reachability bounds, clarifying when higher-order information leads to tighter certificates.

Together, these components define HITAB (**Hi**erarchical End-to-End **Ta**ylor **B**ounds), a framework that yields tighter reachability bounds for smooth neural networks and can be embedded within complete branch-and-bound verification pipelines.

1.1. Related Work

Reachability Analysis of Differentiable Networks Unlike piecewise linear neural networks, whose complete verification can be formulated using mixed-integer linear programs, smooth neural networks lack such general complete formulations. Consequently, much research has focused on incomplete reachability analysis.

These methods often rely on mathematical abstractions or optimization-based relaxations. For instance, Hu et al. (2020) presents a semidefinite program (SDP) approach for networks with slope-bounded activations, covering both piecewise linear and differentiable functions. ReachNN Huang et al. (2019) utilizes Bernstein polynomials for Lipschitz networks, while Kochdumper et al. (2023) employs polynomial zonotopes to capture non-convex reachable sets for ReLU, sigmoid, and $\tanh$ functions. Other methods are highly specialized, such as Ivanov et al. (2019, 2021), which target

sigmoid and $\tanh$ activations specifically. These methods generally follow a common pattern: abstract individual neurons and layers first, then compose them to obtain the end-to-end map. An alternative is to abstract the end-to-end map directly and then relax it. Along these lines, Sharifi and Fazlyab (2024) recently proposed a framework that leverages the network's gradient information to derive an end-to-end abstraction via a first-order Taylor expansion. Our approach builds directly on this perspective, constructing an end-to-end *third*-order Taylor abstraction that captures both gradient and curvature information.

Efforts to achieve complete verification for smooth networks often build upon these analyses. ReachLipBnB Entesari et al. (2023), for example, wraps a Lipschitz-based analysis in a branching strategy to yield a complete verifier. Similarly, GenBaB Shi et al. (2025) adapts the CROWN-like Zhang et al. (2018) Branch-and-Bound (BaB) framework, using element-wise bounding for smooth activations.

Lipschitz Estimation Lipschitz constant estimation for neural networks has been at the center of robustness analysis since the advent of adversarial attacks Szegedy et al. (2013). However, the spectral bound of Szegedy et al. (2013) proved to be highly conservative with limited practical utility. Subsequent works focused on obtaining local Lipschitz constants, which can significantly reduce conservatism at the expense of increased computation, either through bound propagation techniques Huang et al. (2021); Shi et al. (2022) or more intensive mixed-integer linear programs Jordan and Dimakis (2020). Later, Fazlyab et al. (2019) introduced LipSDP, an SDP to calculate global and local Lipschitz constants for neural networks with *slope-bounded* activations. This framework inspired a wide range of follow-up works, expanding and specializing the LipSDP algorithm to specific architectures Pauli et al. (2023), exploiting the structure of layers Araujo et al. (2023); Wang and Manchester (2023), extending the methodology to certain non-slope-restricted activation functions Pauli et al. (2024), and providing improved spectral-like bounds Fazlyab et al. (2023).
Beyond the Lipschitz constant of a neural network, estimating the Lipschitz constant of the gradient (or equivalently, the norm of the Hessian) has also received attention. Singla and Feizi (2020) first addressed this problem and provided an algorithm for computing bounds on the Hessian of scalar logits of a feedforward network. Sharifi and Fazlyab (2024) studied local versions of this bound. Entesari et al. (2024) presented a different approach, providing a formulation for calculating the Lipschitz constant of the gradient of general function compositions, with specializations to neural networks. Here, we go beyond these and bound the Lipschitz constant of the *second* derivative.

2. Background and Problem Statement

We consider a twice-differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ indexed by a neural network and the problem of upper bounding its maximum over a Euclidean ball centered at a nominal point $x_c \in \mathbb{R}^n$:

$$\max_{x \in \mathcal{B}(x_c,\varepsilon)} f(x), \quad \text{where} \quad \mathcal{B}(x_c,\varepsilon) \coloneqq \{x \in \mathbb{R}^n \mid \|x - x_c\|_2 \le \varepsilon\}. \tag{1}$$

This problem arises in a variety of applications, including certification against adversarial perturbations in neural network classifiers and reachability analysis of learning-enabled dynamical systems. In general, (1) is nonconvex and computationally intractable to solve exactly. Accordingly, our goal is to derive tractable upper bounds on its optimal value. To this end, we seek a provable pointwise majorizer $\bar{f}$ such that

$$f(x_c + \delta) \le \bar{f}(x_c, \delta), \qquad \forall \delta \text{ with } \|\delta\|_2 \le \varepsilon. \tag{2}$$

The majorizer $\bar{f}$ is constructed using local smoothness information at x_c, namely the function value $f(x_c)$, gradient $\nabla f(x_c)$, Hessian $\nabla^2 f(x_c)$, and their Lipschitz bounds. Given such a pointwise upper bound, we obtain the induced global certificate

$$f^*(x_c, \varepsilon) \coloneqq \sup_{\|\delta\|_2 \le \varepsilon} f(x_c + \delta) \le \sup_{\|\delta\|_2 \le \varepsilon} \bar{f}(x_c, \delta) =: \bar{f}^*(x_c, \varepsilon). \tag{3}$$

For practical verification, we require $\bar{f}$ to be amenable to optimization over $\mathcal{B}(x_c, \varepsilon)$, ideally yielding a closed-form or efficiently computable upper bound on $\bar{f}^*(x_c, \varepsilon)$.

Such upper bounds are useful in applications including neural network verification Singla and Feizi (2020); Fazlyab et al. (2023) and reachable-set overapproximation Entesari et al. (2023); Sharifi and Fazlyab (2024). In this work, we unify prior constructions of $\bar{f}$, which are typically based on zeroth-, first-, or second-order local models with bounded remainder terms, and strengthen them by incorporating Hessian Lipschitz continuity. This yields tighter certificates for smooth neural networks by controlling the cubic remainder of an end-to-end second-order Taylor model.

3. Hierarchy of End-to-End Taylor Models

In this section, we develop a hierarchy of upper bounds for (1) using local information of increasing order. We index the hierarchy by the highest-order derivative information used in the construction, rather than by the degree of the resulting majorizer.

3.1. Zeroth-Order Information

The simplest majorizer uses only the local function value and the Lipschitz constant L_f. In this case, Lipschitz continuity directly gives

$$f(x_c + \delta) \le \bar{f}_0(x_c, \delta) \coloneqq f(x_c) + L_f \|\delta\|_2.$$

Maximizing over the perturbation set $\{\delta : \|\delta\|_2 \le \varepsilon\}$ yields the corresponding worst-case certificate

$$\sup_{\|\delta\|_2 \le \varepsilon} f(x_c + \delta) \le \bar{f}_0^*(x_c, \varepsilon) \coloneqq f(x_c) + L_f \varepsilon.$$

This majorizer only preserves the function value at x_c.

3.2. First-Order Information

The zeroth-order bound can be improved by incorporating first-order local information, namely the gradient $\nabla f(x_c)$ and the gradient Lipschitz constant $L_{\nabla f}$. Using a first-order Taylor expansion with a quadratic remainder bound gives the pointwise estimate

$$f(x_c + \delta) \le \bar{f}_1(x_c, \delta) \coloneqq f(x_c) + \nabla f(x_c)^\top \delta + \frac{1}{2} L_{\nabla f} \|\delta\|_2^2.$$

This majorizer preserves the function value and gradient at x_c. Maximizing the upper bound over $\|\delta\|_2 \le \varepsilon$ yields the closed-form certificate

$$\sup_{\|\delta\|_2 \le \varepsilon} f(x_c + \delta) \le \bar{f}_1^*(x_c, \varepsilon) \coloneqq f(x_c) + \|\nabla f(x_c)\|_2 \, \varepsilon + \frac{1}{2} L_{\nabla f} \varepsilon^2.$$

See Entesari et al. (2024) for derivations and proofs. Compared to the zeroth-order bound, the first-order bound is tighter whenever

$$\varepsilon \leq \frac{2(L_f - \|\nabla f(x_c)\|_2)}{L_{\nabla f}}. \tag{4}$$

We note that the threshold is always nonnegative since $L_f \geq \sup_x \|\nabla f(x)\|_2$. This condition makes explicit when incorporating gradient information is beneficial: the improvement is most pronounced when the local slope, captured by $\|\nabla f(x_c)\|_2$, is significantly smaller than the global Lipschitz constant L_f of f, while the local curvature, as quantified by $L_{\nabla f}$, remains moderate.

3.3. Second-Order Information

To further tighten the upper bound on $f(x_c + \delta)$, we incorporate second-order local information, namely the Hessian $\nabla^2 f(x_c)$ together with the Hessian Lipschitz constant $L_{\nabla^2 f}$. This yields the pointwise upper bound

$$f(x_c + \delta) \leq \bar{f}_2(x_c, \delta) \coloneqq f(x_c) + \nabla f(x_c)^\top \delta + \frac{1}{2}\delta^\top \nabla^2 f(x_c)\delta + \frac{1}{6} L_{\nabla^2 f} \|\delta\|_2^3. \tag{5}$$

That is, $\bar{f}_2$ is an end-to-end second-order Taylor model augmented with a certified cubic remainder term. We provide a formal derivation of (5) in Appendix A.1. By capturing both local curvature and higher-order smoothness, this bound can yield substantially tighter certificates, particularly in regions where f is well approximated by a quadratic model.
In contrast to $\bar{f}_0$ and $\bar{f}_1$, the second-order majorizer generally does not admit a closed-form worst-case bound: $\bar{f}_2^*(x_c, \varepsilon) \coloneqq \sup_{\|\delta\|_2 \leq \varepsilon} \bar{f}_2(x_c, \delta)$. This optimization is generally intractable because it involves maximizing a nonconvex quadratic term together with a cubic remainder over a Euclidean ball. Nonetheless, as we show next, it is possible to derive informative and computationally tractable upper bounds on $\bar{f}_2^*$ by exploiting structure in the quadratic form or spectral properties of $\nabla^2 f(x_c)$.

Proposition 1 (Split and Bound) *Let $\bar{f}_2^{sb}(x_c, \varepsilon)$ be defined as*

$$\bar{f}_2^{sb}(x_c, \varepsilon) = f(x_c) + \|\nabla f(x_c)\|_2 \, \varepsilon + \frac{1}{2}\left(\lambda_{\max}(\nabla^2 f(x_c))\right)_+ \varepsilon^2 + \frac{1}{6} L_{\nabla^2 f} \varepsilon^3,$$

where $(a)_+ = \max(a, 0)$. Then $\bar{f}_2^*(x_c, \varepsilon) \leq \bar{f}_2^{sb}(x_c, \varepsilon)$.

We defer the proof to Appendix A.2. The closed-form certificate $\bar{f}_2^{\mathrm{sb}}(x_c, \varepsilon)$ provides a practical upper bound on the second-order worst-case value $\bar{f}_2^*(x_c, \varepsilon)$. Moreover, it allows us to characterize regimes, analogous to (4), in which the bound using second-order information is provably tighter than the corresponding first- and zeroth-order bounds. As expected, this improvement is most pronounced for small perturbation budgets, where the cubic remainder is dominated by lower-order terms. We make this statement precise in the next result and defer the proof to Appendix A.3.

Proposition 2 *The split-and-bound certificate $\bar{f}_2^{\mathrm{sb}}(x_c, \varepsilon)$ satisfies $\bar{f}_2^{\mathrm{sb}}(x_c, \varepsilon) \leq \bar{f}_1^*(x_c, \varepsilon)$ whenever*

$$\varepsilon \leq \frac{3}{L_{\nabla^2 f}} \left(L_{\nabla f} - \left(\lambda_{\max} \nabla^2 f(x_c)\right)_+ \right)$$

Note that the threshold in Proposition 2 is always nonnegative, since $L_{\nabla f} \geq \sup_x \lambda_{\max}(\nabla^2 f(x))$.

Moreover, once $\bar{f}_2^{\mathrm{sb}}(x_c, \varepsilon)$ has been computed, the corresponding first- and zeroth-order certificates, $\bar{f}_1^*(x_c, \varepsilon)$ and $\bar{f}_0^*(x_c, \varepsilon)$, can be obtained with essentially no additional overhead, since all required quantities are already available. This observation allows us to combine the hierarchy into a single certificate that is guaranteed to be at least as tight as each individual bound:

$$\bar{f}_m(x_c, \varepsilon) := \min\{\bar{f}_0^*(x_c, \varepsilon),\ \bar{f}_1^*(x_c, \varepsilon),\ \bar{f}_2^{\mathrm{sb}}(x_c, \varepsilon)\}.$$

By construction, $\bar{f}_m(x_c, \varepsilon)$ dominates all three certificates and therefore never performs worse than any of the individual methods.

Remark 3 Proposition 1 yields a practical closed-form upper bound on $\bar{f}_2^*(x_c, \varepsilon)$. In principle, one can obtain less conservative bounds by replacing the split-and-bound relaxation with tighter numerical procedures, which typically require solving a nonconvex optimization problem or a system of nonlinear equations. In our experiments, however, the resulting gains were modest relative to the additional computational burden. For this reason, we adopt the closed-form certificate here and leave the design of tighter numerical relaxations to future work.

To make the proposed bound operational, it remains to estimate the Hessian Lipschitz constant $L_{\nabla^2 f}$. We develop this construction in Section 4. Before turning to that problem, however, we derive the analogous upper bounds for the case in which the input region is an ℓ_∞ ball.

3.4. ℓ_∞-Bounded Inputs

As established in Entesari et al. (2024), one can derive ℓ_∞ counterparts of the upper bounds $\bar{f}_i$. Doing so directly, however, requires Lipschitz constants with respect to the ℓ_∞ norm. Since the methods developed later in Section 4 are primarily geared toward estimating ℓ_2 Lipschitz constants, we reuse the previously established pointwise upper bounds $\bar{f}_i$ and convert them into valid certificates over ℓ_∞ input balls via norm relations.

We consider generalized ℓ_∞-bounded perturbations and study the problem

$$\max_{\delta}\ f(x_c + \delta) \qquad \text{subject to} \qquad \|D^{-1}\delta\|_\infty \leq 1, \tag{6}$$

where D is a diagonal positive definite matrix. The feasible set is an axis-aligned box, and the standard ℓ_∞ perturbation model is recovered by taking $D = \varepsilon I$. Analogously to the ℓ_2-bounded case in (3), we define

$$f^*(x_c, D) := \sup_{\|D^{-1}\delta\|_\infty \leq 1} f(x_c + \delta) \leq \sup_{\|D^{-1}\delta\|_\infty \leq 1} \bar{f}(x_c, \delta) =: \bar{f}^*(x_c, D). \tag{7}$$

As established in Entesari et al. (2023), the zeroth-order certificate takes the form $\bar{f}_0^*(x_c, D) = f(x_c) + L_f\|D\|_F$, where $\|\cdot\|_F$ denotes the Frobenius norm. Moreover, Sharifi and Fazlyab (2024) shows that the first-order certificate is given by $\bar{f}_1^*(x_c, D) = f(x_c) + \|D\nabla f(x_c)\|_1 + \frac{L_{\nabla f}}{2}\|D\|_F^2$. We next derive a counterpart of the second-order certificate for generalized ℓ_∞ perturbations using the same split-and-bound idea. The proof is deferred to Appendix A.4.

Proposition 4 *Let $\bar{f}_2^{sb}(x_c, D)$ be defined as*

$$\bar{f}_2^{sb}(x_c, D) = f(x_c) + \|D\nabla f(x_c)\|_1 + \frac{1}{2}n\left(\lambda_{\max}\left(D\nabla^2 f(x_c)D\right)\right)_+ + \frac{L_{\nabla^2 f}}{6}\|D\|_F^3,$$

where $(a)_+ = \max(a, 0)$. Then $\bar{f}_2^*(x_c, D) \leq \bar{f}_2^{sb}(x_c, D)$.

4. Efficient Estimation of the Lipschitz Constant of the Hessian

To make the second-order certificate practical, it remains to compute a tractable upper bound on the Hessian Lipschitz constant $L_{\nabla^2 f}$. In this section, we develop such a bound for smooth fully connected neural networks. Specifically, we consider an L-layer twice-differentiable network $x \mapsto z^L(x)$ with input dimension N_0 and output dimension N_L, and consider scalar outputs of the form $f(x) = c^\top z^L(x)$. For an input $x \in \mathbb{R}^{N_0}$, let $z^I(x) \in \mathbb{R}^{N_I}$ and $a^I(x) \in \mathbb{R}^{N_I}$ denote the pre-activation and post-activation vectors at layer I, respectively, with $a^0(x) = x$. When no confusion arises, we omit the explicit dependence on x. The network evolves according to

$$z^I = W^I a^{I-1} + b^I, \qquad a^I = \sigma(z^I),$$

where $W^I \in \mathbb{R}^{N_I \times N_{I-1}}$ and $b^I \in \mathbb{R}^{N_I}$ are the weight matrix and bias vector of layer I, and σ is an elementwise twice-differentiable activation function. We assume that σ, σ', and σ'' are Lipschitz continuous, with Lipschitz constants L_σ, $L_{\sigma'}$, and $L_{\sigma''}$.

It has been established that for this architecture, z^L and ∇z^L are Lipschitz continuous and one can find certified Lipschitz constants L_{z^L} and $L_{\nabla z^L}$ for them, respectively, Entesari et al. (2024); Singla and Feizi (2020). Here, we establish an algorithm to acquire a bound on $L_{\nabla^2 z^L}$. To the best of our knowledge, this is the first work to establish a computationally tractable algorithm to provide upper bounds on the Lipschitz constant of the Hessian of a neural network.

To derive a tractable bound on the Hessian Lipschitz constant, we exploit the compositional structure of the network. In particular, we first establish a general composition rule for the Lipschitz constant of the Hessian, and then specialize it to feedforward neural networks. For each layer I, define $F^I : \mathbb{R}^{N_{I-1}} \to \mathbb{R}^{N_I}$ by $F^I(x) = \sigma(W^I x + b^I)$, and for the final layer let $F^L(x) = W^L x + b^L$. Then $a^I(x) = F^I \circ a^{I-1}(x)$. Let F^I_j denote the jth coordinate of F^I. By the chain rule, $D(F^I_j \circ a^{I-1})(x) = \nabla F^I_j(a^{I-1}(x))^\top Da^{I-1}(x)$, and

$$D^2(F^I_j \circ a^{I-1})(x) = Da^{I-1}(x)^\top \nabla^2 F^I_j(a^{I-1}(x)) Da^{I-1}(x) + \sum_{i=1}^{N_{I-1}} \frac{\partial F^I_j}{\partial x_i}\big(a^{I-1}(x)\big)\, \nabla^2 a^{I-1}_i(x),$$

where D denotes the differential operator. The following theorem establishes our main compositional bound; we defer the proof to Appendix A.5.

Theorem 5 *Define $L_{\nabla^2 a^I_j}$ as follows*

$$\begin{aligned} L_{\nabla^2 a^I_j} \coloneqq\ & L_{\nabla^2 F^I_j}\, L^3_{a^{I-1}} + 2\, L_{Da^{I-1}}\, L_{a^{I-1}}\, L_{\nabla F^I_j} \\ & + \sum_{i=1}^{N_{I-1}} \left(L_{\partial_i F^I_j}\, L_{a^{I-1}}\, L_{Da^{I-1}_i} + L_{\nabla^2 a^{I-1}_i} \sup_x \big|\partial_i F^I_j(x)\big| \right), \end{aligned}$$

where $\partial_i F^I_j(x) \coloneqq \frac{\partial F^I_j(x)}{\partial x_i}$. Then $L_{\nabla^2 a^I_j}$ is a Lipschitz constant of the Hessian of the map $x \mapsto a^I_j(x)$.

Theorem 5 yields a modular procedure for bounding the Hessian Lipschitz constant of each layer. Moreover, all quantities appearing in the theorem can be computed efficiently using existing tools. In particular, Lipschitz constants of the activations a^I can be bounded using Fazlyab et al. (2023, 2019), while Lipschitz constants of their gradients, namely L_{Da^I} and $L_{Da^I_i}$, can be obtained using

Entesari et al. (2024). Moreoever, for the layer map $F^I(x) = \sigma(W^I x + b^I)$, we have $DF^I(x) = \mathrm{Diag}(\sigma'(W^I x + b^I))W^I$. Consequently,

$$\sup_x \left| \frac{\partial F_j^I(x)}{\partial x_i} \right| = L_\sigma |W_{ji}^I| \tag{8}$$

$$L_{\partial_i F_j^I} = L_{\sigma'} \|W_{j,:}^I\|_2 |W_{ji}^I| \tag{9}$$

$$L_{\nabla F_j^I} = L_{\sigma'} \|W_{j,:}^I\|_2^2 \tag{10}$$

$$L_{\nabla^2 F_j^I} = L_{\sigma''} \|W_{j,:}^I\|_2^3 \tag{11}$$

where $\partial_i F_j^I := \frac{\partial F_j^I}{\partial x_i}$. The derivation of (8)–(11) is deferred to Appendix A.6. Together, these expressions allow the quantities $L_{\nabla^2 a_i^I}$ to be computed sequentially across layers, ultimately yielding bounds on $L_{\nabla^2 z_j^L}$.

To bound $L_{\nabla^2 f}$ for $f(x) = c^\top z^L(x)$, one could directly replace the final affine layer $F^L(x) = W^L x + b^L$ by $\hat{F}^L(x) = c^\top W^L x + c^\top b^L$ and apply the same recursion. However, a tighter bound can be obtained by combining the final two layers and instead considering

$$\hat{F}(x) = c^\top W^L \sigma(W^{L-1} x + b^{L-1}) + c^\top b^L.$$

Given all quantities up to layer $L-2$, we then apply the same compositional bound to $\hat{F}$. Defining $A := (W^{L-1})^\top \mathrm{Diag}(c^\top W^L)$, we obtain

$$\sup_x \left| \frac{\partial \hat{F}(x)}{\partial x_i} \right| = L_\sigma \sum_j |A_{ij}| \qquad L_{\partial_i \hat{F}} = L_{\sigma'} \sum_j |A_{ij}| \|W_{j,:}^{L-1}\|_2$$

and $L_{\nabla \hat{F}} = L_{\sigma'} \|A\|_2 \|W^{L-1}\|_2$. For $L_{\nabla^2 \hat{F}}$, we use Theorem 3.8 of Entesari et al. (2024), which yields a tighter estimate than the naive bound $L_{\nabla^2 \hat{F}} = L_{\sigma''} \|A\|_2 \|W^{L-1}\|_2 \max_i \|W_{i,:}^{L-1}\|_2$. The derivation of these bounds is deferred to Appendix A.7.

5. Integration of Components: The Final Algorithm

We now combine the ingredients developed in the previous sections into a practical algorithm for the generalized ℓ_∞-bounded problem (6). Given the Lipschitz constants computed as in Section 4, we first apply the upper-bounding framework of Section 2 to construct the certificate $\bar{f}_m(x_c, D)$.

When this bound is not sufficiently tight for the application at hand, we further embed the method within a branch-and-bound (BaB) framework. Specifically, we adopt the BaB procedure of Entesari et al. (2023), which iteratively partitions the input domain to reduce the relaxation gap until a prescribed tolerance is met. Our top-level procedure is summarized in Algorithm 1.

The remaining ingredient in the proposed algorithm is the computation of $\lambda_{\max}\big(D\nabla^2 f(x) D\big)$. Several approaches are available for this task. A straightforward option is to use modern automatic differentiation frameworks to form the Hessian $\nabla^2 f(x)$ explicitly, form the scaled matrix $D\nabla^2 f(x) D$, and then compute its largest eigenvalue using standard eigendecomposition routines. This approach is practical for smaller networks and moderate input dimensions.

When the Hessian is too expensive to compute explicitly, one may instead rely on Hessian–vector products, which are supported efficiently in modern automatic differentiation frameworks, together with iterative methods such as power iteration or Lanczos Van Loan and Golub (1996), to estimate $\lambda_{\max}\big(D\nabla^2 f(x) D\big)$ directly. In our experiments, the first approach yielded better runtime and was therefore used throughout.

Algorithm 1 Hierarchical End-to-End Taylor Bounds

Input: Desired point x_c, input box D_0, function $f(\cdot)$, gradient calculator $\nabla f(\cdot)$, vector-hessian calculator $h(x, v) := \nabla^2 f(x)v$, Stateful branching algorithm $\mathcal{B}$, tolerance ε_{tol}
Output: Upper Bound on $\max_{\|D_0^{-1}\delta\|_\infty \leq 1} f(x_c + \delta)$.
Initialize: $x \leftarrow x_c, D \leftarrow D_0$, Gap $\leftarrow \infty$, Initialize $\mathcal{B}(f, x_c, D_0)$
$(L_f, L_{a^I}) \leftarrow$ Fazlyab et al. (2023) (f)
$(L_{\nabla f}, L_{Da^I}, L_{Da_i^I}) \leftarrow$ Entesari et al. (2024) (f, L_{a^I})
$(L_{\nabla^2 f}, L_{\nabla^2 a_j^I}) \leftarrow$ Theorem 5 $(f, L_{a^I}, L_{Da^I}, L_{Da_i^I})$
while Gap $> \varepsilon_{\text{tol}}$ **do**
  $\bar{f}_0^* \leftarrow f(x) + L_f \|D\|_F$ and $\bar{f}_1^* \leftarrow f(x) + \|D\nabla f(x)\|_1 + \frac{L_{\nabla f}}{2}\|D\|_F^2$
  $\lambda \leftarrow \lambda_{\max}(D\nabla^2 f(x)D)$
  $\bar{f}_2^{\text{sb}} \leftarrow f(x) + \|D\nabla f(x)\|_1 + \frac{1}{2}n\,(\lambda)_+ + \frac{L_{\nabla^2 f}}{6}\|D\|_F^3$
  $\bar{f}_m \leftarrow \min\{\bar{f}_0^*, \bar{f}_1^*, \bar{f}_2^{\text{sb}}\}$
  $(x, D, Gap) \leftarrow \mathcal{B}(x, D, \bar{f}_m)$. {Given the current upper bound, the branching algorithm will provide the problem that needs to be solved at the next step.}
end while
Return: $\bar{f}_m$

6. Experiments

In this section, we present numerical experiments on reachability analysis under ℓ_∞-bounded state perturbations for LTI systems controlled by neural networks. Specifically, we consider the optimization problem

$$\max_x \quad c^\top \big(Ax + Bz^L(x) + d\big) \quad \text{s.t.} \quad \|D(x - x_c)\|_\infty \leq 1, \tag{12}$$

where $A \in \mathbb{R}^{n_0 \times n_0}$, $B \in \mathbb{R}^{n_0 \times n_u}$, and $d \in \mathbb{R}^{n_0}$ define the system dynamics, and n_u denotes the number of control inputs. The neural network controller z^L is trained to imitate a model predictive controller (MPC).

Our goal is to bound the set of reachable states of the closed-loop system over multiple time steps and verify that the system can both avoid unsafe regions and reach its target state. By solving (12) over a suitable set of directions c, we obtain a polyhedral outer approximation of the reachable set at the next time step. This approximation can then be propagated recursively for multi-step reachability analysis. Different choices of directions yield different geometric templates; in this work, we use rotated rectangles, following Entesari et al. (2023).

We consider the standard 6D quadrotor benchmark. For a discretization step Δt, the system dynamics are given by

$$A = I_{6\times 6} + \Delta t \times \begin{bmatrix} 0_{3\times 3} & I_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} \end{bmatrix}, \quad B = \Delta t \times \begin{bmatrix} & g & 0 & 0 \\ 0_{3\times 3} & 0 & -g & 0 \\ & 0 & 0 & 1 \end{bmatrix}^\top, \quad d = \Delta t \times \begin{bmatrix} 0_{5\times 1} \\ -g \end{bmatrix}$$

The control input is $u = \begin{bmatrix}\tan(\theta) & \tan(\phi) & \tau\end{bmatrix}^\top$, which in our setting is produced by the neural network controller z^L. Following Sharifi and Fazlyab (2024), training data is collected from an MPC controller that steers the system toward the origin while avoiding two spherical obstacles; see Figure 1. A neural network with architecture $6 \times 32 \times 32 \times 3$ and $\tanh$ activations is then trained to imitate the MPC controller. We consider a hyperrectangular initial set and compute its reachable sets over 10 time steps, corresponding to a horizon of 1.0 seconds.

We compare our method against the closest baseline, namely the first-order local bounding approach of Sharifi and Fazlyab (2024). For both methods, the Lipschitz constants L_{a^I} are computed using LipLT Fazlyab et al. (2023). The results are reported in Figure 1 and Table 1.

As shown in Table 1, the bound using second-order information substantially reduces the number of branches generated by the BaB procedure at both tolerance levels. This directly reflects the tighter local approximation provided by the second-order bound, especially on smaller subdomains where higher-order information becomes most effective.

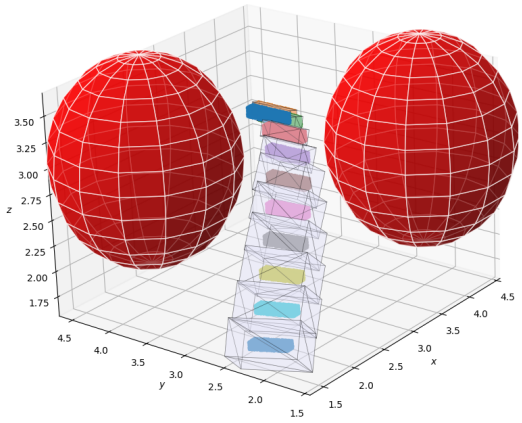

Figure 1: The quadrotor problem setup. The point clouds show trajectory samples from the system via numerical simulation. The obstacles are shown as two spheres. The reachable sets are calculated with a tolerance of 0.001.

BaB Tol	Bounding Method	Branches	Run Time (s)
10^{-2}	1st order (local)	908.0k $\pm$ 36.5k	320.4 $\pm$ 13.4
	HITAB (Ours)	119.7k $\pm$ 9.3k	108.7 $\pm$ 11.5
10^{-3}	1st order (local)	1613k $\pm$ 113.8k	555.0 $\pm$ 44.6
	HITAB (Ours)	434.8k $\pm$ 26.4k	531.4 $\pm$ 37.1

Table 1: Run times and number of branches for the BaB algorithm using various bounding algorithms and tolerances. We report statistics of mean $\pm$ standard deviation over 10 runs.

7. Conclusion

In this work, we introduced HITAB, a practical framework for verification of smooth neural networks that systematically exploits Hessian Lipschitz continuity. Our approach addresses a key limitation of existing verification methods by incorporating second-order smoothness information, leading to tighter certificates for smooth architectures. We developed a hierarchy of bounds based on zeroth-, first-, and second-order information, together with explicit conditions under which higher-order constructions provably improve upon lower-order ones. Our main technical contribution is a compositional method (Theorem 5) for efficiently upper-bounding $L_{\nabla^2 f}$ in deep neural networks through layerwise propagation of curvature information. We further extended the second-order framework to ℓ_∞-bounded perturbations (Proposition 4) and integrated it into a branch-and-bound pipeline through Algorithm 1.

Limitations and Future Work. Our experiments focused on ℓ_∞-bounded problems. In the ℓ_2 setting, tighter estimates of $\bar{f}_2^*(x_c, \varepsilon)$ may be achievable, albeit at greater computational cost, for example by solving the associated optimality conditions numerically. Exploring such refinements, together with a broader empirical study of second-order bounds under different input geometries, is an important direction for future work. In addition, our current implementation is CPU-based and does not exploit GPU parallelism. Adapting the major components of the algorithm to GPU architectures could substantially improve runtime and scalability.

References

Alexandre Araujo, Aaron Havens, Blaise Delattre, Alexandre Allauzen, and Bin Hu. A unified algebraic perspective on lipschitz neural networks. *arXiv preprint arXiv:2303.03169*, 2023.

Taha Entesari, Sina Sharifi, and Mahyar Fazlyab. Reachlipbnb: A branch-and-bound method for reachability analysis of neural autonomous systems using lipschitz bounds. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1003–1010. IEEE, 2023.

Taha Entesari, Sina Sharifi, and Mahyar Fazlyab. Compositional curvature bounds for deep neural networks. *arXiv preprint arXiv:2406.05119*, 2024.

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, Taha Entesari, Aniket Roy, and Rama Chellappa. Certified robustness via dynamic margin maximization and improved lipschitz regularization. *Advances in Neural Information Processing Systems*, 36:34451–34464, 2023.

Haimin Hu, Mahyar Fazlyab, Manfred Morari, and George J Pappas. Reach-sdp: Reachability analysis of closed-loop systems with neural network controllers via semidefinite programming. In *2020 59th IEEE conference on decision and control (CDC)*, pages 5929–5934. IEEE, 2020.

Chao Huang, Jiameng Fan, Wenchao Li, Xin Chen, and Qi Zhu. Reachnn: Reachability analysis of neural-network controlled systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 18(5s):1–22, 2019.

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

Ahmed Dawod Mohammed Ibrahum, Manzoor Hussain, and Jang-Eui Hong. Deep learning adversarial attacks and defenses in autonomous vehicles: A systematic literature review from a safety perspective. *Artificial Intelligence Review*, 58(1):28, 2024.

Radoslav Ivanov, James Weimer, Rajeev Alur, George J Pappas, and Insup Lee. Verisig: verifying safety properties of hybrid systems with neural network controllers. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, pages 169–178, 2019.

Radoslav Ivanov, Taylor Carpenter, James Weimer, Rajeev Alur, George Pappas, and Insup Lee. Verisig 2.0: Verification of neural network controllers using taylor model preconditioning. In *International Conference on Computer Aided Verification*, pages 249–262. Springer, 2021.

Haseeb Javed, Shaker El-Sappagh, and Tamer Abuhmed. Robustness in deep learning models for medical diagnostics: security and adversarial challenges towards robust ai applications. *Artificial Intelligence Review*, 58(1):12, 2024.

Matt Jordan and Alexandros G Dimakis. Exactly computing the local lipschitz constant of relu networks. *Advances in Neural Information Processing Systems*, 33:7344–7353, 2020.

Niklas Kochdumper, Christian Schilling, Matthias Althoff, and Stanley Bak. Open-and closed-loop neural network verification using polynomial zonotopes. In *NASA Formal Methods Symposium*, pages 16–36. Springer, 2023.

Patricia Pauli, Dennis Gramlich, and Frank Allgöwer. Lipschitz constant estimation for 1d convolutional neural networks. In *Learning for Dynamics and Control Conference*, pages 1321–1332. PMLR, 2023.

Patricia Pauli, Aaron Havens, Alexandre Araujo, Siddharth Garg, Farshad Khorrami, Frank Allgöwer, and Bin Hu. Novel quadratic constraints for extending lipsdp beyond slope-restricted activations. *arXiv preprint arXiv:2401.14033*, 2024.

Sina Sharifi and Mahyar Fazlyab. Provable bounds on the hessian of neural networks: Derivative-preserving reachability analysis. *arXiv preprint arXiv:2406.04476*, 2024.

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

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*, pages 315–335. Springer, 2025.

Sahil Singla and Soheil Feizi. Second-order provable defenses against adversarial attacks. In *International conference on machine learning*, pages 8981–8991. PMLR, 2020.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Florian Tambon, Gabriel Laberge, Le An, Amin Nikanjam, Paulina Stevia Nouwou Mindom, Yann Pequignot, Foutse Khomh, Giulio Antoniol, Ettore Merlo, and Francois Laviolette. How to certify machine learning based safety-critical systems? a systematic literature review. *Automated Software Engineering*, 29(2):38, 2022.

Charles F Van Loan and G Golub. Matrix computations (johns hopkins studies in mathematical sciences). *Matrix Computations*, 5:32, 1996.

Ruigang Wang and Ian Manchester. Direct parameterization of lipschitz-bounded deep networks. In *International Conference on Machine Learning*, pages 36093–36110. PMLR, 2023.

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.

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. *arXiv preprint arXiv:2011.13824*, 2020.

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

Appendix A. Proofs

A.1. Proof of equation 5

Proof To prove the desired bound, we utilize the mean-value theorem. Define the scalar function $\phi(t) = f(x + t\delta)$. We have that

$$\phi'(t) = \nabla f(x + t\delta)^\top \delta, \quad \phi''(t) = \delta^\top \nabla^2 f(x + t\delta)\delta.$$

The mean-value theorem guarantees

$$\begin{aligned}
\phi(1) &= \phi(0) + \int_0^1 \phi'(t)dt \\
&= \phi(0) + \phi'(0) + \int_0^1 \left(\phi'(t) - \phi'(0)\right) dt \\
&= \phi(0) + \phi'(0) + \int_0^1 \left(\int_0^t \phi''(s)ds\right) dt \\
&= \phi(0) + \phi'(0) + \int_0^1 \left(\int_0^t \left(\phi''(s) \pm \phi''(0)\right) ds\right) dt \\
&= \phi(0) + \phi'(0) + \frac{1}{2}\phi''(0) + \int_0^1 \left(\int_0^t \left(\phi''(s) - \phi''(0)\right) ds\right) dt.
\end{aligned}$$

Next, we have

$$\begin{aligned}
\left|\phi(1) - \phi(0) - \phi'(0) - \frac{1}{2}\phi''(0)\right| &= \left|\int_0^1 \left(\int_0^t \left(\phi''(s) - \phi''(0)\right) ds\right) dt\right| \\
&\leq \int_0^1 \int_0^t \left|\phi''(s) - \phi''(0)\right| dsdt \\
&\leq \int_0^1 \int_0^t L_{\phi''} sdsdt \\
&= \frac{1}{6} L_{\phi''}.
\end{aligned}$$

Finally, note that $L_{\phi''} \leq L_{\nabla^2 f} \|\delta\|_2^3$. Putting all these together, we arrive at the desired bound. ■

A.2. Proof of Proposition 1

Proof Expanding the sup operator over the individual terms, we have

$$\begin{aligned}
\bar{f}_2^*(x_c, \varepsilon) &\leq \sup_{\|\delta\|_2 \leq \varepsilon} f(x_c) + \sup_{\|\delta\|_2 \leq \varepsilon} \nabla f(x_c)^\top \delta + \sup_{\|\delta\|_2 \leq \varepsilon} \frac{1}{2}\delta^\top f(x_c)\delta + \sup_{\|\delta\|_2 \leq \varepsilon} \frac{1}{6} L_{\nabla^2 f} \|\delta\|_2^3 \\
&= f(x_c) + \|\nabla f(x_c)\|_2 \varepsilon + \frac{1}{2}\left(\lambda_{\max} \nabla^2 f(x_c)\right)_+ \varepsilon^2 + \frac{1}{6} L_{\nabla^2 f} \varepsilon^3,
\end{aligned}$$

where the $\left(\lambda_{\max} \nabla^2 f(x_c)\right)_+$ shows up because if $\lambda_{\max} \nabla^2 f(x_c) < 0$, then the corresponding sup would choose $\delta = 0$ to maximize its objective. ■

A.3. Proof of Proposition 2

Proof We simply solve for when the bound provided by the second-order method is smaller than the one provided by the first-order method. We have:

$$f(x_c) + \|\nabla f(x_c)\|_2 \varepsilon + \frac{1}{2}\left(\lambda_{\max}\nabla^2 f(x_c)\right)_+ \varepsilon^2 + \frac{1}{6}L_{\nabla^2 f}\varepsilon^3 \leq f(x_c) + \|\nabla f(x_c)\|_2 \varepsilon + \frac{1}{2}L_{\nabla f}\varepsilon^2,$$

$$\frac{1}{2}\left(\lambda_{\max}\nabla^2 f(x_c)\right)_+ + \frac{1}{6}L_{\nabla^2 f}\varepsilon \leq \frac{1}{2}L_{\nabla f},$$

Which yields the desired result. ■

A.4. Proof of Proposition 4

Proof By definition, we have that

$$\bar{f}_2^*(x_c, D) = \max_{\|D^{-1}\delta\|_\infty \leq 1} f(x_c) + \nabla f(x_c)^\top \delta + \frac{1}{2}\delta^\top \nabla^2 f(x_c)\delta + \frac{1}{6}L_{\nabla^2 f}\|\delta\|_2^3$$

We upper bound the right-hand side by taking the maximization individually over each term. We have

$$\max_{\|D^{-1}\delta\|_\infty \leq 1} \nabla f(x_c)^\top \delta = \max_{\|\nu\|_\infty \leq 1} \nabla f(x_c)^\top D\nu = \|D\nabla f(x_c)\|_1,$$

where we have used the definition of the dual norm. Moreover, we have

$$\begin{aligned}
\max_{\|D^{-1}\delta\|_\infty \leq 1} \delta^\top \nabla^2 f(x_c)\delta &= \max_{\|\nu\|_\infty \leq 1} \nu^\top D\nabla^2 f(x_c)D\nu \\
&\leq \max_{\|\nu\|_2 \leq \sqrt{n}} \nu^\top D\nabla^2 f(x_c)D\nu \\
&= n \max_{\|\nu\|_2 \leq 1} \nu^\top D\nabla^2 f(x_c)D\nu \\
&= n\left(\lambda_{\max}\left(D\nabla^2 f(x_c)D\right)\right)_+ \\
&\leq n d_{\max}^2 \left(\lambda_{\max}\nabla^2 f(x_c)\right)_+,
\end{aligned}$$

where $d_{\max}$ is the largest diagonal element of D. Finally, we have

$$\max_{\|D^{-1}\delta\|_\infty \leq 1} \|\delta\|_2^3 = \max_{\|\delta\|_\infty \leq 1} \|D\delta\|_2^3 = \|D\|_F^3,$$

where the last equality holds as D is a diagonal matrix. This concludes the proof.

■

A.5. Proof of Theorem 5

Proof Writing the definition of Lipschitz continuity, we have

$$\begin{aligned}
&\|D^2F_j^I \circ a^{I-1}(x) - D^2F_j^I \circ a^{I-1}(y)\|_2 \\
&\quad\leq \underbrace{\left\|Da^{I-1}(x)^\top \nabla^2 F_j^I\big(a^{I-1}(x)\big)Da^{I-1}(x) - Da^{I-1}(y)^\top \nabla^2 F_j^I\big(a^{I-1}(y)\big)Da^{I-1}(y)\right\|_2}_{I} \\
&\qquad + \underbrace{\sum_{i=1}^{N_{I-1}} \left\|\frac{\partial F_j^I(x)}{\partial x_i}\Big|_{a^{I-1}(x)} \nabla^2 a_i^{I-1}(x) - \frac{\partial F_j^I(x)}{\partial x_i}\Big|_{a^{I-1}(y)} \nabla^2 a_i^{I-1}(y)\right\|_2}_{II}.
\end{aligned}$$

We can further bound each term. We have

$$\begin{aligned}
I &= \Big\|Da^{I-1}(x)^\top \nabla^2 F_j^I\big(a^{I-1}(x)\big)Da^{I-1}(x) \\
&\qquad \pm Da^{I-1}(x)^\top \nabla^2 F_j^I\big(a^{I-1}(y)\big)Da^{I-1}(y) \\
&\qquad - Da^{I-1}(y)^\top \nabla^2 F_j^I\big(a^{I-1}(y)\big)Da^{I-1}(y)\Big\|_2 \\
&\leq \left\|Da^{I-1}(x)^\top\right\|_2 \cdot \Big\|\nabla^2 F_j^I\big(a^{I-1}(x)\big)Da^{I-1}(x) \\
&\qquad\qquad \pm \nabla^2 F_j^I\big(a^{I-1}(y)\big)Da^{I-1}(x) \\
&\qquad\qquad - \nabla^2 F_j^I\big(a^{I-1}(y)\big)Da^{I-1}(y)\Big\|_2 \\
&\qquad + \left\|\big(Da^{I-1}(x) - Da^{I-1}(y)\big)^\top\right\|_2 \cdot \left\|\nabla^2 F_j^I\big(a^{I-1}(y)\big)\right\|_2 \cdot \left\|Da^{I-1}(y)\right\|_2.
\end{aligned}$$

We use the fact that over matrices we have $\|A\|_2 = \left\|A^\top\right\|_2$ to further simplify. We have

$$\begin{aligned}
I &\leq \left\|\nabla^2 F_j^I\big(a^{I-1}(x)\big) - \nabla^2 F_j^I\big(a^{I-1}(y)\big)\right\|_2 \cdot \left\|Da^{I-1}(x)\right\|_2^2 \\
&\qquad + \left\|Da^{I-1}(x)\right\|_2 \cdot \left\|\nabla^2 F_j^I\big(a^{I-1}(y)\big)\right\|_2 \cdot \left\|Da^{I-1}(x) - Da^{I-1}(y)\right\|_2 \\
&\qquad + \left\|Da^{I-1}(x) - Da^{I-1}(y)\right\|_2 \cdot \left\|\nabla^2 F_j^I\big(a^{I-1}(y)\big)\right\|_2 \cdot \left\|Da^{I-1}(y)\right\|_2 \\
&\leq \Big(L_{\nabla^2 F_j^I} L_{a^{I-1}} \left\|Da^{I-1}(x)\right\|_2^2 \\
&\qquad + L_{Da^{I-1}} \left\|Da^{I-1}(x)\right\|_2 \cdot \left\|\nabla^2 F_j^I\big(a^{I-1}(y)\big)\right\|_2 \\
&\qquad + L_{Da^{I-1}} \left\|Da^{I-1}(y)\right\|_2 \cdot \left\|\nabla^2 F_j^I\big(a^{I-1}(y)\big)\right\|_2\Big) \|x-y\|_2 \\
&\leq \left(L_{\nabla^2 F_j^I} L_{a^{I-1}}^3 + 2L_{Da^{I-1}} L_{a^{I-1}} L_{\nabla F_j^I}\right) \|x-y\|_2.
\end{aligned}$$

Furthermore

$$
\begin{aligned}
II = \sum_{i=1}^{N_{I-1}} \Bigg\| & \left.\frac{\partial F_j^I(x)}{\partial x_i}\right|_{a^{I-1}(x)} \nabla^2 a_i^{I-1}(x) \\
& \pm \left.\frac{\partial F_j^I(x)}{\partial x_i}\right|_{a^{I-1}(y)} \nabla^2 a_i^{I-1}(x) - \left.\frac{\partial F_j^I(x)}{\partial x_i}\right|_{a^{I-1}(y)} \nabla^2 a_i^{I-1}(y) \Bigg\|_2 \\
\leq \sum_{i=1}^{N_{I-1}} \Bigg(& \left| \left.\frac{\partial F_j^I(x)}{\partial x_i}\right|_{a^{I-1}(x)} - \left.\frac{\partial F_j^I(x)}{\partial x_i}\right|_{a^{I-1}(y)} \right| \cdot \left\| \nabla^2 a_i^{I-1}(x) \right\|_2 \\
& + \left| \left.\frac{\partial F_j^I(x)}{\partial x_i}\right|_{a^{I-1}(y)} \right| \cdot \left\| \nabla^2 a_i^{I-1}(x) - \nabla^2 a_i^{I-1}(y) \right\|_2 \Bigg) \\
\leq \sum_{i=1}^{N_{I-1}} & \left(L_{\frac{\partial F_j^I(x)}{\partial x_i}} L_{a^{I-1}} L_{\nabla a_i^{I-1}} + L_{\nabla^2 a_i^{I-1}} \sup_x \left| \left.\frac{\partial F_j^I(x)}{\partial x_i}\right|_x \right| \right) \|x - y\|_2.
\end{aligned}
$$

Putting cases I and II together we get

$$
\begin{aligned}
\|D^2 F_j^I \circ a^{I-1}(x) - D^2 F_j^I \circ a^{I-1}(y)\|_2 \leq & \\
\Bigg(& L_{\nabla^2 F_j^I} L_{a^{I-1}}^3 + 2 L_{Da^{I-1}} L_{a^{I-1}} L_{\nabla F_j^I} \\
& + \sum_{i=1}^{N_{I-1}} \left(L_{\frac{\partial F_j^I(x)}{\partial x_i}} L_{a^{I-1}} L_{\nabla a_i^{I-1}} + L_{\nabla^2 a_i^{I-1}} \sup_x \left| \left.\frac{\partial F_j^I(x)}{\partial x_i}\right|_x \right| \right) \Bigg) \|x - y\|_2
\end{aligned}
$$

■

A.6. Derivation of equations 8, 9, 10, and 11

As stated, we have

$$DF^I(x) = \text{Diag}(\sigma'(W^I x + b^I)) W^I.$$

Subsequently, we can derive each of the desired terms. First, we have that

$$\frac{\partial F_j^I(x)}{\partial x_i} = W_{ji}^I \sigma'(W_{j,:}^I x + b_j^I).$$

Consequently, as $\sup_x \sigma'(x) \leq L_\sigma$, we have that

$$\sup \frac{\partial F_j^I(x)}{\partial x_i} = L_\sigma |W_{ji}^I|.$$

Moreover, using the Lipschitz property of σ' we have that

$$L_{\frac{\partial F_j^I(x)}{\partial x_i}} = L_{\sigma'} \left\| W_{j,:}^I \right\|_2 |W_{ji}^I|.$$

To derive $L_{\nabla F_j^I}$, we simply write the definition of Lipschitz continuity for ∇F_j^I. We have

$$\begin{aligned}\left\|\nabla F_j^I(x) - \nabla F_j^I(y)\right\|_2 &= \left\|\left(\sigma'(W_{j,:}^I x + b_j^I) - \sigma'(W_{j,:}^I y + b_j^I)\right) {W_{j,:}^I}^\top\right\|_2 \\ &\leq \left\|\sigma'(W_{j,:}^I x + b_j^I) - \sigma'(W_{j,:}^I y + b_j^I)\right\|_2 \cdot \left\|W_{j,:}^I\right\|_2 \\ &\leq L_{\sigma'} \left\|W_{j,:}^I\right\|_2^2.\end{aligned}$$

This result is also given if we were to use Theorem 3.9 of Entesari et al. (2024). Going one step further, we have

$$\nabla^2 F_j^I = \sigma''(W_{j,:}^I x + b_j^I) {W_{j,:}^I}^\top W_{j,:}^I.$$

As $\left\|{W_{j,:}^I}^\top W_{j,:}^I\right\|_2 = \left\|W_{j,:}^I\right\|_2^2$, this yields the desired result for $L_{\nabla^2 F_j^I}$

A.7. Derivation of Lipschitz terms for $\hat{F}$

$\hat{F} = c^\top W^L \sigma(W^{L-1}x + b^{L-1}) + c^\top b^L$ The basis for the proofs of this section are Entesari et al. (2024) and we only establish the connection. We have

$$\nabla \hat{F}(x) = {W^{L-1}}^\top \text{Diag}(\sigma'(W^{L-1}x + b^{L-1})) {W^L}^\top c$$

We can rewrite this as

$$\nabla \hat{F}(x) = {W^{L-1}}^\top \text{Diag}({W^L}^\top c)\sigma'(W^{L-1}x + b^{L-1}).$$

This simply follows from the exchange of terms between a diagonal matrix and multiplication by a vector. As instructed in the text, we define $A = {W^{L-1}}^\top \text{Diag}({W^L}^\top c)$. Now we have $\nabla \hat{F}(x) = A\sigma'(W^{L-1}x + b^{L-1})$. From this, we first obtain that

$$\sup \frac{\partial \hat{F}(x)}{\partial x_i} = L_\sigma \sum_j |A_{ij}|,$$

and then

$$L_{\frac{\partial \hat{F}(x)}{\partial x_i}} = L_{\sigma'} \sum_j |A_{ij}| \left\|W_{j,:}^{L-1}\right\|_2.$$

The naive Lipschitz bound establishes that

$$L_{\nabla \hat{F}} = L_{\sigma'} \|A\|_2 \left\|W^{L-1}\right\|_2.$$

Finally, the rewritten $\nabla \hat{F}(x) = A\sigma'(W^{L-1}x + b^{L-1})$ is in the form of a standard neural network layer as described in Entesari et al. (2024). As such we can utilize Theorem 3.8 therein to acquire $L_{\nabla^2 \hat{F}}$, by simply noting that $L_{D\nabla \hat{F}} = L_{\nabla^2 \hat{F}}$.