FLIP: Fast and Accurate Global Lipschitz Estimation for Large Feedforward Networks

Hongbo Chen
Independent Researcher
Wenchang, Hainan 571300, China
hongboc616@gmail.com

Zihao Ren
The College of Control Science and Engineering
Zhejiang University
zhren2000@zju.edu.cn

Abstract

The Lipschitz constant of neural networks is widely used in robustness, control, generalization bounds, and training stability. As modern neural networks continue to scale, existing methods for Lipschitz estimation are either very time-consuming or overly loose. In this work, we focus on both fast and accurate Lipschitz estimation for large feedforward neural networks. We first reformulate the well-known semidefinite program (SDP) framework into an explicit recursive form, and introduce a key theoretical result, bound backpropagation: the SDP objective has upper and lower bounds that can be propagated backward along the recursive computation graph and depend only on partial variables. Leveraging this property, we decompose the original SDP into a sequence of small subproblems, each minimizing a combination of upper and lower bounds over partial variables. We prove that each subproblem is one-dimensional and convex, enabling fast solution via the secant method. Finally, we present the full FLIP algorithm and its complexity, yielding fast and tight Lipschitz estimation for large feedforward networks. Experiments on both randomly generated and trained networks show that FLIP improves speed and tightness by several orders of magnitude over prior methods, giving the tightest Lipschitz estimates for 100M-parameter networks in seconds. Our code is available at: https://github.com/hongboc616/FLIP.

1 Introduction

The global Lipschitz constant of a neural network quantifies its maximum output effect to input perturbations. It has been widely used in many areas, such as providing robustness certificates [23, 9]; ensuring the stability of neural network controllers in deep reinforcement learning [2, 31, 5]; deriving generalization bounds and distribution shift theory [4, 21, 7]; and bounding gradient norms to improve training stability [1, 18, 12]. Since exact estimation of the global Lipschitz constant has been shown to be NP-hard [26], existing studies focus on estimating tight upper bounds, including SeqLip based on heuristic optimization [26] and the semidefinite program (SDP) framework LipSDP [10].

Large neural networks are now widely used, yet estimating tight Lipschitz upper bounds for them remains challenging. On the one hand, methods such as LipSDP have high time complexity [30], making Lipschitz estimation extremely time-consuming as the number of parameters grows. On the other hand, faster methods, such as the product of layerwise spectral norms (the naive upper bound) or ECLipsE-fast [30], often yield loose bounds because large neural networks are highly complex. These two challenges make it difficult for researchers to estimate and apply Lipschitz constants for large neural networks in broad applications.

In this paper, we focus on fast and tight global Lipschitz estimation for large feedforward neural networks. We first apply the Schur complement to the diagonal blocks of the large matrix in the SDP framework, reformulating it into an explicit recursive form. We then propose a key theoretical result

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

on this form, bound backpropagation: the SDP objective has upper and lower bounds determined by partial variables, and the bounds can be propagated backward layer by layer along the recursive computation graph. These upper and lower bounds respectively characterize sufficient and necessary conditions for minimizing the SDP objective. Leveraging this result, we decompose the original SDP into a sequence of small subproblems, and each subproblem optimizes a combination of upper and lower bounds over partial variables. We prove that each subproblem is a simple one-dimensional convex optimization, which can be efficiently solved by the secant method. Finally, we propose FLIP, a fast Lipschitz estimation algorithm, whose time complexity approaches the naive upper bound and is much lower than the original SDP, while producing Lipschitz estimates even tighter than prior methods. We compare FLIP with prior methods on both randomly generated and SVHN-trained neural networks, at medium and large scales. FLIP achieves the tightest Lipschitz bounds while improving speed by several orders of magnitude, computing the Lipschitz estimates for 100M-parameter neural networks in just a few seconds. In summary, our contributions are as follows:

- We convert the SDP framework into an explicit recursive form, and then prove an interesting result: the SDP objective has a sequence of upper and lower bounds, which can be propagated backward along the recursive computation graph and depend only on partial variables.
- By this theory, we decompose SDP into many small subproblems, prove each subproblem convex, and solve it efficiently via the secant method. We further propose the FLIP algorithm with low time complexity, enabling fast and accurate Lipschitz estimation for large feedforward networks.
- We evaluate FLIP on both randomly generated and trained large neural networks, showing that FLIP improves speed and tightness over other methods by several orders of magnitude and estimates the tightest Lipschitz bound for 100M-parameter neural networks in seconds.

2 Preliminaries

2.1 Problem Statement

Consider an l-hidden-layer feedforward neural network $f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_{l+1}}$ defined as:

$$\mathbf{x}_0 = \mathbf{x}, \quad \mathbf{x}_{i+1} = \sigma(W_i\mathbf{x}_i + b_i) \text{ for } i = 0, \ldots, l-1, \quad f(\mathbf{x}) = W_l\mathbf{x}_l + b_l, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{n_0}$ denotes the input, $W_i \in \mathbb{R}^{n_{i+1} \times n_i}$ and $b_i \in \mathbb{R}^{n_{i+1}}$ are the weight matrix and bias vector of the i-th hidden layer. $f(\mathbf{x}) = W_l\mathbf{x}_l + b_l$ denotes the output layer, and $\sigma : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i}$ is the element-wise activation function. The global Lipschitz constant of f is defined as the smallest $L \geq 0$ such that $\|f(\mathbf{x}) - f(\mathbf{y})\|_2 \leq L\|\mathbf{x} - \mathbf{y}\|_2$ for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{n_0}$.

Computing the exact Lipschitz constant of a neural network is NP-hard [26]. In line with many existing studies [26, 10, 30, 27], our goal is to estimate a tight upper bound on the Lipschitz constant.

The Naive Upper Bound The simplest Lipschitz upper bound is the product of the spectral norms of all weight matrices, $L_{\text{naive}} = \prod_i \|W_i\|_2$. Since computing the spectral norm of an n-dimensional square matrix costs $O(n^3)$, for a neural network with l hidden layers, width n, and output dimension d, the bound has time complexity $O(ln^3 + d^3)$. In terms of tightness, since this method ignores inter-layer interactions, the resulting bound is often very loose.

2.2 SDP Framework

A well-known framework for bounding Lipschitz constant is LipSDP [10], whose expanded form is:

Theorem 1 (LipSDP)**.** *Consider a network described by* (1)*. Suppose the activation function satisfies* $0 \leq \frac{\sigma(x)-\sigma(y)}{x-y} \leq 1, \forall x, y \in \mathbb{R}$*, if there exist positive diagonal matrices* $\{T_i\}_{i=1}^{l}$ *and* $L \geq 0$ *satisfying*

$$M(L, \{T_i\}) := \begin{bmatrix} L^2 I & -W_0^\top T_1 & 0 & \cdots & 0 & 0 & 0 \\ -T_1 W_0 & 2T_1 & -W_1^\top T_2 & \cdots & 0 & 0 & 0 \\ 0 & -T_2 W_1 & 2T_2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2T_{l-2} & -W_{l-2}^\top T_{l-1} & 0 \\ 0 & 0 & 0 & \cdots & -T_{l-1}W_{l-2} & 2T_{l-1} & -W_{l-1}^\top T_l \\ 0 & 0 & 0 & \cdots & 0 & -T_l W_{l-1} & 2T_l - W_l^\top W_l \end{bmatrix} \succeq 0,$$

then the Lipschitz constant of (1) *has an upper bound* L*. The tightest upper bound can be obtained by solving the following semidefinite program (SDP):*

$$\min_{L\geq 0,\ \{T_i\}} L^2 \qquad \text{s.t. } M(L,\{T_i\}) \succeq 0. \tag{2}$$

Remark The above SDP problem is commonly solved by interior-point methods. For a neural network with l hidden layers and width n, the complexity of each interior-point iteration is $O(l^4n^4)$ [30]. If the method takes K_{SDP} iterations, the overall time complexity is $O(K_{\text{SDP}}l^4n^4)$, which grows rapidly with both network depth and width. Consequently, directly solving this SDP becomes computationally expensive for modern large-scale neural networks. This motivates the development of more efficient algorithms for Lipschitz estimation.

3 Main Result

In this section, we present a method for fast and accurate Lipschitz estimation. Section 3.1 reformulates SDP into an explicit recursive form. Section 3.2 derives the following bound backpropagation theory and the FLIP algorithm. Section 3.3 solves the resulting subproblem and analyzes the complexity.

3.1 From the Implicit SDP to an Explicit Recursive Form

In the original SDP (2), objective L^2 depends implicitly on $\{T_i\}$. We show that L^2 can be reformulated into an explicit recursive form using $\{T_i\}$ and $\{W_i\}$. Since the case $L=0$ corresponds to a neural network with constant outputs and is therefore trivial, we assume $L>0$ without loss of generality.

Lemma 1 (Decoupling L and $\{T_i\}$)**.** *Define* $T_i' = \frac{T_i}{L^2}$ *for* $i = 1,\dots,l$*, Problem* (2) *is equivalent to*

$$\min_{L\geq 0,\ \{T_i'\}} L^2 \qquad \text{s.t.} \begin{bmatrix} L^2 H\big(\{T_i'\}\big) & \tilde{W}_l^\top \\ \tilde{W}_l & I \end{bmatrix} \succeq 0, \tag{3}$$

where $\{T_i'\}_{i=1}^l$ *are positive diagonal matrices and* $H(\{T_i'\})$ *is defined as*

$$H\big(\{T_i'\}\big) = \begin{bmatrix} I & -W_0^\top T_1' & 0 & \cdots & 0 \\ -T_1'W_0 & 2T_1' & -W_1^\top T_2' & \cdots & 0 \\ 0 & -T_2'W_1 & 2T_2' & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -W_{l-1}^\top T_l' \\ 0 & 0 & \cdots & -T_l'W_{l-1} & 2T_l' \end{bmatrix}, \quad \tilde{W}_l = \begin{bmatrix} 0 & W_l \end{bmatrix}.$$

The following lemma characterizes the feasibility of Problem (3) in terms of $H(\{T_i'\})$.

Lemma 2 (Feasibility of SDP)**.** *Consider Problem* (3)*, given a fixed* $\{T_i'\}$*, then:*

(i) If $H(\{T_i'\}) \succ 0$*, the optimal* L *exists and admits a closed-form expression in* $\{T_i'\}$*.*

(ii) If $H(\{T_i'\}) \succeq 0$ *is singular, there is no solution in the general case.*

(iii) If $H(\{T_i'\}) \not\succeq 0$*, the problem is infeasible.*

Proof of Lemma 2(i): Since $H\left(\{T_i'\}\right) \succ 0$, each diagonal block satisfies $T_i' \succ 0$. Applying the Schur complement (B.1) sequentially to the diagonal blocks of H, and let the first diagonal block in the upper-left corner be C_i^{-1}, $C_0 = I$. Let $B_i = W_iC_iW_i^\top$, we have

$$\begin{aligned} C_i &= \big(2T_i' - T_i'W_{i-1}C_{i-1}W_{i-1}^\top T_i'\big)^{-1} = (2T_i' - T_i'B_{i-1}T_i')^{-1} = \frac{T_i'^{-1}}{2} + \frac{1}{4}\left(B_{i-1}^{-1} - \frac{T_i'}{2}\right)^{-1} \\ &= \frac{1}{4}\Big(2T_i'^{-1} + B_{i-1} + B_{i-1}\Big(2T_i'^{-1} - B_{i-1}\Big)^{-1}B_{i-1}\Big) \end{aligned} \tag{4}$$

for $i = 1, ..., l$, where the last two steps use the Woodbury matrix identity (B.2). Since the Schur complement preserves positive semidefiniteness, the constraint of Problem (3) reduces to

$$\begin{bmatrix} L^2H(\{T_i'\}) & \tilde{W}_l^\top \\ \tilde{W}_l & I \end{bmatrix} \succeq 0 \Rightarrow \begin{bmatrix} L^2C_l^{-1} & W_l^\top \\ W_l & I \end{bmatrix} \succeq 0 \Rightarrow I - \frac{1}{L^2}W_lC_lW_l^\top \succeq 0 \Rightarrow L^2 \geq \|W_lC_lW_l^\top\|_2,$$

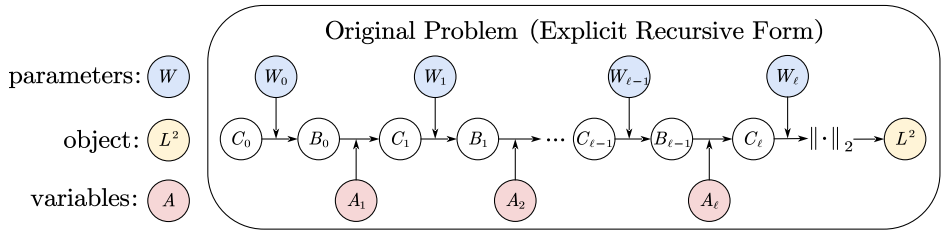


Figure 1: Computation graph for explicit recursive form (5) of SDP framework.

where the second and third inequalities are obtained by the Schur complement. Thus, when $H(\{T_i'\}) \succ 0$, Problem (3) admits a closed-form optimal solution, $L^* = \sqrt{\|W_l C_l W_l^\top\|_2}$. □

Remark Lemma 2(i) shows that, for any $\{T_i'\}$ with $H(\{T_i'\}) \succ 0$, the optimal L admits an explicit recursive expression using $\{T_i'\}$ and $\{W_i\}$. Lemma 2(ii)–(iii) shows that almost all feasible solutions of Problem (3) satisfy $H(\{T_i'\}) \succ 0$. The following lemma gives an equivalent condition for $H(\{T_i'\}) \succ 0$ in the recursive form.

Lemma 3 (Equivalent Condition)**.** $H(\{T_i'\}) \succ 0$ *if and only if* $2T_i'^{-1} - B_{i-1} \succ 0$ *for* $i = 1, \ldots, l$.

Combining Lemmas 1, 2, and 3, we obtain the following explicit recursive form.

Theorem 2 (Explicit Recursive Form for SDP Framework)**.** *Define reparameterized variables* $A_i = 2T_i'^{-1}$ *for* $i = 1, \ldots, l$, $\{A_i\}_{i=1}^l$ *are positive diagonal matrices. Without loss of generality, Problem* (2) *is equivalent to*

$$\min_{\{A_i\}_{i=1}^l} L^2 = \left\|W_l C_l W_l^\top\right\|_2 \quad \text{s.t.} \quad \begin{aligned} &C_0 = I, \\ &B_{i-1} = W_{i-1} C_{i-1} W_{i-1}^\top, \\ &C_i = \frac{1}{4}A_i + \frac{1}{4}B_{i-1} + \frac{1}{4}B_{i-1}\left(A_i - B_{i-1}\right)^{-1} B_{i-1}, \\ &A_i - B_{i-1} \succ 0, \quad i = 1, \ldots, l. \end{aligned} \tag{5}$$

Remark Theorem 2 transforms the original SDP (2) into an explicit recursive optimization problem. Each variable A_i influences the final objective L^2 through the forward recursion of (C_i, B_i) across layers, as illustrated in computation graph (Fig. 1).

3.2 Bound backpropagation

For Theorem 2, objective L^2 depends on all variables $\{A_i\}$. In this subsection, we derive a sequence of upper and lower bounds on the objective, each depending only on partial variables, enabling sequential optimization for each A_i. This leads to the FLIP algorithm for fast Lipschitz estimation.

In the explicit recursive form (5), the main complexity lies in the mapping $(A_i, B_{i-1}) \mapsto C_i$. The following lemma characterizes this mapping.

Lemma 4. *Let* B *be a fixed nonzero positive semidefinite matrix and variable* A *be a diagonal matrix satisfying* $A - B \succ 0$. *Define* $F(A, B) := A + B(A - B)^{-1}B$, *the following properties hold:*

(i) Upper bound: $\min_{A-B\succ 0} \|F(A,B)\|_2 \le 3\|B\|_2$.

(ii) For any matrix W, *lower bound:* $\min_{A-B\succ 0} \left\|WF(A,B)W^\top + WBW^\top\right\|_2 \ge 4\|WBW^\top\|_2$.

By Lemma 4(ii), we derive a sequence of lower bounds on the objective of recursive form (5):

Theorem 3 (Lower Bound Backpropagation)**.** *Objective of* (5) *admits a sequence of lower bounds:*

$$\min_{\{A_i, i\le l\}} \left\|W_l C_l W_l^\top\right\|_2 \ \ge \ \min_{\{A_i,\, i\le l-1\}} \Phi_{l-1}^{\text{low}}(C_{l-1}) \ \ge \ \ldots \ \ge \ \min_{\{A_i,\, i\le l-z\}} \Phi_{l-z}^{\text{low}}(C_{l-z}), \tag{6}$$

where $z = 1, \ldots, l-1$, $\Phi_{l-z}^{\text{low}}(C_{l-z}) := \left\|\Omega_{l-z} C_{l-z} \Omega_{l-z}^\top\right\|_2$, *involving only partial variables* $\{A_i, i \le l - z\}$ *and* $\Omega_i := \prod_{k=i}^l W_k$ *for* $i = 1, \ldots, l$.

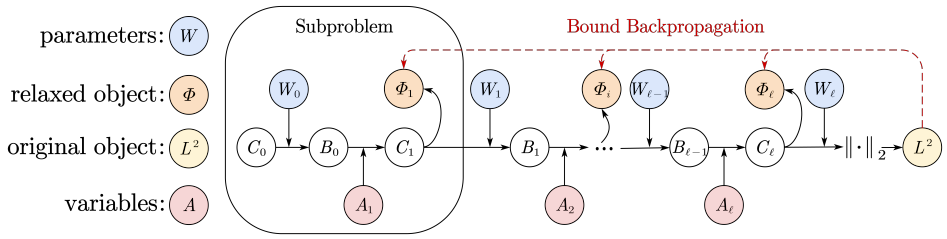


Figure 2: Computation graph for FLIP algorithm 1.

Proof of Theorem 3: We have

$$\begin{aligned}
&\min_{\{A_i, i\le l\}} \|W_l C_l W_l^\top\|_2 = \min_{\{A_i, i\le l-1\}} \min_{A_l \succ B_{l-1}} \frac{1}{4}\|W_l F\left(A_l, B_{l-1}\right) W_l^\top + W_l B_{l-1} W_l^\top\|_2 \\
\ge &\min_{\{A_i, i\le l-1\}} \|W_l B_{l-1} W_l^\top\|_2 = \min_{\{A_i, i\le l-1\}} \|\Omega_{l-1} C_{l-1} \Omega_{l-1}^\top\|_2 \\
\ge &\min_{\{A_i,\, i\le l-z\}} \|\Omega_{l-z} C_{l-z} \Omega_{l-z}^\top\|_2 = \min_{\{A_i,\, i\le l-z\}} \Phi_{l-z}^{\text{low}}(C_{l-z}).
\end{aligned}$$

Starting from the original objective, we repeatedly apply Lemma 4(ii) to obtain lower bounds. The bound over the first $l-z-1$ variables is obtained from the bound over the first $l-z$ variables. As z increases to $l-1$, the lower bound decreases to $\min_{\{A_i,\, i\le 1\}} \Phi_1^{\text{low}}(C_1)$, which only involves first variable A_1. This completes the proof. □

Similarly, by Lemma 4(i), we derive a sequence of upper bounds:

Theorem 4 (Upper Bound Backpropagation)**.** *Objective of* (5) *admits a sequence of upper bounds:*

$$\min_{\{A_i, i\le l\}} \left\|W_l C_l W_l^\top\right\|_2 \le \min_{\{A_i,\, i\le l-1\}} \Phi_{l-1}^{\text{up}}(C_{l-1}) \le \ldots \le \min_{\{A_i,\, i\le l-z\}} \Phi_{l-z}^{\text{up}}(C_{l-z}), \tag{7}$$

where $z = 1, \ldots, l-1$, $\Phi_{l-z}^{\text{up}}(C_{l-z}) = \left(1 - \left(\frac{1}{4}\right)^z\right) s_{l-z}\|C_{l-z}\|_2 + \left(\frac{1}{4}\right)^z \Phi_{l-z}^{\text{low}}(C_{l-z})$, *involving only partial variables* $\{A_i, i \le l-z\}$ *and* $s_i := \prod_{k=i}^{l} \|W_k\|_2^2$.

Remark Theorems 3 and 4 characterize lower and upper bounds of the objective using only the first $l-z$ decision variables $\{A_i, i \le l-z\}$. Leveraging the recursive structure in (5), an important observation emerges: the bounds of the objective are progressively derived from downstream to upstream along the recursive computational graph. We refer to this phenomenon as *bound backpropagation*.

Bound backpropagation provides lower and upper bounds depending only on partial variables, enabling us to decompose the recursive problem into small subproblems and solve it efficiently.

Subproblem Decomposition On partial variable $\{A_i\}_{i=1}^{l-z}$, lower bound $\min_{\{A_i,\, i \le l-z\}} \Phi_{l-z}^{\text{low}}$ and upper bound $\min_{\{A_i, i\le l-z\}} \Phi_{l-z}^{\text{up}}$ represent the necessary and sufficient conditions for minimizing the objective of (5) respectively. We define the combination of Φ_{l-z}^{low} and Φ_{l-z}^{up} as the relaxed objective:

$$\begin{aligned}
\Phi_{l-z}^{\text{comb}}(C_{l-z}) &:= (1-\gamma)\Phi_{l-z}^{\text{low}}(C_{l-z}) + \gamma\, \Phi_{l-z}^{\text{up}}(C_{l-z}) \\
&= \gamma\left(1 - \left(\frac{1}{4}\right)^z\right) s_{l-z}\|C_{l-z}\|_2 + \left(1 - \gamma + \gamma\left(\frac{1}{4}\right)^z\right) \left\|\Omega_{l-z} C_{l-z} \Omega_{l-z}^\top\right\|_2 .
\end{aligned} \tag{8}$$

Here, the constant $0 \le \gamma \le 1$ balances the necessity and sufficiency of minimizing the original objective. However, the coefficient $s_i = \prod_{k=i}^{l} \|W_k\|_2^2$ may substantially overweight $\|C_{l-z}\|_2$ as the depth increases, causing the first term to dominate the relaxed objective. To balance the contributions of the two terms, we calibrate their spectral scales by replacing scalar s_i with $\widetilde{s}_i := \left\|\prod_{k=i}^{l} W_k\right\|_2^2 = \|\Omega_i\|_2^2$, thereby yielding the (balanced) relaxed objective $\Phi_{l-z}(C_{l-z})$. Since each relaxed objective $\Phi_{l-z}(C_{l-z})$ depends only on partial variables such as A_{l-z}, we can solve a sequence of smaller subproblems. The explicit recursive form (5) is decomposed into small subproblems, each optimizing

positive diagonal matrix A_{l-z} with $\Phi_{l-z}(C_{l-z})$ as the relaxed objective:

$$\begin{aligned} \min_{A_{l-z}-B_{l-z-1}\succ 0} \; & \Phi_{l-z}(C_{l-z}) \\ \text{s.t. } C_{l-z} = & \frac{1}{4}A_{l-z} + \frac{1}{4}B_{l-z-1} + \frac{1}{4}B_{l-z-1}(A_{l-z}-B_{l-z-1})^{-1}B_{l-z-1}. \end{aligned} \tag{9}$$

As shown in Fig. 2, along the recursive process, the subproblems provide a fast sequential solution for $A_1, \ldots, A_l$. Since the objective of each subproblem is induced by upper and lower bounds of the objective in the recursive problem (5), minimizing these subproblems still drives the original objective $L^2 = \|W_l C_l W_l^\top\|_2$ to a low value.

Here, we present the FLIP algorithm 1, which efficiently computes a global Lipschitz estimate for the given network weights. SUBSOLVER solves each subproblem (9), as detailed in the next subsection.

Algorithm 1 FLIP: Fast computation of the global Lipschitz constant

Input network weights $\{W_i\}_{i=0}^l$
Compute $\Omega_i = \prod_{k=i}^l W_k$ and $\widetilde{s}_i = \|\Omega_i\|_2^2$ for $i = 1, \ldots, l$
$\widehat{C}_0 \leftarrow I_{n_0}$
for $i = 1$ to l **do**
  $\widehat{B}_{i-1} \leftarrow W_{i-1}\widehat{C}_{i-1}W_{i-1}^\top$
  $A_i^* = \text{SUBSOLVER}(\widehat{B}_{i-1}, \Omega_i, \widetilde{s}_i, i)$ (Algorithm 2 in the subsequent subsection)
  $\widehat{C}_i \leftarrow \frac{1}{4}A_i^* + \frac{1}{4}\widehat{B}_{i-1}(A_i^* - \widehat{B}_{i-1})^{-1}\widehat{B}_{i-1} + \frac{1}{4}\widehat{B}_{i-1}$
end for
$L^2_{\text{LipSDP}} \leftarrow \|W_l\widehat{C}_l W_l^\top\|_2$
return $\widehat{L} = \sqrt{L^2_{\text{LipSDP}}}$

As FLIP approximates the original LipSDP by sequentially solving the relaxed subproblems, we provide the following theory to show that its suboptimality can be bounded a posteriori.

Theorem 5 (Suboptimality Bound)**.** *Let L^2_{LipSDP} denote the optimal objective value of the original LipSDP, and let L^2_{FLIP} denote the objective value attained by FLIP, with intermediate results $\{\widehat{C}_i\}_{i=1}^l$. Define the sandwich gap $\Delta_i(C_i) := \Phi_i^{\text{up}}(C_i) - \Phi_i^{\text{low}}(C_i) \geq 0$, as long as $\Omega_i \neq 0$ for all i, we have*

$$0 \leq L^2_{\text{FLIP}} - L^2_{\text{LipSDP}} \leq \sum_{i=1}^{l} \left[(1-\gamma)\Delta_i(\widehat{C}_i) + \gamma\Delta_i(C_i^{\text{low}})\right], \tag{10}$$

where C_i^{low} minimize Φ_i^{low} over the feasible set with the preceding result $\widehat{C}_{i-1}$.

Remark This theory shows that the gap between the FLIP result and the original LipSDP optimum is guaranteed. The condition $\Omega_i = \prod_{k=i}^l W_k \neq 0$ is trivial and holds in the general case. Each sandwich gap Δ_i controls the maximum error introduced by the approximation of the corresponding FLIP subproblem. And the overall suboptimality of FLIP is bounded by the accumulated sandwich gaps. This bound further guarantees the reliability of FLIP's results.

3.3 Solving the Subproblem

In this subsection, we focus on solving the subproblems in (9) and derive the complexity of FLIP. We first simplify the notation by letting $\eta := \gamma\big(1 - \big(\frac{1}{4}\big)^{l-i}\big) \in [0, 1)$. Moreover, we further assume $A = aI$ for some scalar $a \in \mathbb{R}$, corresponding to the LipSDP-Layer setting [10].

Under these simplifications, the problem in (9) is equivalently reformulated as

$$\begin{aligned} \min_{a\in\mathbb{R}} \quad & J(a) = \eta s\|C\|_2 + (1-\eta)\|\Omega C \Omega^\top\|_2 \\ \text{s.t.} \quad & C = \frac{1}{4}aI + \frac{1}{4}B(aI-B)^{-1}B + \frac{1}{4}B, \quad aI \succ B, \end{aligned} \tag{11}$$

Let $B \succeq 0$ have the eigen decomposition $B = V\Lambda V^\top$, and $\mathbf{u}_{\max}$ denotes its leading eigenvector. We first establish the following result regarding the well-posedness of (11).

Theorem 6 (Well-posedness of Subproblem)**.** *Problem* (11) *is a convex optimization problem with nonempty feasible set. As long as* $\Omega \mathbf{u}_{\max} \neq 0$*, the optimal solution is attainable.*

As the condition $\Omega \mathbf{u}_{\max} \neq 0$ holds almost surely, theorem 6 guarantees the existence of at least one optimal solution in general. Since $J(a)$ is nonsmooth due to the involvement of the spectral norm, we next analyze the subdifferential structure of the objective function, denoted $\partial J(a)$. The expression for $\partial J(a)$ is given in the following proposition.

Proposition 1 (Subdifferential Expression)**.** *The subdifferential of Problem* (11) *at* α *is given by:*

$$\begin{aligned}\partial J(a) =& \frac{\eta s}{4}\left(1-\frac{\lambda_{\max}^2}{(a-\lambda_{\max})^2}\right) \\ &+\frac{1-\eta}{4}\operatorname{conv}\left\{\sum_j [\mathbf{q}]_j^2\left(1-\frac{\lambda_j^2}{(a-\lambda_j)^2}\right)\;\middle|\;\mathbf{q}=V^\top\Omega^\top\mathbf{v},\ \mathbf{v}\in\mathcal{E}_{\max}(\Omega C\Omega^\top),\|\mathbf{v}\|_2=1\right\},\end{aligned} \tag{12}$$

where $\operatorname{conv}(\cdot)$ *denotes the convex hull,* $B = V\Lambda V^\top$*,* $\{\lambda_j\}$ *are the eigenvalues of* Λ*, and* $\lambda_{\max} := \max_j \lambda_j$*. Moreover,* $\mathcal{E}_{\max}(\cdot)$ *denotes the eigenspace associated with the largest eigenvalue.*

Since the problem (11) is convex, the first-order optimality condition implies that α^* is optimal if and only if $0 \in \partial J(\alpha^*)$. The key properties of $\partial J(a)$ are summarized as follows.

Proposition 2 (Monotonicity of $\partial J(a)$)**.** *The subdifferential* $\partial J(a)$ *is monotone in the feasible region, i.e., for any two points* $a_1, a_2 \in \mathbb{R}$ *in the feasible region and any* $g_1 \in \partial J(a_1), g_2 \in \partial J(a_2)$*, it holds* $(g_1 - g_2)(a_1 - a_2) \geq 0$.

Theorem 7 (Bracketing Interval)**.** *If* $\Omega \mathbf{u}_{\max} \neq 0$*, there exist* $a_{\text{low}}, a_{\text{up}} \in \mathbb{R}$ *in the feasible region of Problem* (11)*, such that* $g \leq 0$ *for any* $g \in \partial J(a_{\text{low}})$ *and* $g \geq 0$ *for any* $g \in \partial J(a_{\text{up}})$*. Specifically,*

(i) If $\eta > 0$*,* a_{low} *has a closed form* $a_{\text{low}} = \lambda_{\max} + \frac{\lambda_{\max}}{\sqrt{1+\frac{1-\eta}{\eta s}\|\Omega\|_2^2}}$. (13)

(ii) a_{up} *has a closed form* $a_{\text{up}} = 2\lambda_{\max}$.

Remark Proposition 2 shows that $\partial J(a)$ is monotone, and Theorem 7 gives the bracketing interval for $\partial J(a)$ with opposite signs. Hence the subproblem reduces to a one-dimensional monotone root-finding problem. When $\eta > 0$, both endpoints of the bracket are available in closed form, so a secant step can be used to accelerate convergence. When $\eta = 0$, the lower endpoint a_{low} exists but has no closed-form; we therefore start from the boundary value $a_{\text{low}} = \lambda_{\max}$ and use bisection until a valid non-positive subdifferential is obtained. This leads to the subproblem solver, as summarized in Algorithm 2. PICKSUB(a) denotes an arbitrary selection from the subdifferential defined in (12).

Algorithm 2 SUBSOLVER: Subproblem Solver

Input: B, Ω, s, i; default tolerance $\varepsilon = 10^{-6}$, default $\gamma = 0.1$
Compute $\eta = \gamma\big(1 - \big(\frac{1}{4}\big)^{l-i}\big)$ and the eigen decomposition $B = V\Lambda V^\top$
Set $a_{\text{up}} \leftarrow 2\lambda_{\max}$ and $g_{\text{up}} \leftarrow$ PICKSUB(a_{up})
if $\eta = 0$ **then** $a_{\text{low}} \leftarrow \lambda_{\max}$, $g_{\text{low}} \leftarrow$ NULL; **else** $a_{\text{low}} \leftarrow$ Eq.(13), $g_{\text{low}} \leftarrow$ PICKSUB(a_{low})
while $|a_{\text{up}} - a_{\text{low}}| > \varepsilon\lambda_{\max}$ **do**
    if $g_{\text{low}} \neq$ NULL **then** $a \leftarrow \dfrac{a_{\text{low}} g_{\text{up}} - a_{\text{up}} g_{\text{low}}}{g_{\text{up}} - g_{\text{low}}}$; **else** $a \leftarrow \dfrac{a_{\text{low}} + a_{\text{up}}}{2}$
    $g \leftarrow$ PICKSUB(a)
    if $g \leq 0$ **then** $a_{\text{low}} \leftarrow a$, $g_{\text{low}} \leftarrow g$; **else** $a_{\text{up}} \leftarrow a$, $g_{\text{up}} \leftarrow g$
end while
$a^\star \leftarrow \dfrac{a_{\text{low}} + a_{\text{up}}}{2}$; **return** $A^* = a^\star I$

We now analyze the time complexity of the subproblem solver and the overall FLIP algorithm.

Proposition 3 (Time Complexity)**.** *For a neural network with* l *hidden layers, width* n*, and output dimension* d*, the time complexities of solving each subproblem and of the overall FLIP algorithm are* $O(n^3 + Kd^3)$ *and* $O(ln^3 + Kld^3)$ *respectively.* K *denotes the number of iterations of Algorithm 2, which is at most logarithmic* $O(\log(1/\varepsilon))$*, where* ε *is the tolerance of Algorithm 2.*

Remark On neural networks of the same size, FLIP has a time complexity close to the naive upper bound, whose complexity is $O(ln^3 + d^3)$. The complexity of FLIP is much lower than the original SDP framework, which reaches $O(K_{\text{SDP}} l^4 n^4)$. FLIP is also significantly faster than ECLipsE, an efficient improvement of the SDP framework with complexity $O(K_{\text{ECLipsE}} ln^4)$ [30]. The low time complexity of FLIP makes it suitable for Lipschitz estimation of large neural networks. The significant improvement over existing methods is demonstrated in the experiments.

4 Experiments

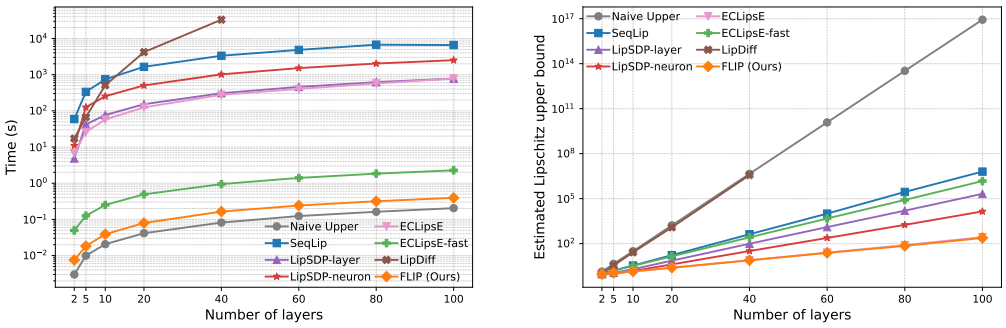
We conducted experiments on both randomly initialized and trained neural networks to demonstrate that our FLIP algorithm can compute tighter Lipschitz estimates more efficiently.

Baselines We compared FLIP with seven existing methods: the naive upper bound $L_{\text{naive}} = \prod_{i=0}^{l} \|W_i\|_2$, SeqLip [26], LipSDP-layer and LipSDP-neuron [10], ECLipsE and ECLipsE-fast [30], and LipDiff [27]. Most of these methods have only been tested on small networks (≤ 5 layers or width ≤ 100). In contrast, we scale our experiments to medium (width 300) and large (width 1000) networks. Due to the computational limitations of SDP solvers, LipSDP-layer and LipSDP-neuron employ the standard partitioning approximation [10], dividing deep networks into shallow subnetworks and multiplying their Lipschitz bounds. Based on the max capacity of SDP solver, we partition medium and large networks into subnetworks of 5 and 3 layers, respectively. All SDP are solved to optimality using the solver's default tolerance. LipDiff was accelerated using an NVIDIA RTX 4090 GPU (24GB), while all other methods were run on an Intel Xeon 6530 CPU.

4.1 Randomly Initialized Neural Networks

We first evaluated the Lipschitz upper bound estimators on randomly initialized feedforward networks. Networks were initialized using a standard Xavier-style scheme [11], where each weight is independently sampled from a uniform distribution $U\left[-\frac{3}{4}\sqrt{\frac{3}{n}}, \frac{3}{4}\sqrt{\frac{3}{n}}\right]$, with n being the layer width. From random matrix theory [25], the spectral norm of a weight matrix $W \in \mathbb{R}^{n\times n}$ satisfies $\|W\|_2 \approx 1.5$, and the overall network Lipschitz constant may grow exponentially with depth. We tested medium (width 300) and large (width 1000) networks with depths from 2 to 100 and an output dimension of 20, reaching up to 100M parameters. For each network, we measured Lipschitz bounds and runtime for all methods, shown in Fig. 3 and Fig. 4 (log-scale for clarity).

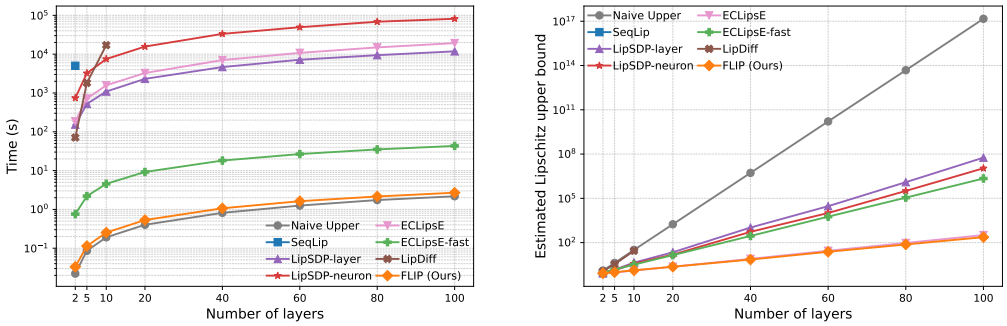
Computational Efficiency As shown in Fig. 3(a) and Fig. 4(a), FLIP's runtime is comparable to the naive upper bound and several orders of magnitude faster than other methods. This confirms our earlier claim that FLIP and the naive upper bound have similar time complexity while being significantly faster than alternatives. For large networks (Fig. 4(a)), FLIP requires only a few seconds—about 10× faster than ECLipsE-fast—while SeqLip, LipSDP, and ECLipsE take several hours. LipDiff encounters Out-of-Memory (OOM) errors as depth increases, and SeqLip fails to converge on large networks.



(a) Runtime (log-scaled) vs. depth

(b) Estimated upper bound (log-scaled) vs. depth

Figure 3: Results on medium-scale networks (width 300).



(a) Runtime (log-scaled) vs. depth

(b) Estimated upper bound (log-scaled) vs. depth

Figure 4: Results on large-scale networks (width 1000).

Table 1: Comparison of different Lipschitz estimation methods on a 100M-parameter network.

Method	Time(s)	Lipschitz Bound	Comparison
FLIP (Ours)	**2.69**	**234.3**	**fast and accurate**
Naive Upper	2.18	1.47×10^{17}	fast but extremely loose
ECLipsE-fast [30]	43.17	2.21×10^{6}	$16.0\times$ slower, $9421\times$ looser
ECLipsE [30]	19249	323.6	$7150\times$ slower, slightly looser
LipSDP-layer [10]	11814	5.70×10^{7}	$4389\times$ slower, much looser
LipSDP-neuron [10]	81554	1.09×10^{7}	$30295\times$ slower, much looser
SeqLip [26]	–	–	non-convergence for large-scale networks
LipDiff [27]	–	–	CUDA OOM for large-scale networks

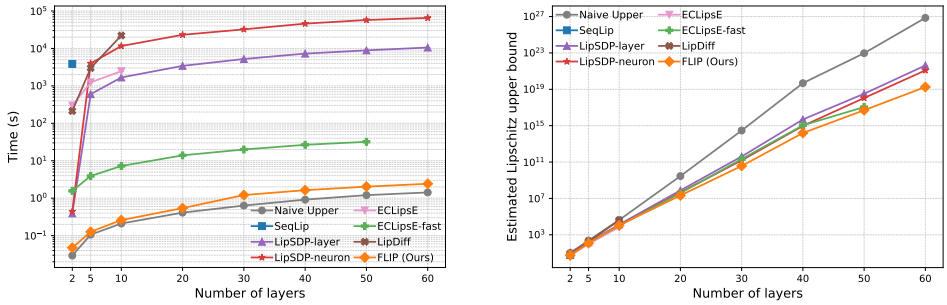
Tightness of Lipschitz Estimates Fig. 3(b) and 4(b) show that, due to the exponential relationship between depth and the true Lipschitz constant in random networks, log-scale plots appear linear. FLIP and ECLipsE consistently produce the tightest upper bounds, both outperforming LipSDP, which partitions deep networks into shallow sub-networks and multiplies their Lipschitz bounds [10], ignoring deep-layer interactions. FLIP and ECLipsE improve on LipSDP by decomposing the original SDP into nested subproblems, avoiding the loose product of isolated sub-network bounds. Other methods (SeqLip, LipDiff, ECLipsE-fast) yield significantly looser bounds.

100M-Parameter Networks Table 1 reports results for large networks with width 1000 and depth 100 (corresponding to Fig. 4). Even in this scenario, FLIP computes the tightest Lipschitz bounds in under 3 seconds. Compared to ECLipsE, FLIP achieves a $7150\times$ speedup with tighter bounds, thanks to theoretically derived subproblem objectives and efficient secant-based solutions, rather than heuristic SDP objectives. Other methods are at least three orders of magnitude slower or produce much looser bounds.

4.2 Neural Networks Trained on SVHN

SVHN [19] consists of $32 \times 32 \times 3$ images with over 600,000 training samples, making it suitable for large networks. We flattened images to 3072-dimensional vectors and trained networks with hidden width 1000, ReLU activations, depth 2–60, and output dimension 10. We held out 20% of training data for validation and trained the remaining data with a learning rate of 1.0×10^{-4} using early stopping for up to 10,000 steps. Test accuracies ranged from 84.5% to 89.2%.

Runtime results for trained networks are shown in Fig. 5(a). As in the random network experiments, LipDiff and SeqLip failed due to OOM and non-convergence, respectively. ECLipsE also fails as depth increases, as trained networks may violate its stronger assumptions [30]. FLIP remains as fast as the naive upper bound and significantly faster than all alternatives. Fig. 5(b) shows that FLIP produces the tightest Lipschitz bounds, with differences clearly visible even on a log scale.



(a) Runtime (log-scaled) vs. depth

(b) Estimated upper bound (log-scaled) vs. depth

Figure 5: Results on trained networks (width 1000).

Table 2: Sanity check across network widths.

Width	6	8	10	12	14	16	18	20
Upper Bound	1.484	2.665	2.157	1.984	2.554	4.152	4.040	3.202
FLIP	1.089	1.654	1.406	1.041	1.860	2.045	2.016	1.640
Lower Bound	0.976	1.192	0.998	0.713	1.174	1.409	1.353	1.117

Table 3: Sanity check across network depths.

Depth	2	3	4	5	6	7
Upper Bound	2.214	2.492	3.086	4.874	6.791	6.156
FLIP	1.317	1.412	1.692	3.100	4.234	4.529
Lower Bound	0.915	0.997	1.323	1.910	2.709	2.564

4.3 Sanity Check on Small Neural Networks

To further validate FLIP, we conduct sanity checks on small networks using LipMIP [15], which can exactly compute local Lipschitz constants under ℓ_1 and ℓ_∞ norms. We consider randomly generated, bias-free ReLU networks with scalar outputs. Since the network is positively homogeneous, $[-1, 1]^n$ contains all activation patterns, so its local Lipschitz constant over this domain equals the global one.

Let L_1, L_2, and L_∞ denote the global Lipschitz constants under the corresponding norms. For scalar-output networks, it holds that: $\max(L_1, L_\infty/\sqrt{n}) \leq L_2 \leq \min(\sqrt{n}L_1, L_\infty)$. We use LipMIP to compute the exact L_1, L_∞, thereby obtaining an interval for L_2 estimation (FLIP) as a sanity check.

We first evaluate three-layer networks with widths n from 6 to 20, and then vary the depth from 2 to 7 while fixing the width at 10. Tables 2 and 3 report the resulting intervals and FLIP estimates. In both experiments, FLIP consistently produces estimates within the intervals derived from exact LipMIP results, further providing a sanity check across network widths and depths.

5 Conclusions

In this paper, we present a fast and accurate global Lipschitz estimation method for large feedforward neural networks. By applying the Schur complement, we transform LipSDP into an explicit recursive form, observing that its optimization objectives have upper and lower bounds along the recursive computation graph, corresponding to sufficiency and necessity for the SDP. Using these bounds, we construct subproblems with partial decision variables, each efficiently solvable via a secant method. Experiments on both random and trained networks demonstrate FLIP achieves the most accurate Lipschitz bounds with runtimes at several orders of magnitude faster than existing methods.

Future Work This study focuses on classical feedforward networks with linear and activation layers. Since pooling layers and residual connections can also be represented in SDP form, we aim to extend FLIP to other large architectures such as CNNs or ResNets, facilitating the study and analysis of Lipschitz properties in practical large-scale models.

Acknowledgments and Disclosure of Funding

This work was conducted as independent research, with all associated expenses personally covered by the author Hongbo Chen. No external funding was received, and the authors declare no competing financial interests.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. Pmlr, 2017.

[2] Anil Aswani, Humberto Gonzalez, S Shankar Sastry, and Claire Tomlin. Provably safe and robust learning-based model predictive control. *Automatica*, 49(5):1216–1226, 2013.

[3] Radu Balan, Maneesh Singh, and Dongmian Zou. Lipschitz properties for deep convolutional networks. *arXiv preprint arXiv:1701.05217*, 2017.

[4] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

[5] Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):411–444, 2022.

[6] Valérie Castin, Pierre Ablin, and Gabriel Peyré. How smooth is attention? *arXiv preprint arXiv:2312.14820*, 2023.

[7] Hongbo Chen and Li Charlie Xia. General quantification of covariate and concept shifts. *arXiv preprint arXiv:2609.11918*, 2026.

[8] George Dasoulas, Kevin Scaman, and Aladin Virmaux. Lipschitz normalization for self-attention layers with application to graph neural networks. In *International Conference on Machine Learning*, pages 2456–2466. PMLR, 2021.

[9] 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.

[10] 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.

[11] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[12] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

[13] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, UK, 2 edition, 2013.

[14] Xixu Hu, Runkai Zheng, Jindong Wang, Cheuk Hang Leung, Qi Wu, and Xing Xie. Specformer: Guarding vision transformer robustness via maximum singular value penalization. In *European Conference on Computer Vision*, pages 345–362. Springer, 2024.

[15] 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.

[16] Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In *International Conference on Machine Learning*, pages 5562–5571. PMLR, 2021.

[17] Fabian Latorre, Paul Rolland, and Volkan Cevher. Lipschitz constant estimation of neural networks via sparse polynomial optimization. *arXiv preprint arXiv:2004.08688*, 2020.

[18] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[19] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 4. Granada, 2011.

[20] Xianbiao Qi, Jianan Wang, Yihao Chen, Yukai Shi, and Lei Zhang. Lipsformer: Introducing lipschitz continuity to vision transformers. *arXiv preprint arXiv:2304.09856*, 2023.

[21] Ievgen Redko, Amaury Habrard, and Marc Sebban. Theoretical analysis of domain adaptation with optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 737–753. Springer, 2017.

[22] Wenjie Ruan, Xiaowei Huang, and Marta Kwiatkowska. Reachability analysis of deep neural networks with provable guarantees. *arXiv preprint arXiv:1805.02242*, 2018.

[23] 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.

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

[25] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices., 2012.

[26] Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in neural information processing systems*, 31, 2018.

[27] Zi Wang, Bin Hu, Aaron J Havens, Alexandre Araujo, Yang Zheng, Yudong Chen, and Somesh Jha. On the scalability and memory efficiency of semidefinite programs for lipschitz constant estimation of neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

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

[29] Xiaojun Xu, Linyi Li, Yu Cheng, Subhabrata Mukherjee, Ahmed Hassan Awadallah, and Bo Li. Certifiably robust transformers with 1-lipschitz self-attention. 2022.

[30] Yuezhu Xu and S Sivaranjani. Eclipse: Efficient compositional lipschitz constant estimation for deep neural networks. *Advances in Neural Information Processing Systems*, 37:10414–10441, 2024.

[31] He Yin, Peter Seiler, and Murat Arcak. Stability analysis using quadratic constraints for systems with neural network controllers. *IEEE Transactions on Automatic Control*, 67(4):1980–1987, 2021.

[32] Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

[33] 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.

[34] Dongmian Zou, Radu Balan, and Maneesh Singh. On lipschitz bounds of general convolutional neural networks. *IEEE Transactions on Information Theory*, 66(3):1738–1759, 2019.

A Related Work

Estimating the Lipschitz constant of neural networks has been studied from several complementary perspectives. A classical and widely used approach is to upper bound the Lipschitz constant by the product of the operator norms of the weight matrices [23]. This idea also underlies a series of spectral-norm-based regularization and training methods, which aim to improve generalization, stability, or robustness by controlling layer-wise spectral norms [32, 18, 24, 12]. Such methods are simple, differentiable, and highly scalable, but they largely ignore the coupling across layers and therefore often produce conservative bounds. To obtain tighter estimates, [26] proposed algorithmic estimators such as AutoLip and SeqLip, while other works formulate Lipschitz estimation through sparse polynomial optimization or semidefinite programming [10, 17]. In particular, LipSDP exploits quadratic constraints induced by slope-restricted activations and provides certified global upper bounds that are often significantly tighter than naive spectral products [10]. However, these tighter optimization-based formulations usually come with much higher computational costs: polynomial and SDP-based methods are difficult to scale to deep or wide networks, while scalable relaxations often trade estimation accuracy for tractability.

Recent work has therefore focused on improving the scalability of optimization-based Lipschitz estimation. For example, eigenvalue-optimization and first-order variants of SDP-based methods reduce the memory burden of standard interior-point solvers and have been applied to larger CNNs and ImageNet-scale models [27]. Compositional SDP approaches further decompose the large matrix verification problem into layer-wise subproblems, leading to faster estimates with different accuracy–efficiency trade-offs [30]. Despite these advances, existing methods still leave a substantial gap between the speed of simple spectral products and the tightness of heavier optimization-based certificates. This motivates the need for Lipschitz estimation methods that are simultaneously accurate, theoretically certified, and practical for large feedforward networks.

Beyond global Lipschitz estimation for standard feedforward networks, Lipschitz continuity has also been studied in several related directions. Some works analyze or regularize specific architectures, including convolutional networks and attention-based models [4, 3, 34, 16, 6]. Others design network structures with prescribed Lipschitz guarantees, such as normalized attention layers, Lipschitz-constrained vision transformers, or singular-value-penalized transformer architectures [8, 20, 29, 14]. Another active line of work focuses on local Lipschitz constants and formal robustness verification, where bound propagation, linear relaxations, and quadratic-constraint-based certificates are used to certify robustness around individual inputs [28, 33, 22, 9]. These works are closely related in motivation, but they address different goals from our setting: they often focus on training, architectural design, or local certificates, whereas our focus is fast and accurate global Lipschitz estimation for large feedforward networks.

B Technical Tools

B.1 Schur Complement

Lemma 5 (Schur complement, [13])**.** *Let*

$$\Phi = \begin{bmatrix} A & B^\top \\ B & C \end{bmatrix}$$

be a symmetric block matrix, where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times n}$, and $C \in \mathbb{R}^{m\times m}$. Suppose that A and C are symmetric and $C \succ 0$. Then:

(i) $\Phi \succeq 0 \iff A - B^\top C^{-1} B \succeq 0.$

(ii) $\Phi \succ 0 \iff A - B^\top C^{-1} B \succ 0.$

B.2 Woodbury Matrix Identity

Lemma 6 (Woodbury matrix identity, [13])**.** *Let $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{k\times k}$, $U \in \mathbb{R}^{n\times k}$, and $V \in \mathbb{R}^{k\times n}$. Suppose that A and C are invertible, and that $C^{-1} + VA^{-1}U$ is invertible. Then*

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U\left(C^{-1} + VA^{-1}U\right)^{-1} VA^{-1}.$$

C Full proofs

C.1 Proof of Lemma 1

It can be observed that Problems (2) and (3) are equivalent if $M' \succeq 0$ is equivalent to $M \succeq 0$. We now proceed to prove that $M' \succeq 0$ if and only if $M \succeq 0$.

Applying Lemma 5(i) to M' with respect to the lower-right block $I \succ 0$, we know that $M' \succeq 0$ if and only if

$$L^2 H\big(\{T_i'\}\big) - \tilde{W}_l^\top I^{-1} \tilde{W}_l \succeq 0.$$

With $T_i' = \frac{T_i}{L^2}$ and the definition of $H(\{T_i'\})$ in mind, it can be easily verified that

$$L^2 H\big(\{T_i'\}\big) - \tilde{W}_l^\top I^{-1} \tilde{W}_l = M,$$

which completes the proof.

C.2 Proof of Lemma 2

C.2.1 Proof of statement (i)

The proof can be seen in Section 3.1.

C.2.2 Proof of statement (ii)

For a vector $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$, we have

$$\begin{aligned}
&\mathbf{x}^\top M' \mathbf{x} \\
&= (\mathbf{x}_1, \mathbf{x}_2)^\top \begin{pmatrix} L^2 H\left(\{T_i'\}\right) & \tilde{W}_l^\top \\ \tilde{W}_l & I \end{pmatrix} (\mathbf{x}_1, \mathbf{x}_2) \\
&= L^2 \mathbf{x}_1^\top H\left(\{T_i'\}\right) \mathbf{x}_1 + \|\mathbf{x}_2\|_2^2 + 2\mathbf{x}_1^\top \tilde{W}_l^\top \mathbf{x}_2.
\end{aligned}$$

Since $H\left(\{T_i'\}\right)$ is singular positive semidefinite, choose $\mathbf{x}_1$ such that $\mathbf{x}_1^\top H\left(\{T_i'\}\right) \mathbf{x}_1 = 0$, and set $\mathbf{x}_2 = -\tilde{W}_l \mathbf{x}_1$. Then the above expression evaluates to $-\|\tilde{W}_l \mathbf{x}_1\|_2^2$.

When $\mathcal{N}(H) \nsubseteq \mathcal{N}(\tilde{W}_l)$, there exists some vector $\mathbf{x}_1$ satisfying $\mathbf{x}_1^\top H\left(\{T_i'\}\right) \mathbf{x}_1 = 0$ while $\tilde{W}_l \mathbf{x}_1$ is not the zero vector. Consequently, there exists a vector $\mathbf{x}$ such that $\mathbf{x}^\top M' \mathbf{x} < 0$, violating the constraint. Hence, Problem (3) does not have a solution.

C.2.3 Proof of statement (iii)

Given that $M' \succeq 0$ has a necessary condition that its upper-left block, $L^2 H\left(\{T_i'\}\right)$, is semipositive, with $L^2 > 0$, we directly complete the proof.

C.3 Proof of Lemma 3

Proof of Sufficiency: Since $H\left(\{T_i'\}\right) \succ 0$, each diagonal block satisfies $T_i' \succ 0$. Apply the Schur complement in Lemma 5(ii) sequentially along the diagonal blocks of $H\left(\{T_i'\}\right)$. By properties of the Schur complement, each upper-left block in the process satisfies

$$2T_i' - T_i' W_{i-1} C_{i-1} W_{i-1}^\top T_i' = T_i' \left(2{T_i'}^{-1} - W_{i-1} C_{i-1} W_{i-1}^\top\right) T_i' \succ 0.$$

Since congruence transformations preserve definiteness, we obtain

$$2{T_i'}^{-1} - W_{i-1} C_{i-1} W_{i-1}^\top = 2{T_i'}^{-1} - B_{i-1} \succ 0.$$

Proof of Necessity: First, we prove by induction that each $T_i' \succ 0$ and $C_i \succ 0$.

We have $C_0 = I_{n_0} \succ 0$ and $B_0 = W_0 C_0 W_0^\top \succeq 0$.

Assume $B_{i-1} \succeq 0$. Since $2{T'_i}^{-1} - B_{i-1} \succ 0$, it follows that ${T'_i}^{-1} \succ 0$. Then

$$C_i = \frac{1}{4}\left(2{T'_i}^{-1} + B_{i-1}\left(2{T'_i}^{-1} - B_{i-1}\right)^{-1} B_{i-1} + B_{i-1}\right) \succ 0,$$

and consequently $B_i = W_i C_i W_i^\top \succeq 0$.

From $C_i \succ 0$, we obtain

$$C_i^{-1} = 2T'_i - T'_i W_{i-1} C_{i-1} W_{i-1}^\top T'_i \succ 0.$$

Starting from $C_l^{-1} \succ 0$ and expanding to the upper-left corner using Schur complements, we finally obtain $H\left(\{T'_i\}\right)$. By properties of the Schur complement, it follows that $H\left(\{T'_i\}\right) \succ 0$.

C.4 Proof of Theorem 2

By Lemma 1, after the reparameterization $T'_i = \frac{T_i}{L^2}$ for $i = 1, \ldots, l$, the original Problem (2) is equivalent to Problem (3). For fixed $\{T'_i\}_{i=1}^l$, Lemma 2 shows that, in the nontrivial feasible case, we restrict to $H(\{T'_i\}) \succ 0$. Moreover, the optimal value admits the recursive representation $L^2 = \|W_l C_l W_l^\top\|_2$, where

$$\begin{aligned} &C_0 = I, \qquad B_{i-1} = W_{i-1} C_{i-1} W_{i-1}^\top, \\ &C_i = \frac{1}{4}\left(2{T'_i}^{-1} + B_{i-1} + B_{i-1}\left(2{T'_i}^{-1} - B_{i-1}\right)^{-1} B_{i-1}\right). \end{aligned}$$

By Lemma 3, the condition $H(\{T'_i\}) \succ 0$ is equivalent to $2{T'_i}^{-1} - B_{i-1} \succ 0$ for $i = 1, \ldots, l$. Now define $A_i := 2{T'_i}^{-1}$ for $i = 1, \ldots, l$. Since each T'_i is positive diagonal, each A_i is also positive diagonal. Substituting this reparameterization into the recursion above gives

$$\begin{aligned} &C_0 = I, \qquad B_{i-1} = W_{i-1} C_{i-1} W_{i-1}^\top, \\ &C_i = \frac{1}{4}A_i + \frac{1}{4}B_{i-1} + \frac{1}{4}B_{i-1}(A_i - B_{i-1})^{-1} B_{i-1}, \end{aligned}$$

with the constraint

$$A_i - B_{i-1} \succ 0.$$

Therefore Problem (2) is equivalently written as the explicit recursive form in Problem (5). Hence the proof is complete.

C.5 Proof of Lemma 4

Proof of (i): Since B is a nonzero positive semidefinite matrix, it admits the orthogonal decomposition $B = V \Lambda V^\top$, where V is an orthogonal matrix satisfying

$$V V^\top = V^\top V = I, \tag{14}$$

and Λ is a diagonal matrix containing the eigenvalues of B.

Let $A = 2\|B\|_2 I$, which clearly satisfies $A \succ B$. Then, by $F(A, B) = A + B(A - B)^{-1} B$ and (14), we have

$$\begin{aligned} \|F(A,B)\|_2 =& \|F(2\|B\|_2 I, B)\|_2 \\ =& \left\|2\|B\|_2 I + B(2\|B\|_2 I - B)^{-1} B\right\|_2 \\ =& \left\|2\|B\|_2 V V^\top + V\Lambda V^\top \left(2V\|B\|_2 V^\top - V\Lambda V^\top\right)^{-1} V\Lambda V^\top\right\|_2 \\ =& \left\|2\|B\|_2 V V^\top + V\Lambda \left(2\|B\|_2 I - \Lambda\right)^{-1} \Lambda V^\top\right\|_2 \\ =& \left\|2\|B\|_2 I + \Lambda^2 \left(2\|B\|_2 I - \Lambda\right)^{-1}\right\|_2, \end{aligned}$$

where the fourth equality follows from

$$\left(2V\|B\|_2 V^\top - V\Lambda V^\top\right)^{-1} = V\left(2\|B\|_2 I - \Lambda\right)^{-1} V^\top, \tag{15}$$

since $2\|B\|_2 I - \Lambda$ is diagonal, and the last equality uses the fact that orthogonal transformations preserve the spectral norm.

Observe that $2\|B\|_2 I + \Lambda^2(2\|B\|_2 I - \Lambda)^{-1}$ is diagonal, and each diagonal entry satisfies

$$\left[2\|B\|_2 I + \Lambda^2(2\|B\|_2 I - \Lambda)^{-1}\right]_{ii} \leq 3\|B\|_2,$$

with equality attained when the index i corresponds to the largest eigenvalue of Λ. Therefore,

$$\|F(A,B)\|_2 = \left\|2\|B\|_2 I + \Lambda^2(2\|B\|_2 I - \Lambda)^{-1}\right\|_2 = 3\|B\|_2.$$

Since this bound is attained by choosing $A = 2\|B\|_2 I$, we obtain

$$\min_{A-B\succ 0} \|F(A,B)\|_2 \leq \|F(2\|B\|_2 I, B)\|_2 = 3\|B\|_2,$$

which completes the proof of (i).

Proof of (ii): For any diagonal matrix A and any nonzero positive semidefinite matrix B satisfying $A \succ B$, we have

$$\begin{aligned}
&WF(A,B)W^\top - 3WBW^\top \\
=&WAW^\top + WB(A-B)^{-1}BW^\top - 3WBW^\top \\
=&W(A-B)W^\top + WB(A-B)^{-1}BW^\top - 2WBW^\top \\
=&W(A-B)W^\top + WB(A-B)^{-1}BW^\top \\
&\quad - 2W(A-B)^{1/2}(A-B)^{-1/2}BW^\top \\
=&\Big(W(A-B)^{1/2} - WB(A-B)^{-1/2}\Big)\Big(W(A-B)^{1/2} - WB(A-B)^{-1/2}\Big)^\top \succeq 0,
\end{aligned}$$

where the first equality follows from the definition of $F(A,B)$.

Hence,

$$WF(A,B)W^\top + WBW^\top \succeq 4WBW^\top.$$

Since both sides are positive semidefinite, taking the spectral norm yields

$$\left\|WF(A,B)W^\top + WBW^\top\right\|_2 \geq 4\|WBW^\top\|_2.$$

Given that the above inequality holds for any admissible A, minimizing the left-hand side over all A such that $A - B \succ 0$ preserves the inequality, which completes the proof of (ii).

C.6 Proof of Theorem 4

We have

$$
\begin{aligned}
&\min_{\{A_i\}} \|W_l C_l W_l^\top\|_2 \\
\le\ &\min_{\{A_i\}-A_l} \min_{A_l} \frac{1}{4}\|W_l F(A_l, B_{l-1}) W_l^\top\|_2 + \frac{1}{4}\|\Omega_{l-1} C_{l-1} \Omega_{l-1}^\top\|_2 \\
\le\ &\min_{\{A_i\}-A_l} \frac{3}{4}\|W_l\|_2^2 \|B_{l-1}\|_2 + \frac{1}{4}\|\Omega_{l-1} C_{l-1} \Omega_{l-1}^\top\|_2 \\
\le\ &\min_{\{A_i\}-A_l} \frac{3}{4} s_{l-1}\|C_{l-1}\|_2 + \frac{1}{4}\|\Omega_{l-1} C_{l-1} \Omega_{l-1}^\top\|_2 \\
\le\ &\min_{\{A_i\}-A_l} \frac{3}{16} s_{l-1}\|F(A_{l-1}, B_{l-2})\|_2 + \frac{3}{16} s_{l-1}\|B_{l-2}\|_2 \\
&\quad + \frac{1}{16}\|\Omega_{l-1} F(A_{l-1}, B_{l-2}) \Omega_{l-1}^\top\|_2 + \frac{1}{16}\|\Omega_{l-2} C_{l-2} \Omega_{l-2}^\top\|_2 \\
\le\ &\min_{\{A_i\}-A_l-A_{l-1}} \frac{9}{16} s_{l-1}\|B_{l-2}\|_2 + \frac{3}{16} s_{l-1}\|B_{l-2}\|_2 \\
&\quad + \frac{3}{16} s_{l-1}\|B_{l-2}\|_2 + \frac{1}{16}\|\Omega_{l-2} C_{l-2} \Omega_{l-2}^\top\|_2 \\
\le\ &\min_{\{A_i\}-A_l-A_{l-1}} \frac{15}{16} s_{l-2}\|C_{l-2}\|_2 + \frac{1}{16}\|\Omega_{l-2} C_{l-2} \Omega_{l-2}^\top\|_2 \\
\le\ &\min_{\{A_i,\, i\le l-z\}} \left(1 - \left(\frac{1}{4}\right)^z\right) s_{l-z}\|C_{l-z}\|_2 + \left(\frac{1}{4}\right)^z \|\Omega_{l-z} C_{l-z} \Omega_{l-z}^\top\|_2 \\
=\ &\min_{\{A_i,\, i\le l-z\}} \Phi_{l-z}^{\mathrm{up}}\left(C_{l-z}\right).
\end{aligned}
$$

Here, the first inequality follows from the recursive structure in (5) together with $\Omega_i = \prod_{k=i}^{l} W_k$ and $s_i := \prod_{k=i}^{l} \|W_k\|_2^2$, the second inequality follows from Lemma 4, and the remaining inequalities are obtained by repeatedly applying the same argument recursively from $\{A_i\}$ to $\{A_i \mid i \le l - z\}$. Then we complete the proof.

C.7 Proof of Theorem 5

For any feasible intermediate state C_i, define the optimal continuation value as

$$
V_i(C_i) := \inf_{A_{i+1},\ldots,A_l} \|W_l C_l W_l^\top\|_2, \tag{16}
$$

where the remaining variables are jointly optimized over the feasible set.

By definition, $V_0(C_0) = L^2_{\mathrm{LipSDP}}$ and $V_l(\widehat{C}_l) = L^2_{\mathrm{FLIP}}$.

For each $i = 1, \ldots, l$, the continuation values satisfy

$$
V_{i-1}(\widehat{C}_{i-1}) = \inf_{A_i} V_i\big(C_i(A_i, \widehat{C}_{i-1})\big). \tag{17}
$$

Define the suboptimality introduced by the i-th FLIP decision as

$$
\delta_i := V_i(\widehat{C}_i) - V_{i-1}(\widehat{C}_{i-1}) \ge 0. \tag{18}
$$

Summing over all layers yields the telescoping identity

$$
L^2_{\mathrm{FLIP}} - L^2_{\mathrm{LipSDP}} = \sum_{i=1}^{l} \delta_i. \tag{19}
$$

By the bound backpropagation results in Theorems 3 and 4, the continuation value satisfies

$$
\Phi_i^{\mathrm{low}}(C_i) \le V_i(C_i) \le \Phi_i^{\mathrm{up}}(C_i) \tag{20}
$$

for every feasible C_i.

Recall that C_i^{low} minimizes Φ_i^{low} over the feasible i-th subproblem with the preceding result $\widehat{C}_{i-1}$. Consequently,

$$\begin{aligned} V_{i-1}(\widehat{C}_{i-1}) &= \inf_{A_i} V_i\big(C_i(A_i, \widehat{C}_{i-1})\big) \\ &\geq \inf_{A_i} \Phi_i^{\text{low}}\big(C_i(A_i, \widehat{C}_{i-1})\big) \\ &= \Phi_i^{\text{low}}(C_i^{\text{low}}). \end{aligned} \tag{21}$$

Recall the original relaxed objective, $\eta_i \geq 0$,

$$\Phi_i^{\text{comb}}(C_i) := (1-\eta_i)\Phi_i^{\text{low}}(C_i) + \eta_i s_i \|C_i\|_2, \tag{22}$$

The i-th FLIP subproblem minimizes the balanced relaxed objective

$$\Phi_i(C_i) := (1-\eta_i)\Phi_i^{\text{low}}(C_i) + \eta_i \widetilde{s}_i \|C_i\|_2, \tag{23}$$

Since $\widehat{C}_i$ minimizes Φ_i and C_i^{low} minimizes Φ_i^{low} over the same feasible set, we have

$$\begin{aligned} \Phi_i(\widehat{C}_i) &\leq \Phi_i(C_i^{\text{low}}), \\ \Phi_i^{\text{low}}(C_i^{\text{low}}) &\leq \Phi_i^{\text{low}}(\widehat{C}_i). \end{aligned} \tag{24}$$

Subtracting the two inequalities yields:

$$\eta_i \widetilde{s}_i \|\widehat{C}_i\|_2 \leq \eta_i \widetilde{s}_i \|C_i^{\text{low}}\|_2 \tag{25}$$

For $\eta_i > 0$, the condition $\Omega_i \neq 0$ implies $\widetilde{s}_i = \|\Omega_i\|_2^2 > 0$, and hence $\|\widehat{C}_i\|_2 \leq \|C_i^{\text{low}}\|_2$. Since $s_i \geq \widetilde{s}_i$, it follows that

$$\begin{aligned} &\Phi_i^{\text{comb}}(\widehat{C}_i) - \Phi_i^{\text{comb}}(C_i^{\text{low}}) \\ =&\Phi_i(\widehat{C}_i) - \Phi_i(C_i^{\text{low}}) \\ &+ \eta_i(s_i - \widetilde{s}_i)\left(\|\widehat{C}_i\|_2 - \|C_i^{\text{low}}\|_2\right) \leq 0. \end{aligned} \tag{26}$$

For $\eta_i = 0$, the inequality (26) holds directly since $\Phi_i^{\text{comb}} = \Phi_i = \Phi_i^{\text{low}}$.

Combining these inequalities gives

$$\begin{aligned} \delta_i &= V_i(\widehat{C}_i) - V_{i-1}(\widehat{C}_{i-1}) \\ &\leq \Phi_i^{\text{up}}(\widehat{C}_i) - \Phi_i^{\text{low}}(C_i^{\text{low}}) \\ &= \Phi_i^{\text{comb}}(\widehat{C}_i) + (1-\gamma)\Delta_i(\widehat{C}_i) - \Phi_i^{\text{low}}(C_i^{\text{low}}) \\ &\leq \Phi_i^{\text{comb}}(C_i^{\text{low}}) + (1-\gamma)\Delta_i(\widehat{C}_i) - \Phi_i^{\text{low}}(C_i^{\text{low}}) \\ &= (1-\gamma)\Delta_i(\widehat{C}_i) + \gamma\Delta_i(C_i^{\text{low}}). \end{aligned} \tag{27}$$

Summing this inequality over $i = 1, \ldots, l$ and applying (19) completes the proof.

C.8 Proof of Theorem 6

C.8.1 Nonemptiness of the Feasible Set

We first verify that the feasible set is nonempty. For (11), the feasible set is given by $\{a \mid aI \succ B\}$. Since B is positive semidefinite, the constraint $aI \succ B$ requires $a > \lambda_{\max}$, where $\lambda_{\max}$ denotes the largest eigenvalue of B. Hence, the feasible set can be written as $\{a \mid a > \lambda_{\max}\}$. Since $\lambda_{\max}$ is finite, the feasible set is nonempty.

C.8.2 Attainability of the Optimal Solution

Recall that $B \succeq 0$ and $\lambda_{\max} > 0$. Since $\mathbf{u}_{\max}$ is a unit eigenvector of B associated with $\lambda_{\max}$, we have

$$C(a) = \frac{1}{4}\left(aI + B + B(aI-B)^{-1}B\right) = \frac{1}{4}V\left(a^2(aI-\Lambda)^{-1}\right)V^\top \succeq \frac{a^2}{4(a-\lambda_{\max})}\mathbf{u}_{\max}\mathbf{u}_{\max}^\top. \tag{28}$$

Consequently,

$$\begin{aligned} J(a) &\geq (1-\eta)\|\Omega C(a)\Omega^\top\|_2 \\ &\geq \frac{(1-\eta)a^2}{4(a-\lambda_{\max})}\|\Omega \mathbf{u}_{\max}\|_2^2. \end{aligned} \tag{29}$$

Since $\eta < 1$, $\lambda_{\max} > 0$, and $\Omega \mathbf{u}_{\max} \neq 0$, by (29), we have:

$$\lim_{a \to \lambda_{\max}^+} J(a) = +\infty. \tag{30}$$

Moreover,

$$J(a) \to +\infty \quad \text{as } a \to +\infty. \tag{31}$$

Since $J(a)$ is continuous on the feasible interval $(\lambda_{\max}, +\infty)$ and diverges at both endpoints (coercive), its minimum is attained at some $a^* > \lambda_{\max}$.

C.8.3 Convexity of $J(a)$ over the Feasible Set

Next, we prove the convexity of $J(a)$ defined in (11) over the feasible set $a > \lambda_{\max}$, whose form is given by

$$J(a) = (1-\eta)\|\Omega C(a)\Omega^\top\|_2 + \eta s\|C(a)\|_2.$$

We first consider the term $\|\Omega C(a)\Omega^\top\|_2$. By definition,

$$\|\Omega C(a)\Omega^\top\|_2 = \max_{\|\mathbf{v}\|_2=1} \mathbf{v}^\top \Omega C(a)\Omega^\top \mathbf{v} = \max_{\|\mathbf{v}\|_2=1} f_{\mathbf{v}}(a), \tag{32}$$

where we define $f_{\mathbf{v}}(a) := \mathbf{v}^\top \Omega C(a)\Omega^\top \mathbf{v}$. The following lemma establishes the convexity of the quadratic form.

Lemma 7. *For $C(a)$ defined in* (11)*, and for any fixed vector $\mathbf{u}$, the function*

$$f_{\mathbf{u}}(a) = \mathbf{u}^\top C(a)\mathbf{u}$$

is convex for $a > \lambda_{\max}$.

Proof. We first examine the expression of $C(a)$ in (11). Since B is positive semidefinite, it admits the orthogonal decomposition $B = V\Lambda V^\top$, where V is an orthogonal matrix satisfying (14), and Λ is a diagonal matrix containing the eigenvalues of B.

Using (14), we obtain

$$\begin{aligned} C(a) &= \frac{1}{4}aI + \frac{1}{4}B(aI-B)^{-1}B + \frac{1}{4}B \\ &= \frac{1}{4}aVV^\top + \frac{1}{4}V\Lambda V^\top \left(aVV^\top - V\Lambda V^\top\right)^{-1} V\Lambda V^\top + \frac{1}{4}V\Lambda V^\top \\ &= \frac{1}{4}aVV^\top + \frac{1}{4}V\Lambda(aI-\Lambda)^{-1}\Lambda V^\top + \frac{1}{4}V\Lambda V^\top \\ &= \frac{1}{4}V\left(aI + \Lambda^2(aI-\Lambda)^{-1} + \Lambda\right)V^\top. \end{aligned} \tag{33}$$

Here, the first equality follows from (11), the second uses the decomposition $B = V\Lambda V^\top$, and the third follows from (15).

Let $\mathbf{q} := V^\top \mathbf{u}$. Then,

$$\begin{aligned} f_{\mathbf{u}}(a) &= \mathbf{u}^\top C(a)\mathbf{u} \\ &= \frac{1}{4}\mathbf{q}^\top \left(aI + \Lambda^2(aI-\Lambda)^{-1} + \Lambda\right)\mathbf{q} \\ &= \frac{1}{4}\sum_j [\mathbf{q}]_j^2 \left(a + \frac{\lambda_j^2}{a-\lambda_j} + \lambda_j\right), \end{aligned} \tag{34}$$

where $\{\lambda_j\}$ are the eigenvalues of Λ.

Consider each scalar term

$$a + \frac{\lambda_j^2}{a - \lambda_j} + \lambda_j. \tag{35}$$

For $a > \lambda_{\max} \geq \lambda_j$, its second derivative is

$$\frac{2\lambda_j^2}{(a - \lambda_j)^3} > 0,$$

which implies that (35) is convex in a. Since $[\mathbf{q}]_j^2 \geq 0$, the function $f_{\mathbf{u}}(a)$ is a nonnegative weighted sum of convex functions, and is therefore convex for $a > \lambda_{\max}$. □

Applying Lemma 7 with $\mathbf{u} := \Omega^\top \mathbf{v}$, we conclude that $f_{\mathbf{v}}(a)$ in (32) is convex for $a > \lambda_{\max}$.

Next we are ready to analyze the convexity of $\|\Omega C(a_1)\Omega^\top\|_2$. Consider any two points $a_1, a_2 > \lambda_{\max}$ in the feasible region and any $0 \leq t \leq 1$. Let $\mathbf{v}$ be a unit eigenvector $\|\mathbf{v}\|_2 = 1$ corresponding to the largest eigenvalue of the matrix $\Omega C\big(ta_1 + (1-t)a_2\big)\Omega^\top$. Then,

$$\begin{aligned}
&\big\|\Omega C\big(ta_1 + (1-t)a_2\big)\Omega^\top\big\|_2 \\
&= \mathbf{v}^\top \Omega C\big(ta_1 + (1-t)a_2\big)\Omega^\top \mathbf{v} \\
&= f_{\Omega^\top \mathbf{v}}\big(ta_1 + (1-t)a_2\big) \\
&\leq t f_{\Omega^\top \mathbf{v}}(a_1) + (1-t) f_{\Omega^\top \mathbf{v}}(a_2) \\
&= t\, \mathbf{v}^\top \Omega C(a_1)\Omega^\top \mathbf{v} + (1-t)\, \mathbf{v}^\top \Omega C(a_2)\Omega^\top \mathbf{v} \\
&\leq t \max_{\|\mathbf{v}\|_2 = 1} \mathbf{v}^\top \Omega C(a_1)\Omega^\top \mathbf{v} + (1-t) \max_{\|\mathbf{v}\|_2 = 1} \mathbf{v}^\top \Omega C(a_2)\Omega^\top \mathbf{v} \\
&= t\|\Omega C(a_1)\Omega^\top\|_2 + (1-t)\|\Omega C(a_2)\Omega^\top\|_2,
\end{aligned}$$

where the first inequality follows from the convexity of $f_{\Omega^\top \mathbf{v}}$ established in Lemma 7. Therefore, $\|\Omega C(a)\Omega^\top\|_2$ is convex over the feasible region $a > \lambda_{\max}$.

For the second term $\|C(a)\|_2$, it can be regarded as a special case of $\|\Omega C(a)\Omega^\top\|_2$ with $\Omega = I$. Since the convexity of $\|\Omega C(a)\Omega^\top\|_2$ has already been established, it follows immediately that $\|C(a)\|_2$ is also convex for $a > \lambda_{\max}$.

Finally, since $1 - \eta \geq 0$ and $\eta s \geq 0$, the objective function $J(a)$ is a nonnegative weighted sum of two convex functions over the feasible set. Hence, $J(a)$ is convex, and the problem is therefore a convex optimization problem.

C.9 Proof of Proposition 1

We analyze the two terms in the objective function $J(a)$ separately. For the term $\eta s\|C(a)\|_2$, we have

$$\begin{aligned}
\|C(a)\|_2 &= \frac{1}{4}\left\|V\left(aI + \Lambda^2(aI - \Lambda)^{-1} + \Lambda\right)V^\top\right\|_2 \\
&= \frac{1}{4}\left\|aI + \Lambda^2(aI - \Lambda)^{-1} + \Lambda\right\|_2 \\
&= \frac{1}{4}\max_j \left(a + \frac{\lambda_j^2}{a - \lambda_j} + \lambda_j\right) \\
&= \frac{1}{4}\left(a + \frac{\lambda_{\max}^2}{a - \lambda_{\max}} + \lambda_{\max}\right),
\end{aligned}$$

where the first equality follows from (33), the second uses (14) and the invariance of the spectral norm under orthogonal transformations, the third holds because Λ is diagonal with eigenvalues $\{\lambda_j\}$, and the last equality follows from the monotonicity of the function $a + \frac{\lambda^2}{a-\lambda} + \lambda$ with respect to λ.

Taking derivatives, we obtain

$$\frac{\partial}{\partial a}\big(\eta s\|C(a)\|_2\big) = \frac{\eta s}{4}\left(1 - \frac{\lambda_{\max}^2}{(a - \lambda_{\max})^2}\right). \tag{36}$$

For the second term, note that

$$\|\Omega C(a)\Omega^\top\|_2 = \max_{\|\mathbf{v}\|_2=1} g(a,\mathbf{v}),$$

where $g(a,\mathbf{v}) := \mathbf{v}^\top \Omega C(a)\Omega^\top \mathbf{v}$. We first compute the derivative of $g(a,\mathbf{v})$. Let $\mathbf{q} := V^\top \Omega^\top \mathbf{v}$. Then

$$\begin{aligned}\frac{\partial g(a,\mathbf{v})}{\partial a} &= \frac{\partial}{\partial a}\mathbf{v}^\top \Omega C(a)\Omega^\top \mathbf{v} \\ &= \frac{1}{4}\frac{\partial}{\partial a}\sum_j q_j^2\left(a + \frac{\lambda_j^2}{a-\lambda_j} + \lambda_j\right) \\ &= \frac{1}{4}\sum_j q_j^2\left(1 - \frac{\lambda_j^2}{(a-\lambda_j)^2}\right),\end{aligned}$$

where the second equality follows from (34).

For a fixed a, let $\mathcal{E}_{\max}\big(\Omega C(a)\Omega^\top\big)$ denote the eigenspace corresponding to the largest eigenvalue of $\Omega C(a)\Omega^\top$. Then any unit vector $\mathbf{v} \in \mathcal{E}_{\max}\big(\Omega C(a)\Omega^\top\big)$ attains the maximum, i.e.,

$$\|\Omega C(a)\Omega^\top\|_2 = g(a,\mathbf{v}), \quad \mathbf{v} \in \mathcal{E}_{\max}\big(\Omega C(a)\Omega^\top\big),\ \|\mathbf{v}\|_2 = 1.$$

Since $g(a,\mathbf{v})$ is differentiable in a for every $\mathbf{v}$, and its derivative is continuous in $\mathbf{v}$, we can apply Danskin's theorem. This yields

$$\begin{aligned}\partial\big\|(1-\eta)\Omega C(a)\Omega^\top\big\|_2 &= (1-\eta)\,\partial\left(\max_{\|\mathbf{v}\|_2=1} g(a,\mathbf{v})\right) \\ &= (1-\eta)\,\mathrm{conv}\left\{\frac{\partial g(a,\mathbf{v})}{\partial a}\ \Big|\ \mathbf{v} \in \mathcal{E}_{\max}\big(\Omega C(a)\Omega^\top\big),\ \|\mathbf{v}\|_2 = 1\right\} \\ &= \frac{1-\eta}{4}\,\mathrm{conv}\left\{\sum_j q_j^2\left(1 - \frac{\lambda_j^2}{(a-\lambda_j)^2}\right)\ \Big|\ \mathbf{q} = V^\top\Omega^\top\mathbf{v},\ \mathbf{v} \in \mathcal{E}_{\max},\ \|\mathbf{v}\|_2 = 1\right\}.\end{aligned} \tag{37}$$

Combining (36) and (37), we obtain the subdifferential of $J(a)$ as stated in (12). This completes the proof.

C.10 Proof of Proposition 2

By Theorem 6, the objective $J(a)$ is convex on the feasible region. Then, for any feasible a_1, a_2 and any $g_1 \in \partial J(a_1)$, we have

$$J(a_2) \geq J(a_1) + g_1(a_2 - a_1).$$

Similarly, for any $g_2 \in \partial J(a_2)$,

$$J(a_1) \geq J(a_2) + g_2(a_1 - a_2).$$

Adding the two inequalities yields

$$J(a_1) + J(a_2) \geq J(a_1) + J(a_2) + g_1(a_2 - a_1) + g_2(a_1 - a_2),$$

which simplifies to

$$(g_1 - g_2)(a_1 - a_2) \geq 0.$$

This completes the proof.

C.11 Proof of Theorem 7

C.11.1 Proof of $g \leq 0$ for $g \in \partial J(a_{\mathrm{low}})$

Let $\mathbf{u} \in \mathcal{E}_{\max}\left(\Omega C(a)\Omega^\top\right)$ with $\|\mathbf{u}\|_2 = 1$, and define $\mathbf{q} = V^\top\Omega^\top\mathbf{u}$. We have

$$\sum_j [\mathbf{q}]_j^2\left(1 - \frac{\lambda_j^2}{(a-\lambda_j)^2}\right) \leq \sum_j [\mathbf{q}]_j^2 \leq \mathbf{q}^\top\mathbf{q} = \mathbf{u}^\top\Omega V V^\top\Omega^\top\mathbf{u} \leq \mathbf{u}^\top\Omega\Omega^\top\mathbf{u} \leq \max_{\|\mathbf{u}\|_2=1}\mathbf{u}^\top\Omega\Omega^\top\mathbf{u} = \|\Omega\|_2^2,$$

where the third inequality follows from (14).

Using the expression for $J(a)$ in (12), we obtain for any $g \in \partial J(a)$, it holds

$$g \leq \frac{\eta s}{4}\left(1 - \frac{\lambda_{\max}^2}{(a-\lambda_{\max})^2}\right) + \frac{1-\eta}{4}\|\Omega\|_2^2.$$

Since the neural network is non-zero, we have $s > 0$. Additionally, η satisfies $\eta = \gamma\big(1-\big(\frac{1}{4}\big)^{l-i}\big) < 1$. Next, consider $\eta > 0$. Letting

$$a = \lambda_{\max} + \frac{\lambda_{\max}}{\sqrt{1 + \frac{1-\eta}{\eta s}\|\Omega\|_2^2}},$$

and substituting this value of a into the inequality for g, we obtain

$$\begin{aligned}
&\frac{\eta s}{4}\left(1 - \frac{\lambda_{\max}^2}{(a-\lambda_{\max})^2}\right) + \frac{1-\eta}{4}\|\Omega\|_2^2 \\
=&\frac{\eta s}{4}\left(1 - \left(1 + \frac{1-\eta}{\eta s}\|\Omega\|_2^2\right)\right) + \frac{1-\eta}{4}\|\Omega\|_2^2 \\
=&-\frac{1-\eta}{4}\|\Omega\|_2^2 + \frac{1-\eta}{4}\|\Omega\|_2^2 = 0.
\end{aligned}$$

Therefore, for $\eta > 0$ and the value of a as specified, we conclude that for any $g \in \partial J(a)$, it holds

$$g \leq \frac{\eta s}{4}\left(1 - \frac{\lambda_{\max}^2}{(a-\lambda_{\max})^2}\right) + \frac{1-\eta}{4}\|\Omega\|_2^2 = 0.$$

C.11.2 Proof of $g \geq 0$ for $g \in \partial J(a_{\mathrm{up}})$

Substituting $a = 2\lambda_{\max}$ into the first term of the subdifferential of $J(a)$ in (12), we have

$$\frac{\eta s}{4}\left(1 - \frac{\lambda_{\max}^2}{(2\lambda_{\max}-\lambda_{\max})^2}\right) = \frac{\eta s}{4}(1-1) = 0.$$

Now, consider any element inside the convex hull of the second term in (12). If $a = 2\lambda_{\max}$, for each j, we have

$$1 - \frac{\lambda_j^2}{(2\lambda_{\max}-\lambda_j)^2} = \frac{(2\lambda_{\max}-\lambda_j)^2 - \lambda_j^2}{(2\lambda_{\max}-\lambda_j)^2} = \frac{4\lambda_{\max}(\lambda_{\max}-\lambda_j)}{(2\lambda_{\max}-\lambda_j)^2} \geq 0.$$

Moreover, as $q_j^2 \geq 0$ for all j we obtain

$$\sum_j q_j^2\left(1 - \frac{\lambda_j^2}{(2\lambda_{\max}-\lambda_j)^2}\right) \geq 0.$$

Hence, using equation (12), for any $g \in \partial J(2\lambda_{\max})$, we conclude that $g \geq 0$. This completes the proof.

D Experimental Resources

Main experiments were conducted on a machine equipped with an Intel(R) Xeon(R) Gold 6530 CPU, using 8 available CPU cores and 62 GB system memory, together with an NVIDIA GeForce RTX 4090 GPU with 24 GB GPU memory. LipDiff was executed on the GPU, while all other methods were executed on the CPU. The total runtime over all methods and network depths was $74{,}841.87$ seconds (20.79 hours) for the random medium networks (Fig. 3), $387{,}573.71$ seconds (107.66 hours) for the random large networks (Fig. 4), and $345{,}794.97$ seconds (96.05 hours) for the trained networks (Fig. 5). In addition, training the networks on SVHN required another 155 minutes, bringing the total computation time for the trained-network experiments to $355{,}094.97$ seconds (98.64 hours). The additional sanity checks required 230,948.33 seconds (64.15 hours) running LipMIP. Overall, the aggregate runtime across all experiments was $1{,}048{,}458.88$ seconds (291.24 hours).

E Broader Impacts

This paper is primarily a theoretical and methodological study of fast Lipschitz estimation for feedforward neural networks. It has no direct societal impact.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction accurately reflect the contributions and scope of the paper. The claimed methodological contributions and empirical findings are supported by the theoretical discussion and experimental results, and the paper states the relevant assumptions and limitations.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses the scope and limitations of the current work, noting that FLIP is presently developed and evaluated for classical feedforward networks with linear and activation layers. Extending the approach to broader architectures such as CNNs and ResNets is identified as future work.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: All theoretical results state their assumptions explicitly, and complete proofs are provided in the appendix. The main paper includes proof sketches where appropriate.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: All experimental details are provided in the Experiments section and in Appendix D, and the code and data required to reproduce the results are included with the submission.

   Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: All the code and data are submitted along with the paper, and will be publicly released after publication.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper details the experimental settings and benchmarks in the Experiments section and in Appendix D.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: The proposed Lipschitz-bound computation is deterministic: for a fixed neural network, FLIP produces the same estimate up to numerical precision. Therefore, the main reported bound comparisons do not rely on stochastic sampling variability, and deviations are limited to numerical accuracy.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We report the compute resources used for the experiments in the Experiments section and in Appendix D, including hardware configurations and runtime information.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and ensured that the research conducted in this paper conforms to these regulations.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: This paper is primarily a theoretical and methodological work on fast Lipschitz estimation for feedforward neural networks, and does not have direct societal impacts.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: This paper does not pose any such risk.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All existing assets used in this paper, including the public SVHN dataset and the compared baseline methods, are publicly available, and we cite their original sources appropriately. We use these assets only for standard experimental evaluation and comparison, respecting their intended research use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We provide the relevant implementation and experimental materials as part of the submission. These assets will be publicly released after publication.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing, user studies, participant annotation, or research with human subjects.

    Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core ideas, theoretical results, algorithms, experiments, and conclusions of this paper are original work by the authors. LLMs were used only for minor language polishing and typo correction.

Guidelines:

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