Lipschitz Continuity in Deep Learning: A Systematic Review of Theoretical Foundations, Estimation Methods, Regularization Approaches and Certifiable Robustness

Anonymous authors
Paper under double-blind review

Abstract

Lipschitz continuity is a fundamental property of neural networks that characterizes their sensitivity to input perturbations. It plays a pivotal role in deep learning, governing **robustness**, **generalization** and **optimization dynamics**. Despite its importance, research on Lipschitz continuity is scattered across various domains, lacking a unified perspective. This paper addresses this gap by providing a systematic review of Lipschitz continuity in deep learning. We explore its **theoretical foundations**, **estimation methods**, **regularization approaches**, and **certifiable robustness**. By reviewing existing research through the lens of Lipschitz continuity, this survey serves as a comprehensive reference for researchers and practitioners seeking a deeper understanding of Lipschitz continuity and its implications in deep learning. Code: `https://anonymous.4open.science/r/lipschitz_survey-DECE/`

1 Introduction

Deep learning has achieved remarkable success across various domains, including computer vision, natural language processing, and graph-based learning. Advances in state-of-the-art architectures, such as convolutional neural networks (CNNs) (Krizhevsky et al., 2012), transformers (Vaswani et al., 2017; Brown et al., 2020), and graph neural networks (GNNs) (Kipf & Welling, 2017), as well as their large-scale applications, including large language models (LLMs) (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023) and large vision-language models (LVLMs), have significantly expanded the frontiers of deep learning, enabling powerful capabilities across a wide range of tasks.

Beyond their remarkable success, ensuring safety *e.g.*, (Amodei et al., 2016; Abrecht et al., 2023), robustness (Madry et al., 2018) and generalization (Zhang et al., 2021), remains a critical challenge. These properties are intrinsically linked to the sensitivity of neural networks to input perturbations (Madry et al., 2018; Tsipras et al., 2019; Luo et al., 2024; 2025a), which can significantly impact their reliability in real-world applications. For instance, adversarial vulnerabilities (Goodfellow et al., 2015; Carlini & Wagner, 2017), and out-of-distribution (OOD) generalization (Hendrycks et al., 2021) are major concerns in developing reliable deep learning models. A fundamental mathematical concept that characterizes network sensitivity and governs these properties is **Lipschitz continuity**, which measures the upper bound of the outputs of neural networks to inputs.

Although numerous studies have explored Lipschitz continuity from both theoretical (Bartlett et al., 2017; Gouk et al., 2021) and applied (Arjovsky et al., 2017; Gulrajani et al., 2017; Miyato et al., 2018) perspectives, a unified survey that consolidates these insights is still lacking. Existing works primarily focus on isolated aspects, such as adversarial robustness (Szegedy et al., 2014; Madry et al., 2018; Weng et al., 2018b), regularization (Arjovsky et al., 2017; Gulrajani et al., 2017), or theoretical bounds without practical considerations (Bartlett et al., 2017). The absence of a comprehensive survey integrating these perspectives leaves a critical gap in the literature. To address this, we provide a systematic review of Lipschitz continuity in deep learning, covering its **theoretical foundations**, **estimation methods**, **regularization approaches in optimization and architecture**, and **certifiable robustness**. By synthesizing key findings across these domains,

this survey serves as a valuable resource for researchers and practitioners seeking a deeper understanding of Lipschitz continuity and its role in trustworthy learning systems.

1.1 Scope

This survey aims to consolidate existing knowledge across these domains, providing a comprehensive resource for researchers and practitioners seeking to understand and leverage Lipschitz continuity in deep learning. To ensure a high-quality review, we focus primarily on the literature published in prestigious venues, such as JMLR, TMLR, ICML, NeurIPS, ICLR, AAAI, and others. The scope of this survey is organized as below:

1. Section 2: Theoretical Foundations presents the formal definitions and fundamental properties of Lipschitz continuity, followed by key theoretical developments in deep learning. It covers the mathematical foundations, Lipschitz properties of activation functions and neural networks, Lipschitz analysis of multi-head self-attention, complexity-theoretic generalization bounds, and advances in training dynamics.
2. Section 3: Estimation Methods surveys the principal methods for estimating the Lipschitz constants of neural networks, including derivative bound propagation, spectral alignment, convex optimization relaxations, and relaxed integer programming approaches.
3. Section 4: Regularization Approaches reviews approaches for enforcing Lipschitz continuity in neural networks through both optimization-based and architectural approaches, covering weight regularization, gradient regularization, activation regularization, and class-margin regularization.
4. Section 5: Certifiable Robustness examines methods for achieving certifiable robustness through Lipschitz bounds, including the theoretical foundations of Lipschitz-margin robustness radii, certification via global Lipschitz bounds, certification via local Lipschitz bounds and certificated robustness in LLMs.

1.2 Contributions

Our key contributions are as follows:

1. **A Unified Systematic Review**. We close a critical gap in the literature by conducting a comprehensive survey of Lipschitz continuity in deep learning. Our review unites disparate research threads — spanning theoretical foundations, estimation methods, regularization approaches, architectural designs, and certified robustness — into a coherent framework that highlights their interconnections from the lens of Lipschitz continuity.
2. **Comprehensive Analysis**. We provide in-depth examinations of existing techniques organized into four categories:
   - **Theoretical Foundations**. We review the formal definitions and properties of Lipschitz continuity, analyze Lipschitz behavior of activations and network architectures, and synthesize key results on generalization bounds, optimization convergence, and training-induced Lipschitz dynamics.
   - **Estimation Methods**. From power-iteration spectral-norm approximations and extreme-value theory to randomized and interval-analysis approaches, we compare each method's tightness, computational overhead, and scalability to modern architectures.
   - **Regularization Approaches**. We critically assess gradient-based penalties, spectral normalization, orthogonal and Parseval constraints, and novel Lipschitz-bounded activations, examining their theoretical guarantees, implementation trade-offs, and empirical effects on robustness, generalization, and training dynamics.
   - **Certifiable Robustness**. We survey Lipschitz-based certification frameworks — covering global and local Lipschitz bounds, path-based and convex-relaxation certificates, randomized smoothing, and Lipschitz-constrained architectures — analyzing their provable guarantees, computational requirements, and applicability to real-world deep models.

3. **Results and Proofs**. We contribute new and complementary results, including corrected Lipschitz constants for common activation functions (validated by our experiments) and a Lipschitz continuity bound for arbitrary neural networks represented as directed acyclic graphs (DAGs). Furthermore, we provide theoretical results missing from prior work, such as rigorous proofs of the closed-form expressions for the Lipschitz constants of common activation functions and a general perturbation bound for certifiable robustness based on Hölder's conjugacy relation.

2 Theoretical Foundations

Lipschitz continuity is a foundational concept in deep learning theory, offering a rigorous framework for quantifying a neural network's sensitivity to input perturbations. It plays a central role in understanding robustness, generalization, and optimization dynamics.

2.1 Preliminaries

Definition 2.1 (p-Norm in Finite-Dimensional Vector Spaces)**.** Let $V \subseteq \mathbb{R}^d$ be a vector space. The p-norm (*i.e.* ℓ_p-norm) in V is defined as:

$$\|z\|_p = \begin{cases} \left(\sum_{i=1}^{d} |z_i|^p\right)^{1/p}, & \text{if } 1 \leq p < \infty, \\ \\ \max_{i=1,\ldots,d} |z_i|, & \text{if } p = \infty, \end{cases} \tag{1}$$

where $z = (z_1, z_2, \ldots, z_d) \in V$ (Rudin, 1976; Tao, 2006; Yosida, 2012).

Definition 2.2 (Globally K-Lipschitz Continuous)**.** Let $f : X \mapsto Y$ be a function, where $X \subseteq \mathbb{R}^d$ and $Y \subseteq \mathbb{R}^c$. The function f is said to be *globally K-Lipschitz continuous* (with respect to the 2-norm) if there exists a constant scalar $K > 0$ such that:

$$\|f(u) - f(v)\|_2 \leq K\|u - v\|_2, \quad \forall\, u, v \in X, \tag{2}$$

(Rudin, 1976; Tao, 2006; Yosida, 2012; Luo et al., 2025a).

Definition 2.3 (Globally $K_{p,q}$-Lipschitz Continuous)**.** Let:

$$1 \leq p, q \leq \infty \tag{3}$$

satisfy the Hölder conjugacy relation:

$$\frac{1}{p} + \frac{1}{q} = 1. \tag{4}$$

A function $f : X \to Y$, with $X \subseteq \mathbb{R}^d$ and $Y \subseteq \mathbb{R}^c$, is said to be *$K_{p,q}$-Lipschitz continuous* if there exists a constant $K_{p,q} > 0$ such that:

$$\|f(u) - f(v)\|_q \;\leq\; K_{p,q}\,\|u - v\|_p, \quad \forall\, u, v \in X, \tag{5}$$

where $\|\cdot\|_p$ and $\|\cdot\|_q$ denote the ℓ_p and ℓ_q norms, respectively.

Lemma 2.4 (Lipschitz Constant Bounds Gradient Norm)**.** *Let $f : \mathbb{R}^m \to \mathbb{R}$ be a differentiable function. Then f is K-Lipschitz continuous (with respect to the 2-norm) if and only if*

$$K = \sup_{x \in \mathrm{dom}(f)} \|\nabla f(x)\|_2, \tag{6}$$

Rudin (1976); Federer (2014). In the case $0 < K < 1$, the function f is called a contraction map.

2.2 Lipschitz continuity of activation functions

The Lipschitz constants of commonly used activation functions are provided in Table 1. The constants are derived as the supremum of the 2-norm of the derivatives of activation functions. Proofs are provided in the Appendix A. To ensure the correctness of the results, we have conducted numerical experiments by using Lemma 2.4 to verify the results.

Notes. The Lipschitz constant for the *softmax* function reported in the literature (Gouk et al., 2021) as 1 is merely an upper bound. We show in Appendix A.6 that the exact Lipschitz constant of *softmax* is indeed $\frac{1}{2}$, verified by our numerical experiment. For *Leaky ReLU* and *ELU*, the parameter α is typically chosen with $\alpha < 1$, so their effective Lipschitz constants are usually 1 in practice.

Auto-Differentiation based Numerical Method. Suppose $f : \mathbb{R}^m \to \mathbb{R}^n$ be a first- and second-order differentiable neural network. Leverage the autograd mechanism of pytorch, we use gradient descent — *e.g.*, Adam (Kingma & Ba, 2014) — to find the supremum:

$$\mathrm{Lip}[f] = \max_x \|f'(x)\|_2, \tag{7}$$

using Lemma 2.4.

Table 1: Lipschitz constants of activation functions.

Activation	**Definition**	**Lipschitz Constant & Proof**	**Citations**
ReLU	$\max(0, x)$	1	(Nair & Hinton, 2010; Virmaux & Scaman, 2018)
Leaky ReLU	$\max(\alpha x, x)$	$\max(1, \alpha)$	(Maas et al., 2013; Virmaux & Scaman, 2018)
Sigmoid	$\frac{1}{1+e^{-x}}$	$\frac{1}{4}$ (Appendix A.1)	(Cybenko, 1989; Gouk et al., 2021)
Tanh	$\tanh(x)$	1 (Appendix A.2)	(LeCun et al., 2002; Virmaux & Scaman, 2018)
Softplus	$\log(1 + e^x)$	1 (Appendix A.3)	(Dugas et al., 2001; Virmaux & Scaman, 2018)
ELU	$\begin{cases} x & x > 0 \\ \alpha(e^x - 1) & x \leq 0 \end{cases}$	$\max(1, \alpha)$	(Clevert et al., 2016; Gouk et al., 2021)
Swish	$x \cdot \sigma(x)$	≈ 1.1 (Appendix A.4)	(Ramachandran et al., 2017; Gouk et al., 2021)
GELU	$\frac{x}{2}\left(1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$	≈ 1.1 (Appendix A.5)	(Hendrycks & Gimpel, 2016)
Softmax	$\mathrm{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$	$\frac{1}{2}$ (Appendix A.6)	(Cisse et al., 2017; Virmaux & Scaman, 2018; Higham & Higham, 2019; Gouk et al., 2021)

2.3 Lipschitz continuity of self-attention

LLMs, built on transformer architectures (Vaswani et al., 2017; Brown et al., 2020), rely heavily on self-attention mechanisms (Vaswani et al., 2017), whose Lipschitz continuity governs their sensitivity to input perturbations, such as token-level adversarial attacks. Self-attention mechanisms, introduced by Vaswani et al. (2017), form the core of transformer architectures, enabling effective modeling of long-range dependencies in sequences for applications such as natural language processing and vision-language tasks.

Single-Head Self-Attention. For an input sequence matrix $x \in \mathbb{R}^{n \times d}$, where n is the sequence length and d is the embedding dimension, a self-attention layer computes the output as

$$\mathrm{Attention}(x) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V, \tag{8}$$

where

$$Q = xW_Q, \quad K = xW_K, \quad V = xW_V, \tag{9}$$

are the *query*, *key*, and *value* matrices, with weight matrices $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$, and the softmax is applied row-wise to normalize attention scores (Vaswani et al., 2017). The scaling factor $\frac{1}{\sqrt{d}}$ stabilizes gradients, but the non-linear softmax operation renders dot-product attention non-Lipschitz continuous, complicating the computation of the Lipschitz constant, Lip[Attention], which quantifies the layer's sensitivity to input perturbations and is critical for ensuring robustness and generalization in transformers.

Multi-Head Self-Attention. In practice, transformers employ h parallel self-attention heads to jointly attend to information from different representation subspaces (Vaswani et al., 2017). Each head i has its own projection matrices $W_{Q,i},\ W_{K,i},\ W_{V,i} \in \mathbb{R}^{d\times d_h}$, where $d_h = d/h$, and produces an output $\text{Attention}_i(x) \in \mathbb{R}^{n\times d_h}$. The outputs are concatenated and projected via an output matrix $W_O \in \mathbb{R}^{d\times d}$. While this architecture improves expressivity, its Lipschitz constant depends on the largest product of spectral norms across heads, scaled by $\sqrt{n}/\sqrt{d}$ (Kim et al., 2021), retaining the same $\sqrt{n}$ growth as the single-head case.

2.3.1 Locally Lipschitz Continuous

Standard dot-product self-attention is locally Lipschitz continuous, and several works have derived explicit upper bounds for its local Lipschitz constant. Hu et al. (2024) show that for the self-attention layer operating on $\text{vec}(X)$ within a ball $B_R(\text{vec}(X))$, the following holds:

$$\text{Lip}(\text{Attn}; B_R(\text{vec}(X))) \leq n(n+1)\left(\|X\|_F + R\right)^2 \left[\|W_V\|_2 \|W_Q\|_2 \|W_K\|_2 + \|W_V\|_2\right], \tag{10}$$

where n is the sequence length, R is the ball radius, $\|\cdot\|_F$ is the Frobenius norm, and $\|\cdot\|_2$ is the spectral norm.

Castin et al. (2023) provide a tighter bound in terms of n, showing that if $X \in B_R^n(0) \subset \mathbb{R}^d$, then:

$$\text{Lip}(\text{Attn}; B_R^n(0)) \leq \sqrt{3}\,\|W_V\|_2 \left(\|A\|_2^2 R^4(4n+1) + n\right)^{\frac{1}{2}}, \tag{11}$$

where A is the attention score matrix.

Most recently, Yudin et al. (2025) refine the bound by explicitly incorporating the Jacobian of the softmax function,

$$\mathcal{M}(P_{i,:}) = \text{diag}(P_{i,:}) - P_{i,:}^\top P_{i,:}, \tag{12}$$

leading to:

$$\|\mathcal{J}_{\text{Attn}}(X)\|_2 \leq \|W_V\|_2 \left(\|P\|_2 + 2\|X\|_2^2 \|A\|_2 \max_i \|\mathcal{M}(P_{i,:})\|_2\right). \tag{13}$$

This refined bound isolates the contribution of the attention-score distribution to the local Lipschitz constant, enabling targeted regularization strategies such as JaSMin (Jacobian Softmax norm Minimization) to directly control this term (Yudin et al., 2025).

2.3.2 Globally Non-Lipschitz Continuous

Standard dot-product self-attention is not globally Lipschitz continuous since the bounds depend on the length of input sequences (Vuckovic et al., 2020; Kim et al., 2021). Vuckovic et al. (2020) show that:

$$\text{Lip}[\text{Attention}] \sim O(\sqrt{d \log n}), \tag{14}$$

which is refined later by (Kim et al., 2021). Kim et al. (2021) derive an upper bound on the Lipschitz constant with respect to the ℓ^2-norm of the input:

$$\text{Lip}[\text{Attention}] \leq \frac{\sqrt{n}}{\sqrt{d/h}} \left(4\phi^{-1}(n-1)+1\right) \left(\sqrt{\sum_{i=1}^{h} \|W_{Q,h_i}\|_2^2 \|W_{V,h_i}\|_2^2}\right) \|W_{O,h_i}\|_2 \tag{15}$$

where:

$$\phi(x) := x \cdot \mathrm{e}^{x+1} \tag{16}$$

is an invertible univariate function on $x > 0$, and n is the input sequence length Kim et al. (2021).

2.4 Lipschitz continuity of neural networks

Definition 2.5 (Feedforward Network)**.** Let $f : \mathbb{R}^m \to \mathbb{R}^n$ denote a feedforward neural network consisting of L layers, defined as the composition:

$$f := h^{(L)} \circ \cdots \circ h^{(2)} \circ h^{(1)}, \tag{17}$$

where each layer $h^{(\ell)}$ is composed of a linear transformation $\phi^{(\ell)}(\cdot)$ followed by a non-linear activation function $\rho^{(\ell)}(\cdot)$:

$$h^{(\ell)} := \rho^{(\ell)} \circ \phi^{(\ell)}. \tag{18}$$

Proposition 2.6 (Lipschitz Constant of a Linear Layer Yoshida & Miyato (2017); Sedghi et al. (2019); Kim et al. (2021); Luo et al. (2025a))**.** *Let $\phi^{(\ell)} : \mathbb{R}^{m_\ell} \to \mathbb{R}^{n_\ell}$ denote a linear transformation implemented either as a convolutional layer:*

$$\phi^{(\ell)}(z^{(\ell-1)}) = \boldsymbol{\theta}^{(\ell)} \circledast z^{(\ell-1)} + b^{(\ell)}, \tag{19}$$

or as a fully connected (dense) layer:

$$\phi^{(\ell)}(z^{(\ell-1)}) = \boldsymbol{\theta}^{(\ell)} z^{(\ell-1)} + b^{(\ell)}, \tag{20}$$

where $\boldsymbol{\theta}^{(\ell)}$ is the weight matrix (or convolution kernel), $b^{(\ell)}$ is the bias term, $\circledast$ denotes the convolution operator, and $z^{(\ell-1)}$ is the output from the previous layer. The Lipschitz constant of $\phi^{(\ell)}$ with respect to the Euclidean norm is given by:

$$\mathrm{Lip}[\phi^{(\ell)}] = \|\boldsymbol{\theta}^{(\ell)}\|_2, \tag{21}$$

where $\|\boldsymbol{\theta}^{(\ell)}\|_2$ denotes the spectral norm of the associated linear operator.

Proposition 2.7 (Spectral Norm via Singular Value Decomposition Golub & Van Loan (2013); Strang (2009); Sedghi et al. (2019); Luo et al. (2025b))**.** *Let $\boldsymbol{\theta} \in \mathbb{R}^{n\times m}$ be a matrix with singular value decomposition (SVD):*

$$\boldsymbol{\theta} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top, \quad \textit{with} \quad \sigma_1 > \sigma_2 > \cdots > \sigma_r > 0, \quad r = \mathrm{rank}(\boldsymbol{\theta}), \tag{22}$$

*where σ_i are the singular values, and $\boldsymbol{u}_i$, $\boldsymbol{v}_i$ are the left and right singular vectors, respectively. Then, the spectral norm of $\boldsymbol{\theta}$ (*i.e.*, its operator norm induced by the Euclidean norm) is given by the largest singular value:*

$$\|\boldsymbol{\theta}\|_2 = \sigma_1. \tag{23}$$

Proposition 2.8 (Upper Bound on the Lipschitz Constant of a Neural Network (Yoshida & Miyato, 2017; Luo et al., 2025a))**.** *Let $f : \mathbb{R}^m \to \mathbb{R}^n$ be an L-layer feedforward neural network composed of linear transformations $\phi^{(\ell)}$ and activation functions $\rho^{(\ell)}$. The Lipschitz constant of f with respect to the Euclidean norm is defined as:*

$$\mathrm{Lip}[f] = \sup_{\forall\ u\neq v} \frac{\|f(u) - f(v)\|_2}{\|u - v\|_2}. \tag{24}$$

This quantity admits the following upper bound:

$$\mathrm{Lip}[f] \leq \prod_{\ell=1}^{L} \mathrm{Lip}[\rho^{(\ell)}] \prod_{\ell=1}^{L} \mathrm{Lip}[\phi^{(\ell)}]. \tag{25}$$

2.4.1 Lipschitz continuity for a DAG

To the best of our knowledge, the existing literature does not present an explicit Lipschitz bound for a general feedforward network with skip connections. Prior literature such as (Yoshida & Miyato, 2017; Gouk et al., 2021) focuses strictly on sequential architectures, while norm-based capacity measures such as the path norm (Neyshabur et al., 2015a) implicitly involve path enumeration but are not formulated as direct Lipschitz bounds. More recent path-metric bounds (Gonon, 2024) apply to arbitrary DAGs but are expressed in a rescaling-invariant metric form rather than the simple sum-over-paths structure we consider. An illustrative example is given in Figure 1, and we derive the explicit bound in Theorem 2.9, which does not appear in the literature.

Theorem 2.9 (Lipschitz Bound for Feedforward DAG)**.** *Let $G = (V, E)$ be a finite directed acyclic graph (DAG) representing a feedforward neural network with unique input node $s \in V$ and output node $t \in V$. Each node $v \in V$ is a module h_v that is Lipschitz continuous with constant* $\mathrm{Lip}[h_v]$*. An edge $(u \to v) \in E$ represents the computation of h_v immediately after h_u. Let $x_{v_i} : x \to x_{v_i}(x)$ represent the output at a node $v_i \in V$. The network is evaluated additively along incoming edges:*

$$x_s(x) = x, \tag{26}$$

$$x_v(x) = \sum_{(u \to v) \in E} h_v\big(x_u(x)\big), \quad v \neq s. \tag{27}$$

A computational path *p from s to t is an ordered sequence of nodes*

$$p = (v_0 = s, v_1, \ldots, v_{L_p} = t), \tag{28}$$

where $(v_{i-1} \to v_i) \in E$ for all $i = 1, \ldots, L_p$. Define for each edge $(u \to v)$ the constant

$$C_{(u \to v)} := \mathrm{Lip}[h_v], \tag{29}$$

and for a path p set

$$C_p := \prod_{i=1}^{L_p} \mathrm{Lip}[h_{v_i}] = \prod_{i=1}^{L_p} C_{(v_{i-1} \to v_i)}. \tag{30}$$

Let $\mathcal{P}$ be the set of all such paths from s to t. Then the Lipschitz constant of the overall network $f(x) := x_t(x)$ satisfies

$$\mathrm{Lip}[f] \;\leq\; \sum_{p \in \mathcal{P}} C_p. \tag{31}$$

Proof. Fix any topological order of V. For $v \in V$, define the scalar

$$S(v) := \begin{cases} 1, & v = s, \\ \displaystyle\sum_{(u \to v) \in E} C_{(u \to v)}\, S(u), & v \neq s. \end{cases} \tag{32}$$

Claim: for all $v \in V$,

$$\mathrm{Lip}[x_v] \;\leq\; S(v). \tag{33}$$

We prove equation 33 by induction along the topological order.

Base case ($v = s$). $x_s(x) = x$, hence $\mathrm{Lip}[x_s] = 1 = S(s)$.

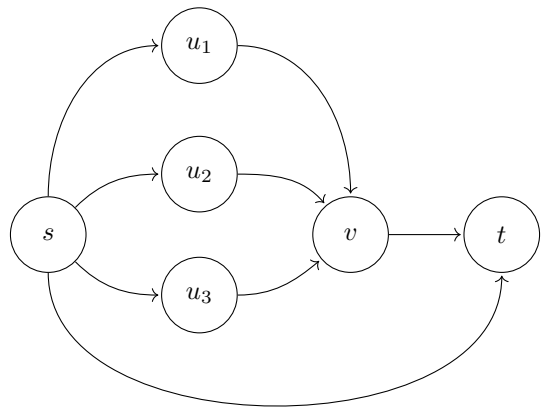

Figure 1: A simple feedforward directed acyclic graph (DAG) neural network. There are four computational paths: $s \to u_1 \to v \to t$, $s \to u_2 \to v \to t$, $s \to u_3 \to v \to t$, and $s \to t$. A node is a module in the DAG neural network f.

Inductive step. Let $v \neq s$ and assume equation 33 holds for all predecessors u of v. Applying Minkowski inequality, for any x, y in the input space,

$$\|x_v(x) - x_v(y)\| = \left\| \sum_{(u \to v)} h_v\big(x_u(x)\big) - \sum_{(u \to v)} h_v\big(x_u(y)\big) \right\| \tag{34}$$

$$\leq \sum_{(u \to v)} \left\| h_v\big(x_u(x)\big) - h_v\big(x_u(y)\big) \right\| \tag{35}$$

$$\leq \sum_{(u \to v)} \mathrm{Lip}[h_v] \left\| x_u(x) - x_u(y) \right\| \tag{36}$$

$$\leq \sum_{(u \to v)} \mathrm{Lip}[h_v]\, \mathrm{Lip}[x_u]\, \|x - y\| \tag{37}$$

$$= \sum_{(u \to v)} C_{(u \to v)}\, \mathrm{Lip}[x_u]\, \|x - y\| \tag{38}$$

$$\leq \left(\sum_{(u \to v)} C_{(u \to v)}\, S(u) \right) \|x - y\| = S(v)\, \|x - y\|, \tag{39}$$

since inductive hypothesis:

$$\mathrm{Lip}[x_u] \leq S(u). \tag{40}$$

Thus $\mathrm{Lip}[x_v] \leq S(v)$. In particular,

$$\mathrm{Lip}[f] = \mathrm{Lip}[x_t] \ \leq \ S(t). \tag{41}$$

It remains to show $S(t) = \sum_{p \in \mathcal{P}} C_p$. More generally:

Lemma (Path expansion of S). For every $v \in V$,

$$S(v) = \sum_{p \in \mathcal{P}(s \to v)} \prod_{e \in p} C_e, \tag{42}$$

where $\mathcal{P}(s \to v)$ is the set of all directed paths from s to v.

Proof of Lemma. Proceed again by induction in topological order. For $v = s$, $\mathcal{P}(s \to s)$ contains only the empty path with product 1, so equation 42 holds. Assume equation 42 holds for all predecessors u of v. Then

$$S(v) = \sum_{(u\to v)} C_{(u\to v)}\, S(u) = \sum_{(u\to v)} C_{(u\to v)} \left(\sum_{p\in\mathcal{P}(s\to u)} \prod_{e\in p} C_e \right) \tag{43}$$

$$= \sum_{(u\to v)} \sum_{p\in\mathcal{P}(s\to u)} \left(\prod_{e\in p} C_e \right) C_{(u\to v)}. \tag{44}$$

Concatenating each $p \in \mathcal{P}(s \to u)$ with the edge $(u \to v)$ yields a bijection onto $\mathcal{P}(s \to v)$, and multiplies the edge constants accordingly; thus equation 42 holds for v. □

Applying the lemma at $v = t$ gives $S(t) = \sum_{p\in\mathcal{P}} C_p$, and hence

$$\mathrm{Lip}[f] \;\le\; S(t) = \sum_{p\in\mathcal{P}} \prod_{i=1}^{L_p} \mathrm{Lip}[h_{v_i}]. \tag{45}$$

□

2.5 Complexity-Theoretic Generalization Bound

The generalization ability of a neural network is governed by Lipschitz continuity. A smaller Lipschitz constant enforces smoother mappings, reducing the complexity of the function class and mitigating overfitting, thereby enhancing robustness to noise and adversarial perturbations (Neyshabur et al., 2015b; Gouk et al., 2021).

2.5.1 Rademacher Complexity

Definition 2.10 (Empirical Rademacher Complexity (Mohri et al., 2018))**.** Let $\mathcal{H}$ be a class of functions mapping from $\mathcal{X}$ to $\mathbb{R}$. Given a sample $S = (x_1, \ldots, x_n) \in \mathcal{X}^n$, the *empirical Rademacher complexity* of $\mathcal{H}$ with respect to S is defined as

$$\widehat{\mathfrak{R}}_S(\mathcal{H}) := \mathbb{E}_\sigma \left[\sup_{h\in\mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(x_i) \right], \tag{46}$$

where $\sigma_1, \ldots, \sigma_n$ are independent *Rademacher variables*, *i.e.*, random variables uniformly distributed on $\{-1, +1\}$. The (expected) *Rademacher complexity* is the expectation of $\widehat{\mathfrak{R}}_S(\mathcal{H})$ over all samples S of size n:

$$\mathfrak{R}_n(\mathcal{H}) := \mathbb{E}_S \left[\widehat{\mathfrak{R}}_S(\mathcal{H}) \right]. \tag{47}$$

2.5.2 Empirical Risk Bound – Binary-Class Classification

Theorem 2.11 (Rademacher Complexity Bounds – Binary Classification (Mohri et al., 2018))**.** *Let $\mathcal{H}$ be a hypothesis class mapping from $\mathcal{X}$ to $[0, 1]$. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over the choice of an i.i.d. sample of size n,*

$$\forall h \in \mathcal{H}, \quad R(h) \;\le\; \hat{R}(h) \;+\; 2\,\widehat{\mathfrak{R}}_n(\mathcal{H}) \;+\; 3\sqrt{\frac{\log(2/\delta)}{2n}}, \tag{48}$$

where $R(h)$ is the true risk, $\hat{R}(h)$ the empirical risk, and $\widehat{\mathfrak{R}}_n(\mathcal{H})$ the empirical Rademacher complexity of $\mathcal{H}$.

Lemma 2.12 (Empirical Rademacher Complexity of ℓ_2-Bounded Linear Functions (Shalev-Shwartz & Ben–David, 2014))**.** *Let $S = \{x_1, \ldots, x_n\} \subset \mathbb{R}^m$ with $\|x_i\|_2 \le B$ for all i. For $\mathcal{F} = \{x \mapsto w^\top x : \|w\|_2 \le 1\}$,*

$$\widehat{\mathfrak{R}}_S(\mathcal{F}) \;\le\; \frac{1}{n}\, \mathbb{E}_\sigma \Big\| \sum_{i=1}^{n} \sigma_i x_i \Big\|_2 \;\le\; \frac{B}{\sqrt{n}}. \tag{49}$$

Lemma 2.13 (Talagrand's Contraction Lemma for Lipschitz Functions (Mohri et al., 2018))**.** *Let $\mathcal{F}$ be a class of functions mapping from $\mathcal{X}$ to $\mathbb{R}^m$, and let $\phi : \mathbb{R}^m \to \mathbb{R}$ be K-Lipschitz with respect to the Euclidean norm and satisfy $\phi(0) = 0$. Then,*

$$\widehat{\mathfrak{R}}_n(\phi \circ \mathcal{F}) \;\leq\; K\,\widehat{\mathfrak{R}}_n(\mathcal{F}). \tag{50}$$

In particular, if $\mathcal{H}$ is a class of scalar-valued functions with $\mathrm{Lip}[h] \leq K$ and $\|x_i\|_2 \leq B$ for all sample points, then

$$\widehat{\mathfrak{R}}_n(\mathcal{H}) \;\leq\; \frac{KB}{\sqrt{n}}. \tag{51}$$

Corollary 2.14 (Lipschitz Continuity Bounds Empirical Risk)**.** *Combining Theorem 2.11 and Lemma 2.13 for a K-Lipschitz hypothesis class with $\|x\|_2 \leq B$ yields:*

$$R(f) \;\leq\; \hat{R}(f) \;+\; \frac{2KB}{\sqrt{n}} \;+\; 3\sqrt{\frac{\log(2/\delta)}{2n}}. \tag{52}$$

Remark 2.15. Corollary 2.14 shows that, in the binary classification setting, the empirical risk bound for a function

$$f := \phi \circ \mathcal{H}, \quad \mathcal{H} : \mathbb{R}^n \to \mathbb{R}^m, \quad \phi : \mathbb{R}^m \to \mathbb{R}, \tag{53}$$

is directly controlled by the Lipschitz constant $K := \mathrm{Lip}[\mathcal{H}]$ of the feature mapping $\mathcal{H}$. In particular, the complexity term in the bound scales linearly with K, so reducing K (e.g., via spectral norm constraints or weight regularization) tightens the bound and can improve generalization.

2.5.3 Empirical Risk Bound – Multi-Class Classification

Theorem 2.16 (Margin Bound for Multi-Class Classification (Mohri et al., 2018))**.** *Let $\mathcal{Y} = \{1, \ldots, k\}$ be a finite label set and let $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}\times\mathcal{Y}}$ be a set of scoring functions, where each $h \in \mathcal{H}$ maps $(x, y) \in \mathcal{X} \times \mathcal{Y}$ to a real-valued score $h(x, y) \in \mathbb{R}$.*

Prediction Margin. *For a sample $S = \{(x_i, y_i)\}_{i=1}^m$ and $\rho > 0$, the* margin *of h at (x, y) is*

$$\rho_h(x, y) \;:=\; h(x, y) - \max_{y' \neq y} h(x, y'). \tag{54}$$

Empirical Margin Loss. *The* empirical ρ-margin loss *is*

$$\widehat{R}_{S,\rho}(h) \;:=\; \frac{1}{m}\sum_{i=1}^{m} \Phi_\rho\big(\rho_h(x_i, y_i)\big), \tag{55}$$

where Φ_ρ is the margin loss function *(Definition 5.5 in Mohri et al., 2018).*

Define the projection operator $\Pi_1(\mathcal{H})$ by

$$\Pi_1(\mathcal{H}) \;:=\; \{x \mapsto h(x, y) \;:\; y \in \mathcal{Y},\; h \in \mathcal{H}\}. \tag{56}$$

Claim. *Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the following bound holds for all $h \in \mathcal{H}$:*

$$R(h) \;\leq\; \widehat{R}_{S,\rho}(h) \;+\; \frac{4k}{\rho}\,\mathfrak{R}_m\big(\Pi_1(\mathcal{H})\big) \;+\; \sqrt{\frac{\log(1/\delta)}{2m}}, \tag{57}$$

where $\mathfrak{R}_m(\cdot)$ denotes the (expected) Rademacher complexity over samples of size m.

Corollary 2.17 (Lipschitz Continuity Bounds Multi-Class Margin)**.** *Assume that for every $h \in \mathcal{H}$ and $y \in \mathcal{Y}$, the map $x \mapsto h(x, y)$ is K-Lipschitz and $h(0, y) = 0$, and that $\|x\|_2 \leq B$ almost surely. Then*

$$\mathfrak{R}_m\big(\Pi_1(\mathcal{H})\big) \;\leq\; \frac{KB}{\sqrt{m}}, \tag{58}$$

and, applying Theorem 2.16,

$$R(h) \;\leq\; \widehat{R}_{S,\rho}(h) \;+\; \frac{4k\,KB}{\rho\sqrt{m}} \;+\; \sqrt{\frac{\log(1/\delta)}{2m}} \qquad \text{for all } h \in \mathcal{H}, \tag{59}$$

with probability at least $1 - \delta$ over the sample S.

Remark 2.18. Corollary 2.17 shows that, in the multi-class setting, the generalization gap is directly controlled by the Lipschitz constant K of the coordinate functions $x \mapsto h(x, y)$. Specifically, when each coordinate is K-Lipschitz and the inputs are bounded by B, the complexity term in the multi-class margin bound scales as $\frac{4kKB}{\rho\sqrt{m}}$. Thus, reducing K — for example, via spectral norm regularization or path-norm constraints — tightens the bound, leading to improved generalization guarantees.

2.6 Training Dynamics of Lipschitz Continuity

The first formal and mathematically rigorous analysis of how gradient-based training governs the evolution of a neural network's Lipschitz constant appears in two consecutive works (Luo et al., 2025b;a). In particular, Luo et al. (2025a) introduces the concept of *optimization-induced dynamics*, and together with Luo et al. (2025b) establishes the first continuous-time stochastic framework that explicitly links the optimization process to the time-varying Lipschitz continuity bound of deep networks.

Definition 2.19 (Luo et al., 2025a) integrates three key components — (i) the vectorized stochastic differential equation (SDE) representation of continuous-time SGD, (ii) a tight feed-forward Lipschitz bound, and (iii) an exact characterization of the operator norm of a linear unit — they derive a closed-form stochastic dynamical system governing the layer-wise and network-level evolution of Lipschitz constants under training.

Definition 2.19 (Stochastic Dynamical System of Lipschitz Continuity Bound (Luo et al., 2025a))**.** Consider an L-layer feed-forward neural network with parameter matrices $\boldsymbol{\theta}^{(\ell)}(t) \in \mathbb{R}^{m_\ell \times n_\ell}$ at training time t, and 1-Lipschitz activation functions. Let $\mathrm{vec}(\cdot)$ denote column-wise vectorization, $\|\cdot\|_{op}$ the matrix operator (spectral) norm, and $\mathcal{L}_f$ the training loss. The continuous-time dynamics of the parameters under stochastic gradient descent (SGD) with learning rate $\eta > 0$ are given by

$$\begin{cases} \mathrm{dvec}(\boldsymbol{\theta}^{(\ell)}) = -\mathrm{vec}\Big[\nabla^{(\ell)}\mathcal{L}_f(\boldsymbol{\theta}(t))\Big]\,\mathrm{d}t + \sqrt{\eta}\,[\boldsymbol{\Sigma}_t^{(\ell)}]^{1/2}\mathrm{d}\boldsymbol{B}_t^{(\ell)}, \\ K^{(\ell)}(t) = \|\boldsymbol{\theta}^{(\ell)}(t)\|_{op}, \quad Z(t) = \displaystyle\sum_{\ell=1}^{L} \log K^{(\ell)}(t), \quad K(t) = e^{Z(t)}, \end{cases}$$

where:

- $\boldsymbol{\theta}^{(\ell)}(t)$ represents the ℓ-th layer parameters.
- $\nabla^{(\ell)}\mathcal{L}_f(\boldsymbol{\theta}(t))$ is the gradient of the loss with respect to the ℓ-th layer parameters.
- $\boldsymbol{\Sigma}_t^{(\ell)} \in \mathbb{R}^{(m_\ell n_\ell)\times(m_\ell n_\ell)}$ is the layer-wise covariance matrix of stochastic gradient noise, arising from mini-batch sampling.
- $\boldsymbol{B}_t^{(\ell)}$ is a standard $(m_\ell n_\ell)$-dimensional Wiener process adapted to the filtration $\mathcal{F}_t$ representing the randomness of mini-batch sampling.
- $K^{(\ell)}(t)$ is the Lipschitz bound of layer ℓ, equal to the operator norm of $\boldsymbol{\theta}^{(\ell)}(t)$.
- $K(t)$ is the overall network Lipschitz bound, and $Z(t)$ is its logarithm for analytical convenience.

This stochastic dynamical system characterizes, in continuous time, the evolution of both layer-wise and network-level Lipschitz continuity bounds induced by the optimization process.

Based on the system of SDEs in Definition 2.19, Luo et al. (2025a) show that the layer-wise and network-level dynamics decompose into three *driving forces*:

1. **Optimization-induced drift**:

$$\mu^{(\ell)}(t) = \frac{\left\langle \boldsymbol{J}_{op}^{(\ell)}(t), -\mathrm{vec}\big[\nabla^{(\ell)}\mathcal{L}_f(\boldsymbol{\theta}(t))\big]\right\rangle}{\sigma_1^{(\ell)}(t)}, \tag{60}$$

   where $\boldsymbol{J}_{op}^{(\ell)}(t)$ is the Jacobian of the operator norm with respect to $\mathrm{vec}(\boldsymbol{\theta}^{(\ell)}(t))$ and $\sigma_1^{(\ell)}(t)$ is its largest singular value. This term captures the deterministic component of Lipschitz evolution driven by the mean gradient flow.

2. **Noise–curvature entropy production**:

$$\kappa^{(\ell)}(t) = \frac{\eta}{2\,\sigma_1^{(\ell)}(t)} \left\langle \boldsymbol{H}_{op}^{(\ell)}(t), \boldsymbol{\Sigma}_t^{(\ell)} \right\rangle \geq 0, \tag{61}$$

   where $\boldsymbol{H}_{op}^{(\ell)}(t)$ is the Hessian of the operator norm and $\boldsymbol{\Sigma}_t^{(\ell)}$ is the gradient-noise covariance. This non-negative term quantifies irreversible growth of the Lipschitz bound due to the interaction between stochastic gradient noise and curvature.

3. **Diffusion-modulation intensity**:

$$\boldsymbol{\lambda}^{(\ell)}(t) = \frac{\sqrt{\eta}}{\sigma_1^{(\ell)}(t)} \left[\boldsymbol{\Sigma}_t^{(\ell)}\right]^{1/2\top} \boldsymbol{J}_{op}^{(\ell)}(t), \tag{62}$$

   which controls the variance of Lipschitz evolution by scaling the Wiener noise term from mini-batch sampling.

At the network level, these quantities aggregate as:

$$\mu_Z(t) = \sum_{\ell=1}^{L} \mu^{(\ell)}(t), \qquad \kappa_Z(t) = \sum_{\ell=1}^{L} \kappa^{(\ell)}(t), \qquad \lambda_Z(t) = \left[\sum_{\ell=1}^{L} \|\boldsymbol{\lambda}^{(\ell)}(t)\|_2^2\right]^{\frac{1}{2}}, \tag{63}$$

governing the deterministic trend, irreversible growth, and stochastic fluctuations of the overall Lipschitz bound $K(t)$.

3 Estimation Methods

Proposition 2.8 gives an upper bound of the Lipschitz constant of a network (Miyato et al., 2018; Luo et al., 2025a). However, exact computation of the Lipschitz constant

$$K = \sup_{x \neq y} \frac{\|f(x) - f(y)\|_2}{\|x - y\|_2} = \sup_x \|\nabla f(x)\|_2 \tag{64}$$

is generally NP-hard for ReLU networks (Virmaux & Scaman, 2018; Jordan & Dimakis, 2020). This section surveys major estimation and bounding techniques, ranging from statistical sampling to certified convex relaxations. We group them into *Extreme Value Theory*, *Derivative Bound Propagation*, *Spectral Alignment*, *Convex Optimization Relaxations*, and *Exact/Relaxed MILP formulations*.

3.1 Power Iteration for Single Linear Unit

Power Iteration (Miyato et al., 2018). Power iteration is the standard way to estimate the Lipschitz constant for a linear layer — fully-connected or convolutional, by approximating the spectral norm of weight matrices, which upper bounds the layer Lipschitz constant (Proposition 2.6); see Yoshida & Miyato (2017); Miyato et al. (2018); Sedghi et al. (2019); Kim et al. (2021); Luo et al. (2025a).

Suppose W is the parameter matrix of a fully-connected layer or convolutional layer, then:

$$\mathrm{Lip}[W] = \sigma_1(W), \tag{65}$$

where $\sigma_1(W)$ is the largest singular value of W (Luo et al., 2025a). Computing exact $\sigma_1(W)$ is intractable for the large matrix, it is approximated using a method called *power iteration* (see Algorithm 1). Starting with random unit vectors u_0, v_0, one step of power iteration updates

$$v_{k+1} \leftarrow \frac{W^\top u_k}{\|W^\top u_k\|_2}, \qquad u_{k+1} \leftarrow \frac{W v_k}{\|W v_k\|_2}, \tag{66}$$

and uses $\hat{\sigma}_{k+1}$ as a differentiable estimate of $\sigma_1(W)$. At the iteration number T, the estimated Lipschitz constant of W is approximated as:

$$\mathrm{Lip}[W] \approx u_T^\top W v_T. \tag{67}$$

Algorithm 1: Power Iteration of Computing $\|W\|_2$

Input: Matrix W, number of power steps T
$u \leftarrow$ random vector of same shape as column dimension of W;
$u \leftarrow \frac{u}{\|u\|_2}$;
$v \leftarrow$ random vector of same shape as row dimension of W;
$v \leftarrow \frac{v}{\|v\|_2}$;
for $t \leftarrow 1$ **to** T **do**
  $v \leftarrow W^\top u$;
  $v \leftarrow \frac{v}{\|v\|_2}$;
  $u \leftarrow W v$;
  $u \leftarrow \frac{u}{\|u\|_2}$;
Output: $u^\top W v$

3.2 Extreme Value Theory

CLEVER (Weng et al., 2018b). The *Cross-Lipschitz Extreme Value for nEtwork Robustness* (CLEVER) method estimates a *local* Lipschitz constant by sampling gradient norms in a neighborhood of the input (Weng et al., 2018b).

Given a classifier $f : \mathbb{R}^d \to \mathbb{R}^K$, input x_0, predicted class $c = \arg\max_k f_k(x_0)$, and any target $j \neq c$, define $g_j(x) = f_c(x) - f_j(x)$. For a radius $R > 0$, p and q are conjugate:

$$\frac{1}{p} + \frac{1}{q} = 1, \tag{68}$$

the local cross-Lipschitz constant is

$$L^j_{q,x_0} = \max_{\|x - x_0\|_p \leq R} \|\nabla g_j(x)\|_q. \tag{69}$$

CLEVER estimates L^j_{q,x_0} by uniform sampling in the ball $\mathcal{B}_p(x_0, R)$: in each of N_b batches, draw N_s points $\{x^{(i)}\}$, compute $\|\nabla g_j(x^{(i)})\|_q$, and record the per-batch maximum. The collection of maxima is then fit to a *generalized extreme value* distribution — *extreme value theory* (Fisher-Tippett-Gnedenko Theorem) ensures that the maximum value of random variables can only follow one of the three extreme value distributions (Weng et al., 2018b). The fitted *location* parameter $\widehat{a}_W$ is taken as the estimator $\widehat{L}^j_{q,x_0}$ of the local Lipschitz constant.

3.3 Derivative Bound Propagation

Fast-Lip (Weng et al., 2018a). Fast-Lip (Weng et al., 2018a) computes a *certified local* Lipschitz upper bound for the margin

$$g(x) = f_c(x) - f_j(x) \tag{70}$$

over the ℓ_p ball $B_p(x_0, \varepsilon)$ by bounding each coordinate of $\nabla g(x)$ and aggregating these bounds in the dual norm q defined by

$$\frac{1}{p} + \frac{1}{q} = 1. \tag{71}$$

A key concept in Fast-Lip is that of *interval bounds* on the pre-activation values of neurons. Let $z_r^{(\ell)}(x)$ denote the pre-activation of neuron r in layer ℓ for input x:

$$z_r^{(\ell)}(x) = W_r^{(\ell)} h^{(\ell-1)}(x) + b_r^{(\ell)}, \tag{72}$$

where $W_r^{(\ell)}$ is the r-th row of the weight matrix $W^{(\ell)}$, $b_r^{(\ell)}$ is the bias, and $h^{(\ell-1)}$ is the post-activation output from the previous layer. For a perturbation set $B_p(x_0, \varepsilon) = \{x : \|x - x_0\|_p \le \varepsilon\}$, *interval bounds* are constants $\ell_r^{(\ell)}$ and $u_r^{(\ell)}$ such that

$$\ell_r^{(\ell)} \;\le\; z_r^{(\ell)}(x) \;\le\; u_r^{(\ell)}, \quad \forall x \in B_p(x_0, \varepsilon). \tag{73}$$

These bounds are propagated layer-by-layer via *interval arithmetic*:

$$\ell^{(\ell)} = W_+^{(\ell)}\, \ell^{(\ell-1)} + W_-^{(\ell)}\, u^{(\ell-1)} + b^{(\ell)}, \tag{74}$$

$$u^{(\ell)} = W_+^{(\ell)}\, u^{(\ell-1)} + W_-^{(\ell)}\, \ell^{(\ell-1)} + b^{(\ell)}, \tag{75}$$

where W_+ and W_- denote the elementwise positive and negative parts of W, respectively.

If $\ell_r^{(\ell)} \ge 0$, neuron (ℓ, r) is *always active* (ReLU slope = 1); if $u_r^{(\ell)} \le 0$, it is *always inactive* (ReLU slope = 0); if $\ell_r^{(\ell)} < 0 < u_r^{(\ell)}$, it is *uncertain* (slope $\in [0, 1]$).

Fast-Lip (Weng et al., 2018a) computes a certified local Lipschitz bound for the margin

$$g(x) = f_c(x) - f_j(x) \tag{76}$$

over $B_p(x_0, \varepsilon)$ by bounding each coordinate of $\nabla g(x)$. The slope masks from the interval bounds fix ReLU derivatives for certain units and branch over $\{0, 1\}$ for uncertain ones. In the two-layer base case, this yields per-coordinate bounds $\max_{x \in B_p(x_0,\varepsilon)} |[\nabla g(x)]_k|$, which are aggregated in the dual norm q (with $\frac{1}{p} + \frac{1}{q} = 1$) to give:

$$\mathrm{Lip}_q[g; B_p(x_0, \varepsilon)] \;\le\; \left(\sum_{k=1}^{n_0} B_k^q \right)^{1/q}. \tag{77}$$

This procedure generalizes recursively to deeper ReLU networks.

3.4 Spectral Alignment

SeqLip (Virmaux & Scaman, 2018) The standard spectral norm bound $\prod_{\ell=1}^{L} \|W_\ell\|_2$ can be overly loose because it assumes perfect alignment between the top singular vectors of consecutive layers. SeqLip improves this by computing the cosine similarity between the right singular vector $v_{\ell,1}$ of W_ℓ and the left singular vector $u_{\ell+1,1}$ of $W_{\ell+1}$:

$$\cos\theta_\ell = |v_{\ell,1}^\top u_{\ell+1,1}|. \tag{78}$$

The refined bound is:

$$\mathrm{Lip}_2[f] \le \left(\prod_{\ell=1}^{L} \|W_\ell\|_2 \right) \left(\prod_{\ell=1}^{L-1} \cos\theta_\ell \right). \tag{79}$$

This is still certified and can be much tighter in practice.

3.5 Convex Optimization Relaxation

LipSDP (Fazlyab et al., 2019) LipSDP formulates the problem of computing a certified global Lipschitz bound as a semidefinite program (SDP) via *incremental quadratic constraints* (IQCs) to encode slope restrictions of nonlinearities (*e.g.*, ReLU, tanh). The bound is obtained by solving:

$$\min_{\rho, T} \rho \quad \text{s.t.} \quad M(\rho, T) \preceq 0, \quad T \in \mathcal{T}_n, \tag{80}$$

where $M(\rho, T)$ is a symmetric block matrix encoding the network's linear layers, IQC multipliers for each nonlinearity, and the ρ-scaled identity term enforcing the Lipschitz bound, and $\mathcal{T}_n$ is the set of admissible IQC matrices satisfying the slope bounds (e.g., $0 \leq d_{ii} \leq 1$ for ReLU). Feasibility of $M(\rho, T) \preceq 0$ guarantees $\text{Lip}_2[f] \leq \rho$, making the bound certified and global. In practice, LipSDP often yields tighter bounds than spectral-norm products or FastLip, at the cost of solving a moderately sized SDP.

3.6 Exact and Relaxed Mixed-Integer Programming

LipMIP (Jordan & Dimakis, 2020) LipMIP encodes the exact Lipschitz computation for ReLU networks as a mixed-integer linear program (MILP). Binary variables encode the on/off state of each ReLU, and the MILP objective maximizes $\|f(x) - f(x')\|_p$ subject to $\|x - x'\|_p \leq 1$. It returns the exact global Lipschitz constant but is limited to small networks due to combinatorial explosion.

$$\text{Lip}[f] = \max_{\|x-x'\|_p \leq 1} \|f(x) - f(x')\|_p \quad \text{s.t. MILP-encoded ReLU constraints.} \tag{81}$$

LipLP (Jordan & Dimakis, 2020) LipLP is a linear programming relaxation of LipMIP, where binary ReLU indicators are replaced with continuous variables in $[0, 1]$. This greatly improves scalability, producing a certified upper bound:

$$\text{Lip}[f] \leq \max_{\|x-x'\|_p \leq 1} \|f(x) - f(x')\|_p \quad \text{s.t. LP-relaxed ReLU constraints.} \tag{82}$$

Bounds are looser than LipMIP but computable for larger networks.

4 Regularization Approaches

A variety of techniques have been developed to explicitly or implicitly enforce Lipschitz continuity in deep neural networks. These approaches can be categorized into:

(i) Section 4.1: **Weight Regularization** — constraining or regularizing weight matrices during initialization and training;

(ii) Section 4.2: **Gradient Regularization** — penalizing or normalizing gradient norms during training;

(iii) Section 4.3: **Activation Regularization** — designing or constraining activation functions to be Lipschitz-bounded;

(iv) Section 4.4: **Class-Margin Regularization** — implicitly enforcing Lipschitz constraints via maximizing class decision boundaries;

(v) Section 4.5: **Architectural Regularization** — enforcing Lipschitz constraints by designing architectures, particularly for transformers.

4.1 Weight Regularization

This subsection reviews methods that control the Lipschitz constant by directly constraining or regularizing the network's weight parameters, either at initialization or during training.

4.1.1 Spectral Norm Regularization

Spectral norm regularization constrains the Lipschitz constant of a neural network by bounding the spectral norm of its weight matrices, which is the largest singular value of the matrix (Proposition 2.6). Miyato et al. (2018) propose normalizing the weight matrix W of a linear layer by (Miyato et al., 2018):

$$W \leftarrow \frac{W}{\|W\|_2}, \tag{83}$$

ensuring a Lipschitz constant of at most 1 for the layer.

This method, widely adopted in generative adversarial networks (GANs) (Goodfellow et al., 2020), stabilizes training by preventing gradient explosion and enhances robustness to adversarial perturbations (Miyato et al., 2018). Yoshida & Miyato (2017) extend this to deep networks, showing that spectral normalization scales effectively across layers, though computing the spectral norm via singular value decomposition is computationally expensive.

4.1.2 Weight Clipping

Weight clipping is a straightforward method to enforce Lipschitz continuity by limiting the magnitude of weight matrices. Arjovsky et al. (2017) propose clipping weights to a fixed range (Arjovsky et al., 2017):

$$W \leftarrow \text{clip}(W, -\epsilon, +\epsilon), \tag{84}$$

where ϵ is a small positive constant. This bounds the spectral norm, ensuring a finite Lipschitz constant, and is widely used in Wasserstein GANs to enforce 1-Lipschitz discriminators (Arjovsky et al., 2017). Weight clipping is computationally efficient but can lead to vanishing gradients if ϵ is too small, reducing model capacity (Gulrajani et al., 2017).

4.1.3 Orthogonal Weight

Parseval Networks (Cisse et al., 2017). Parseval networks enforce approximate orthonormality on weight matrices to bound their spectral norms (Cisse et al., 2017). For a weight matrix $W^{(\ell)}$ in layer ℓ, Cisse et al. (2017) minimize the Frobenius norm of the deviation from orthonormality:

$$\|W^{(\ell)\top} W^{(\ell)} - \mathrm{I}\|_F, \tag{85}$$

ensuring $\|W^{(\ell)}\|_2 \leq 1$ (Cisse et al., 2017, Sec. 3). For a feedforward network with L layers and 1-Lipschitz activations (*e.g.*, ReLU), the global Lipschitz constant is bounded by $K \leq \prod_{\ell=1}^{L} \|W^{(\ell)}\|_2 \leq 1$ (Proposition 2.8). Parseval training is efficient, and improves robustness in image classification tasks (Cisse et al., 2017). However, maintaining orthonormality can limit expressivity, and the bound may be loose for deep networks (Fazlyab et al., 2019).

Orthogonal and Unitary Layers (Lezcano-Casado & Martınez-Rubio, 2019). Orthogonal and unitary layers ensure $\|W^{(\ell)}\|_2 = 1$ by construction, using techniques like Householder reflections, Cayley transforms, or parameterized QR factorizations (Lezcano-Casado & Martınez-Rubio, 2019). Lezcano-Casado & Martınez-Rubio (2019) apply the method to RNNs with orthogonal recurrent weights, parameterized via exponential maps to maintain:

$$W^{(\ell)\top} W^{(\ell)} = \mathrm{I}, \tag{86}$$

guaranteeing $\|W^{(\ell)}\|_2 = 1$ (Lezcano-Casado & Martınez-Rubio, 2019). For an L-layer network with 1-Lipschitz activations, this yields $K \leq 1$.

4.2 Gradient Regularization

Gradient-based regularization methods enforce Lipschitz continuity by constraining the gradient norm of the loss function or network output with respect to inputs, ensuring smooth decision boundaries and robustness.

4.2.1 Direct Gradient Penalty

A straightforward way to constrain the Lipschitz constant of a differentiable network f_W — parameterized by W — is to directly penalize the norm of its input gradients, exploiting the fact that $\mathrm{Lip}[f_W] = \sup_x \|\nabla_x f_W(x)\|_2$. This approach augments the task loss with a regularization term of the form

$$\min_W \; \mathcal{L}_{\text{task}}(f_W) + \lambda \, \mathbb{E}_{x\sim\mathcal{D}} \left[\|\nabla_x f_W(x)\|_2\right], \tag{87}$$

where $\lambda > 0$ controls the trade-off between the task objective and Lipschitz control. Unlike offset-squared penalties such as WGAN-GP (Gulrajani et al., 2017), this formulation explicitly discourages large gradient norms, yielding tighter empirical Lipschitz bounds and improving robustness (Cisse et al., 2017).

4.2.2 Gradient Norm Normalization

Gradient penalty methods penalize large gradient norms to enforce approximate Lipschitz continuity. Gulrajani et al. (2017) introduce a gradient penalty for Wasserstein GANs, minimizing by:

$$\left[\|\nabla_x f(x)\|_2 - 1\right]^2, \tag{88}$$

where f is the discriminator and x is a sample. This ensures the network is approximately 1-Lipschitz, stabilizing training and improving generative quality (Gulrajani et al., 2017).

4.3 Activation Regularization

Activation regularization methods modulates Lipschitz continuity by constraining the properties of activation functions or their outputs. Such approaches may impose explicit norm bounds, design activations that are inherently norm-preserving, or apply penalties to activation magnitudes, thereby limiting the contribution of nonlinearities to the overall Lipschitz constant of the network.

4.3.1 Lipschitz-Bounded Activation

Anil et al. (2019) propose a provably 1-Lipschitz method with respect to the ℓ_2-norm while preserving universal approximation capability. Each linear layer W is constrained so that its spectral norm satisfies

$$\|W\|_2 \;=\; \sigma_{\max}(W) \;\leq\; 1, \tag{89}$$

ensuring the layer is 1-Lipschitz. These are interleaved with the *GroupSort* activation: given $x \in \mathbb{R}^d$ and a group size k dividing d, GroupSort partitions x into $\frac{d}{k}$ contiguous groups $(x_{(g)})_{g=1}^{d/k}$, each of size k, and sorts each $x_{(g)}$ in ascending order:

$$\mathrm{GroupSort}_k(x) \;=\; \big(\mathrm{sort}(x_{(1)}), \ldots, \mathrm{sort}(x_{(d/k)})\big). \tag{90}$$

Sorting corresponds to multiplying by a permutation matrix P, hence is an isometry for all ℓ_p norms:

$$\|Pv\|_p = \|v\|_p, \quad \|P\|_2 = 1. \tag{91}$$

Thus its Jacobian $J_{\mathrm{GroupSort}} = P$ satisfies

$$\|\nabla_x \mathrm{GroupSort}(x)\|_2 = 1, \tag{92}$$

so GroupSort is exactly 1-Lipschitz. Composing spectral-norm-constrained linear maps with GroupSort activations yields

$$\mathrm{Lip}[f] \;=\; \prod_{\ell=1}^{L} \mathrm{Lip}[h_\ell] \;\leq\; 1, \tag{93}$$

where h_ℓ are the individual layers.

4.3.2 Lipschitz-Preserving Activation

Lipschitz-preserving activations, such as LipSwish (CLipSwish) (Perugachi-Diaz et al., 2021), enforce a unit Lipschitz constant by construction. In Perugachi-Diaz et al. (2021), the (scalar) LipSwish is defined as:

$$\mathrm{LipSwish}(x) \;=\; \frac{x\,\sigma(\beta x)}{1.1}, \tag{94}$$

where σ is the sigmoid and typically $\beta = 1$. The factor 1.1 upper-bounds $\sup_x \left|\frac{\mathrm{d}}{\mathrm{d}x}\, x\sigma(\beta x)\right|$, hence $\sup_x |\mathrm{LipSwish}'(x)| \le 1$. For the concatenated 1D map

$$\Phi(x) \;=\; \begin{bmatrix} \mathrm{LipSwish}(x) \\ \mathrm{LipSwish}(-x) \end{bmatrix}, \qquad J_\Phi(x) \;=\; \begin{bmatrix} \frac{\partial\,\mathrm{LipSwish}(x)}{\partial x} \\ \frac{\partial\,\mathrm{LipSwish}(-x)}{\partial x} \end{bmatrix}, \tag{95}$$

they further define the normalized activation

$$\mathrm{CLipSwish}(x) \;=\; \frac{\Phi(x)}{1.004}, \tag{96}$$

so that $\sup_x \|J_{\mathrm{CLipSwish}}(x)\|_2 \le 1$. When combined with spectral-normalized layers satisfying $\|W^{(\ell)}\|_2 \le 1$, the whole network is 1-Lipschitz.

4.4 Class-Margin Regularization

Lipschitz-Margin Training (Tsuzuku et al., 2018). The key idea is to make the classifier's decision margin large enough—relative to its Lipschitz constant—to guarantee robustness to ℓ_2 perturbations up to a radius c. Let $F_k(x)$ denote the logit for class k and t_x the true class. If L_F is a valid upper bound on the ℓ_2-Lipschitz constant of the logit map F, then the prediction is certifiably unchanged under any ℓ_2 perturbation $\|\delta\|_2 \ge c$ whenever

$$\min_{i \ne t_x} \left[F_{t_x}(x) - F_i(x)\right] \;\ge\; \sqrt{2c}\, L_F. \tag{97}$$

Since computing L_F exactly is intractable, Tsuzuku et al. (2018) estimate it on-the-fly via the product of per-layer spectral norms, yielding a differentiable approximation $\widehat{L}_F$. During training, they shift all non-true logits upward by $\sqrt{2c}\,\widehat{L}_F$ before applying the usual classification loss. This directly enforces the margin condition in each SGD step, scaling the required margin to the current Lipschitz estimate and producing models with certifiable robustness to radius c perturbations.

4.5 Architectural Regularization

LipsFormer (Qi et al., 2023). The LipsFormer architecture (Qi et al., 2023) modifies several core Transformer components to be Lipschitz-continuous, thereby improving training stability and enabling provable bounds on sensitivity. The main changes are:

1. **CenterNorm instead of LayerNorm.** As background, the standard LayerNorm (Ba et al., 2016; Qi et al., 2023) for input $\mathbf{x} \in \mathbb{R}^D$ is

$$\mathrm{LN}(\mathbf{x}) = \gamma \odot \frac{\mathbf{x} - \mu(\mathbf{x})}{\sigma(\mathbf{x})} + \beta, \tag{98}$$

where $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$ are the mean and standard deviation of $\mathbf{x}$ across its feature dimension, $\odot$ denotes element-wise multiplication, and γ, β are learnable affine parameters. However, LayerNorm is *not* Lipschitz-continuous: for zero variance inputs, the Jacobian becomes unbounded, which can destabilize training without learning-rate warmup.

To address this, LipsFormer introduces *CenterNorm*:

$$\mathrm{CN}(\mathbf{x}) = \gamma \odot \frac{D}{D-1}\left(I - \frac{1}{D}\mathbf{1}\mathbf{1}^\top\right)\mathbf{x} + \beta, \tag{99}$$

where D is the feature dimension. The factor $\frac{D}{D-1}$ heuristically avoids eigenvalue contraction from the centering matrix $I - \frac{1}{D}\mathbf{1}\mathbf{1}^\top$. The Lipschitz constant satisfies

$$\|\mathrm{CN}(\mathbf{x}_1) - \mathrm{CN}(\mathbf{x}_2)\| \leq \mathrm{Lip}(\mathrm{CN}_\mathbf{x})\,\|\mathbf{x}_1 - \mathbf{x}_2\|, \quad \mathrm{Lip}(\mathrm{CN}_\mathbf{x}) = \frac{D}{D-1} \tag{100}$$

for $\gamma = 1$ and $\beta = 0$, which approaches 1 as D increases. By constraining γ and β to a pre-defined range, Lipschitz continuity is preserved during training.

2. **Scaled cosine similarity attention (SCSA).** To replace the standard dot-product attention, which is not Lipschitz-continuous (Kim et al., 2021), LipsFormer adopts a scaled cosine similarity attention that is Lipschitz-continuous by construction. It is defined as

$$\mathrm{SCSA}(X, W^Q, W^K, W^V, \nu, \tau) = \nu P V, \qquad P = \mathrm{softmax}(\tau Q K^\top), \tag{101}$$

where ν and τ are predefined or learnable scalars, and

$$Q = \begin{bmatrix} - & \mathbf{q}_1^\top & - \\ \vdots & \vdots & \vdots \\ - & \mathbf{q}_N^\top & - \end{bmatrix}, \quad K = \begin{bmatrix} - & \mathbf{k}_1^\top & - \\ \vdots & \vdots & \vdots \\ - & \mathbf{k}_N^\top & - \end{bmatrix}, \quad V = \begin{bmatrix} - & \mathbf{v}_1^\top & - \\ \vdots & \vdots & \vdots \\ - & \mathbf{v}_N^\top & - \end{bmatrix}. \tag{102}$$

Here Q, K, V are obtained by ℓ_2 row-normalizing the projections of X:

$$\mathbf{q}_i = \frac{(x_i^\top W^Q)^\top}{\|x_i^\top W^Q\|_2 + \epsilon_c}, \qquad \mathbf{k}_i = \frac{(x_i^\top W^K)^\top}{\|x_i^\top W^K\|_2 + \epsilon_c}, \qquad \mathbf{v}_i = \frac{(x_i^\top W^V)^\top}{\|x_i^\top W^V\|_2 + \epsilon_c}, \tag{103}$$

where $\epsilon_c > 0$ is a smoothing factor to guarantee numerical stability.

For any pair of rows $\mathbf{q}_i$ and $\mathbf{k}_j$, their cosine similarity is proportional to their ℓ_2 dot product, ensuring the attention scores remain bounded. This design yields an explicit upper bound on the Lipschitz constant of SCSA with respect to both $\|\cdot\|_2$ and $\|\cdot\|_\infty$, as proven in Theorem 1 of (Qi et al., 2023).

3. **Spectral-based weight initialization.** All projection matrices are initialized with spectral norm 1, ensuring they are 1-Lipschitz from initialization.

4. **Weighted residual shortcut.** To prevent exponential growth of the Lipschitz constant when stacking residual blocks, the standard

$$\mathrm{RS}(\mathbf{x}, W) = \mathbf{x} + f(\mathbf{x}, W) \tag{104}$$

is replaced by

$$\mathrm{WRS}(\mathbf{x}, W) = \mathbf{x} + \alpha \odot f(\mathbf{x}, W), \tag{105}$$

where α is a learnable vector that limits the residual branch's contribution to the overall Lipschitz bound.

JaSMin – Jacobian Softmax norm Minimization (Yudin et al., 2025). Recent theoretical work provides a refined *local* Lipschitz bound for self-attention blocks. The analysis yields a closed-form expression for the spectral norm of the softmax Jacobian, revealing how attention-score distributions impact the local Lipschitz constant. Based on this, the authors introduce the JaSMin (Jacobian Softmax norm Minimization) regularizer, which empirically reduces local Lipschitz values and enhances robustness Yudin et al. (2025).

For the self-attention mechanism, Yudin et al. (2025) establish a local bound for the Jacobian of self-attention operation:

$$\|\mathcal{J}_{\mathrm{Attn}}(X)\|_2 \leq \|W_h^V\|_2 \left(\|P^h\|_2 + 2\|X\|_2^2 \|A_h\|_2 \max_i \|\mathcal{M}(P_{i,:}^h)\|_2 \right), \tag{106}$$

where A_h and P^h are attention matrices, and

$$\mathcal{M}(P^h_{i,:}) := \mathrm{diag}(P^h_{i,:}) - P^{h\top}_{i,:} P^h_{i,:} \tag{107}$$

is the Jacobian of the softmax function for the i-th row of P^h. The inequality decomposes the local Lipschitz constant into: (i) *input-only* terms $\|X\|_2^2$, (ii) *weight-dependent* terms $\|W_h^V\|_2$ and $\|A_h\|_2$, and (iii) *mixed* terms involving the attention score distribution $\mathcal{M}(P^h_{i,:})$. JaSMin directly targets the third term by penalizing the norm of $\mathcal{M}(P^h_{i,:})$, thereby reducing the sensitivity of the attention mechanism to small perturbations in the input distribution. The regularized loss is given by

$$\mathcal{L} = \mathcal{L}_{\text{model}} + \lambda \mathcal{L}_{\text{JaSMin}_k}, \tag{108}$$

where $\mathcal{L}_{\text{model}}$ is the unregularized model loss, λ is a hyperparameter, and $\mathcal{L}_{\text{JaSMin}_k}$ penalizes the logarithm of the upper bound on the local Lipschitz constant for the k-th attention head.

5 Certifiable Robustness

Certifiable robustness, a cornerstone of trustworthy deep learning, guarantees that a neural network's predictions remain consistent under adversarial perturbations within a specified norm ball. Unlike empirical defenses like adversarial training, which lack formal guarantees, Lipschitz-based certification methods leverage the Lipschitz constant to bound sensitivity to input changes, ensuring provable robustness. While some literature, such as (Clevert et al., 2016), have been introduced previously from the perspectives — such as **theoretical foundations**, **estimation methods** and **regularization approaches**, we may revisit some of them in this section from the perspective of **certifiable robustness**.

5.1 Formal Guarantees

Theorem 5.1 (Lipschitz Margin Robustness Radius in ℓ_2-Normed Spaces)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}^n$ be a classifier and $c = \arg\max_i f_i(x)$ its predicted class at x. Define the (one-versus-rest) margin*

$$m(x) \;:=\; f_c(x) - \max_{j \neq c} f_j(x). \tag{109}$$

Assume f is K-Lipschitz from the input ℓ_2 norm to the output ℓ_2 norm, i.e.*,*

$$\|f(x) - f(y)\|_2 \;\leq\; K\,\|x - y\|_2 \quad \textit{for all } x, y. \tag{110}$$

Then for any perturbation δ with

$$\|\delta\|_2 \;<\; \frac{m(x)}{\sqrt{2}\,K}, \tag{111}$$

the prediction is unchanged: $\arg\max_i f_i(x + \delta) = c$. *Similar results have been discussed in the literature (Szegedy et al., 2014; Hein & Andriushchenko, 2017; Cisse et al., 2017; Tsuzuku et al., 2018).*

Proof. Let $e_i \in \mathbb{R}^n$ denote the i-th standard basis vector, *i.e.*, e_i has a 1 in the i-th position and 0 elsewhere. Then

$$f_i(x) = e_i^\top f(x). \tag{112}$$

For each $j \neq c$, define the pairwise margin function:

$$g_j(x) := f_c(x) - f_j(x) = (e_c - e_j)^\top f(x). \tag{113}$$

By Hölder's inequality,

$$|g_j(x) - g_j(y)| = |(e_c - e_j)^\top (f(x) - f(y))| \tag{114}$$
$$\leq \|e_c - e_j\|_2 \cdot \|f(x) - f(y)\|_2 \tag{115}$$
$$\leq \sqrt{2} \cdot K \cdot \|x - y\|_2, \tag{116}$$

so each g_j is $\sqrt{2}K$-Lipschitz:

$$\mathrm{Lip}[g_j] = \sqrt{2}K. \tag{117}$$

Note that the margin $m(x)$ can be expressed as

$$m(x) = f_c(x) - \max_{j \neq c} f_j(x) = \min_{j \neq c} g_j(x). \tag{118}$$

Now let $\delta \in \mathbb{R}^d$ be a perturbation satisfying

$$\|\delta\|_2 < \frac{m(x)}{\sqrt{2}K}. \tag{119}$$

Then for all $j \neq c$,

$$g_j(x + \delta) \geq g_j(x) - \mathrm{Lip}[g_j] \cdot \|\delta\|_2 \tag{120}$$
$$> m(x) - \sqrt{2}K \cdot \frac{m(x)}{\sqrt{2}K} = 0. \tag{121}$$

Hence, $f_c(x + \delta) > f_j(x + \delta)$ for all $j \neq c$, which implies

$$\arg\max_i f_i(x + \delta) = c. \tag{122}$$

□

Theorem 5.2 (Lipschitz Margin Robustness Radius in Conjugate-Normed Spaces)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}^n$ be a classifier with predicted class $c = \arg\max_i f_i(x)$ and one-versus-rest margin*

$$m(x) := f_c(x) - \max_{j \neq c} f_j(x). \tag{123}$$

Assume f is K-Lipschitz from the input ℓ_p norm to the output ℓ_q norm:

$$\|f(x) - f(y)\|_q \;\leq\; K\,\|x - y\|_p. \tag{124}$$

Let p be the dual exponent of q, i.e., $\frac{1}{q} + \frac{1}{p} = 1$. *Then the classification of x is certifiably robust to any perturbation δ with*

$$\|\delta\|_p \;<\; r_{\mathrm{cert}}(x) := \frac{m(x)}{K\,\alpha}, \qquad \alpha := \max_{j \neq c} \|e_c - e_j\|_p, \tag{125}$$

where $e_i \in \mathbb{R}^n$ denotes the i-th standard basis vector — e.g., *one-hot vector. Similar results have been discussed in the literature (Hein & Andriushchenko, 2017; Weng et al., 2018b).*

Proof. Let $e_i \in \mathbb{R}^n$ denote the i-th standard basis vector, *i.e.*, e_i has a 1 in the i-th position and 0 elsewhere. Then

$$f_i(x) = e_i^\top f(x). \tag{126}$$

For each $j \neq c$, define the pairwise margin function:

$$g_j(x) := f_c(x) - f_j(x) = (e_c - e_j)^\top f(x). \tag{127}$$

By Hölder's inequality and the Lipschitz property of f,

$$|g_j(x) - g_j(y)| \leq \|e_c - e_j\|_p \, \|f(x) - f(y)\|_q \tag{128}$$
$$\leq K \, \|e_c - e_j\|_p \, \|x - y\|_p. \tag{129}$$

Thus the Lipschitz constant of g_j is:

$$\mathrm{Lip}[g_j] \leq K \, \|e_c - e_j\|_p. \tag{130}$$

If:

$$\|x' - x\|_p < \frac{g_j(x)}{\mathrm{Lip}[g_j]}, \tag{131}$$

for all $j \neq c$, then $g_j(x') > 0$ for all $j \neq c$, preserving the predicted class. Taking the minimum over j yields

$$r_{\mathrm{cert}}(x) = \min_{j \neq c} \frac{g_j(x)}{\mathrm{Lip}[g_j]} \geq \frac{m(x)}{K \, \max_{j \neq c} \|e_c - e_j\|_p}. \tag{132}$$

□

Remark 5.3. Common output norm choices yield:

$$\ell_2 \text{ output norm:} \quad \alpha = \sqrt{2} \quad \Rightarrow \quad r_{\mathrm{cert}} \geq \frac{m(x)}{\sqrt{2}\, K}, \tag{133}$$

$$\ell_\infty \text{ output norm:} \quad \alpha = 2 \quad \Rightarrow \quad r_{\mathrm{cert}} \geq \frac{m(x)}{2K}, \tag{134}$$

$$\ell_1 \text{ output norm:} \quad \alpha = 1 \quad \Rightarrow \quad r_{\mathrm{cert}} \geq \frac{m(x)}{K}. \tag{135}$$

Note that the number of classes n affects the certified radius only through the margin $m(x)$; it does not appear explicitly in the bound.

5.2 Certifiable Robustness via Global Lipschitz Bound

For networks with 1-Lipschitz activations and linear layers $W^{(\ell)}$, submultiplicativity yields

$$\mathrm{Lip}[f] \leq \prod_{\ell=1}^{L} \|W^{(\ell)}\|_2, \tag{136}$$

providing a *certified global bound*. This is computationally cheap but typically loose. LipSDP (Fazlyab et al., 2019) tightens global bounds via a semidefinite relaxation using *incremental quadratic constraints* (IQCs) for slope-bounded nonlinearities. Solving

$$\min_{\rho,\, T} \ \rho \quad \text{s.t.} \quad M(\rho, T) \succeq 0, \quad T \in \mathcal{T}_n \tag{137}$$

yields $\mathrm{Lip}[f] \leq \sqrt{\rho}$, a certified global bound often much tighter than spectral norm products. SeqLip (Virmaux & Scaman, 2018) refines the spectral product by incorporating singular-vector alignment between layers, also producing certified global bounds.

5.3 Certifiable Robustness via Local Lipschitz Bound

Local Lipschitz bounds are computed over a neighborhood $B_p(x, R)$ and substituted for L_j in the margin condition. FastLip (Weng et al., 2018a) propagates interval bounds on partial derivatives layer-by-layer, classifying ReLUs as active, inactive, or uncertain. For each input coordinate, the worst-case derivative magnitude is computed over the ball, then aggregated in the dual norm to yield a *certified local* Lipschitz bound. MILP-based methods (Jordan & Dimakis, 2020) encode the exact piecewise-linear constraints of ReLU networks within the ball, yielding exact local Lipschitz constants (and thus exact certified radii), but are limited to small networks. CLEVER (Weng et al., 2018b) estimates local constants via extreme value theory on sampled gradient norms; this improves tightness but does not yield a formal certificate.

5.4 Certifiable Robustness in LLMs

Bounded Levenshtein Distance (Rocamora et al., 2025). The *Levenshtein distance* $d_{\mathrm{Lev}}(x, x')$ between two token sequences x and x' is defined as the minimum number of insertions, deletions, and substitutions required to transform x into x'. In the context of robustness certification, a model f is said to be *L_f-Lipschitz with respect to d_{Lev}* if

$$\|f(x) - f(x')\| \leq L_f \, d_{\mathrm{Lev}}(x, x'), \tag{138}$$

for all x, x'. Enforcing $L_f \leq 1$ ensures that the model output changes by at most the number of edits applied to the input.

When inputs or intermediate representations are sequences of real-valued vectors (*e.g.*, word embeddings), d_{Lev} cannot be directly applied. In this case, the *edit distance with real penalty* (ERP), $d^p_{\mathrm{ERP}}(A, B)$, which extends d_{Lev} to $\mathbb{R}^d$-valued sequences and recovers d_{Lev} when A and B are one-hot sequences and $p = \infty$.

Rocamora et al. (2025) introduce **LipsLev**, a method for training convolutional text classifiers that are *1-Lipschitz* with respect to d_{Lev}. They derive a global Lipschitz constant bound for the classification margin $g_{y,\hat{y}}$ between classes y and $\hat{y}$:

$$|g_{y,\hat{y}}(P) - g_{y,\hat{y}}(Q)| \leq \|w_y - w_{\hat{y}}\|_r \left(\prod_{i=1}^{l} M(\mathcal{K}^{(i)}) \right) \cdot M(E) \cdot d_{\mathrm{Lev}}(P, Q), \tag{139}$$

where P, Q are sequences of one-hot vectors, $M(\mathcal{K}^{(i)})$ is the Lipschitz constant of the i-th convolutional block, and $M(E)$ depends on the embedding layer geometry.

To enforce $L_f \leq 1$, LipsLev divides the weights of each layer by its Lipschitz constant during training, ensuring the whole network is 1-Lipschitz with respect to d_{Lev}. They further provide a *local* Lipschitz bound adapted to a specific input sequence P:

$$|\hat{g}_{y,\hat{y}}(P) - \hat{g}_{y,\hat{y}}(Q)| \leq \frac{\|w_y - w_{\hat{y}}\|_r}{M(W)} \cdot \frac{M(E, P)}{M(E)} \cdot d_{\mathrm{Lev}}(P, Q), \tag{140}$$

where $M(W) = \max_{y,\hat{y}} \|w_y - w_{\hat{y}}\|_r$. This local bound can yield tighter certified radii and demonstrates that their method bridges continuous Lipschitz verification with discrete edit-distance robustness.

6 Conclusions

Lipschitz continuity serves as a cornerstone for enhancing the robustness, generalization, and optimization stability of deep neural networks, providing a rigorous mathematical framework to quantify and control sensitivity to input perturbations. This comprehensive survey synthesizes key insights across theoretical foundations, estimation techniques, regularization approaches, and certifiable robustness, highlighting their interconnections and practical implications. By unifying disparate research threads, we address a critical gap in the literature, offering a cohesive resource for researchers and practitioners. Challenges remain in balancing bound tightness with computational scalability, preserving model expressivity under constraints, and extending certifications to diverse norms and large-scale architectures like transformers, paving the way for future advancements in trustworthy deep learning systems.

References

Stephanie Abrecht, Alexander Hirsch, Shervin Raafatnia, and Matthias Woehrle. Deep learning safety concerns in automated driving perception. *arXiv preprint arXiv:2309.03774*, 2023.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

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

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 214–223. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/arjovsky17a.html`.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*, pp. 39–57. IEEE, 2017.

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

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning (ICML)*, pp. 854–863. PMLR, 2017. URL `https://arxiv.org/abs/1704.08847`.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *ICLR*, 2016.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. *NIPS*, 2001.

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.

Herbert Federer. *Geometric measure theory*. Springer, 2014.

Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

Antoine Gonon. A rescaling-invariant lipschitz bound based on path-metrics. *arXiv preprint arXiv:2405.15006*, 2024.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

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.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. *Advances in neural information processing systems*, 30, 2017.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8340–8349, 2021.

Catherine F. Higham and Desmond J. Higham. Deep learning: An introduction for applied mathematicians. *SIAM Journal on Mathematics of Data Science*, 1(1):1–24, 2019. doi: 10.1137/18M1165748.

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*, pp. 345–362. Springer, 2024.

Matt Jordan and Alexandros G. Dimakis. Exactly computing the local lipschitz constant of relu networks. In *International Conference on Machine Learning (ICML)*, pp. 4985–4994. PMLR, 2020. URL `https://arxiv.org/abs/2002.11572`.

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pp. 9–50. Springer, 2002.

Mario Lezcano-Casado and David Martınez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *International Conference on Machine Learning*, pp. 3794–3803. PMLR, 2019.

Róisín Luo, James McDermott, and Colm O'Riordan. Interpreting global perturbation robustness of image models using axiomatic spectral importance decomposition. *Transactions on Machine Learning Research (TMLR)*, 2024.

Róisín Luo, James McDermott, Christian Gagné, Qiang Sun, and Colm O'Riordan. Optimization-induced dynamics of Lipschitz continuity in neural networks, 2025a. URL `https://arxiv.org/abs/2506.18588`.

Róisín Luo, James McDermott, and Colm O'Riordan. Higher-order singular-value derivatives of rectangular real matrices, 2025b. URL https://arxiv.org/abs/2506.03764.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

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

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

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. *ICML*, 2010.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on learning theory*, pp. 1376–1401. PMLR, 2015a.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on learning theory*, pp. 1376–1401. PMLR, 2015b.

Yura Perugachi-Diaz, Jakub Tomczak, and Sandjai Bhulai. Invertible densenets with concatenated lipswish. *Advances in Neural Information Processing Systems*, 34:17246–17257, 2021.

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

Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

Elias Abad Rocamora, Grigorios G Chrysos, and Volkan Cevher. Certified robustness under bounded levenshtein distance. *arXiv preprint arXiv:2501.13676*, 2025.

Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 3 edition, 1976.

Hanie Sedghi, Vineet Gupta, and Philip M Long. The singular values of convolutional layers. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=Hygvb2C5FX.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, Cambridge, UK, 2014. ISBN 9781107057135. URL https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/copy.html.

Gilbert Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 4th edition, 2009.

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

Terence Tao. *Analysis*, volume 1. Springer, 2006.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Shankar Batra, Pallavi Bhargava, Shruti Bhosale, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1805.12152.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

James Vuckovic, Aristide Baratin, and Remi Tachet des Combes. A mathematical theory of attention. *arXiv preprint arXiv:2007.02876*, 2020.

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

Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv preprint arXiv:1801.10578*, 2018b.

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

Kôsaku Yosida. *Functional analysis*, volume 123. Springer Science & Business Media, 2012.

Nikolay Yudin, Alexander Gaponov, Sergei Kudriashov, and Maxim Rakhuba. Pay attention to attention distribution: A new local lipschitz bound for transformers. *arXiv preprint arXiv:2507.07814*, 2025.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning still requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

A Proofs: Lipschitz Constants of Activation Functions

A.1 Proof: Lipschitz Constant of Sigmoid

Proof. Consider the sigmoid function:

$$\text{Sigmoid}(x) = f(x) = (1 + e^{-x})^{-1}. \tag{141}$$

Let $u = 1 + e^{-x}$, so $f(x) = u^{-1}$. The derivative is:

$$f'(x) = -\frac{1}{u^2} \cdot \frac{du}{dx}. \tag{142}$$

Compute $\frac{du}{dx}$:

$$u = 1 + e^{-x}, \quad \frac{du}{dx} = -e^{-x}. \tag{143}$$

Thus:

$$f'(x) = -\frac{1}{(1 + e^{-x})^2} \cdot (-e^{-x}) = \frac{e^{-x}}{(1 + e^{-x})^2}. \tag{144}$$

Express the derivative in terms of $f(x)$:

$$1 - f(x) = \frac{e^{-x}}{1 + e^{-x}}, \tag{145}$$

and:

$$f'(x) = \frac{1}{1 + e^{-x}} \cdot \frac{e^{-x}}{1 + e^{-x}} = f(x)(1 - f(x)). \tag{146}$$

Thus:

$$|f'(x)| = f(x)(1 - f(x)). \tag{147}$$

Maximize the Derivative. Since $0 < f(x) < 1$, we maximize $g(z) = z(1 - z)$ for $z = f(x) \in (0, 1)$:

$$g'(z) = 1 - 2z = 0 \implies z = \frac{1}{2}, \tag{148}$$

thus:

$$g\left(\frac{1}{2}\right) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}. \tag{149}$$

Find when $f(x) = \frac{1}{2}$:

$$\frac{1}{1 + e^{-x}} = \frac{1}{2} \implies e^{-x} = 1 \implies x = 0, \tag{150}$$

at $x = 0$:

$$f'(0) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}. \tag{151}$$

As $x \to \infty$, $e^{-x} \to 0$, so $f'(x) \to 0$. As $x \to -\infty$, $e^{-x} \to \infty$, so:

$$f'(x) \approx \frac{e^{-x}}{e^{-2x}} = e^{x} \to 0. \tag{152}$$

Hence:

$$\text{Lip}\left[\text{Sigmoid}(x)\right] = \sup_{x} |f'(x)| = \frac{1}{4}. \tag{153}$$

□

A.2 Proof: Lipschitz Constant of Tanh

Proof. The tanh is defined as:

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}, \tag{154}$$

for all $x, y \in \mathbb{R}$.

Let $g(x) = \sinh(x)$, $h(x) = \cosh(x)$, so $f(x) = \frac{g(x)}{h(x)}$. The derivative is:

$$f'(x) = \frac{g'(x)h(x) - g(x)h'(x)}{h(x)^2}. \tag{155}$$

Since $g'(x) = \cosh(x)$, $h'(x) = \sinh(x)$, and $h(x)^2 = \cosh^2(x)$:

$$f'(x) = \frac{\cosh(x) \cdot \cosh(x) - \sinh(x) \cdot \sinh(x)}{\cosh^2(x)} = \frac{\cosh^2(x) - \sinh^2(x)}{\cosh^2(x)}. \tag{156}$$

Using the identity $\cosh^2(x) - \sinh^2(x) = 1$:

$$f'(x) = \frac{1}{\cosh^2(x)} = \operatorname{sech}^2(x). \tag{157}$$

Since $\cosh(x) \geq 1$, we have:

$$|f'(x)| = \operatorname{sech}^2(x). \tag{158}$$

Express the derivative in terms of $f(x)$:

$$f'(x) = \operatorname{sech}^2(x) = 1 - \tanh^2(x) = 1 - f(x)^2. \tag{159}$$

Since $-1 \leq f(x) \leq 1$, we have $|f'(x)| = 1 - f(x)^2 \leq 1$.

Maximize the Derivative. Maximize $|f'(x)| = \operatorname{sech}^2(x)$:

$$\cosh(x) = \frac{e^x + e^{-x}}{2}. \tag{160}$$

At $x = 0$:

$$\cosh(0) = \frac{e^0 + e^0}{2} = 1, \quad \operatorname{sech}^2(0) = \frac{1}{\cosh^2(0)} = 1. \tag{161}$$

As $|x| \to \infty$, $\cosh(x) \approx \frac{e^{|x|}}{2}$, so:

$$\operatorname{sech}^2(x) \approx \frac{4}{e^{2|x|}} \to 0. \tag{162}$$

The supremum of $|f'(x)|$ is 1 at $x = 0$. Alternatively, since $f(x)^2 \leq 1$, the supremum of $1 - f(x)^2$ occurs when $f(x) = 0$:

$$\tanh(0) = 0, \quad f'(0) = 1 - 0^2 = 1. \tag{163}$$

Hence:

$$\operatorname{Lip}[\tanh(x)] = \sup_x |f'(x)| = 1. \tag{164}$$

□

A.3 Proof: Lipschitz Constant of Softplus

Proof. Consider the softplus function:

$$f(x) = \ln(1 + e^x). \tag{165}$$

The derivative is:

$$f'(x) = \frac{d}{dx} \ln(1 + e^x) = \frac{e^x}{1 + e^x}. \tag{166}$$

Since $e^x > 0$ and $1 + e^x > 1$, we have $f'(x) > 0$, so:

$$|f'(x)| = \frac{e^x}{1 + e^x}. \tag{167}$$

Maximize the Derivative. To find the Lipschitz constant, we need to compute:

$$K = \sup_{x \in \mathbb{R}} \frac{e^x}{1 + e^x}. \tag{168}$$

Notice that $\frac{e^x}{1+e^x}$ is the sigmoid function, which ranges between 0 and 1. Analyze its behavior:

1. As $x \to \infty$, e^x grows large, so:

$$\frac{e^x}{1 + e^x} \approx \frac{e^x}{e^x} = 1. \tag{169}$$

2. As $x \to -\infty$, $e^x \to 0$, so:

$$\frac{e^x}{1 + e^x} \to \frac{0}{1} = 0. \tag{170}$$

To confirm the supremum, consider the function $g(x) = \frac{e^x}{1+e^x}$. Its derivative is:

$$g'(x) = \frac{e^x(1 + e^x) - e^x \cdot e^x}{(1 + e^x)^2} = \frac{e^x}{(1 + e^x)^2}. \tag{171}$$

Since $g'(x) > 0$ for all x, $g(x)$ is strictly increasing, approaching 0 as $x \to -\infty$ and 1 as $x \to \infty$. Thus:

$$\sup_{x \in \mathbb{R}} \frac{e^x}{1 + e^x} = 1 \tag{172}$$

Hence:

$$\text{Lip}\,[\text{Softplus}(x)] = \sup_x |f'(x)| = 1. \tag{173}$$

□

A.4 Proof: Lipschitz Constant of Swish

Proof. Let

$$f(x) = x\,\sigma(x), \quad \sigma(x) = \frac{1}{1 + e^{-x}}. \tag{174}$$

The derivative is

$$g(x) := f'(x) = \sigma(x) + x\,\sigma(x)\big(1 - \sigma(x)\big). \tag{175}$$

Since $\sigma(-x) = 1 - \sigma(x)$, we have

$$g(-x) = 1 - g(x). \tag{176}$$

Using $\sigma(x) = \frac{1}{2}\big(1 + \tanh\big(\frac{x}{2}\big)\big)$ and $\sigma(x)(1 - \sigma(x)) = \frac{1}{4}\,\mathrm{sech}^2(\frac{x}{2})$,

$$g(x) = \frac{1}{2}\Big(1 + \tanh\frac{x}{2}\Big) + \frac{x}{4}\,\mathrm{sech}^2\frac{x}{2}. \tag{177}$$

Differentiating,

$$g'(x) = \frac{1}{4}\,\mathrm{sech}^2\frac{x}{2}\,\Big(2 - x\,\tanh\frac{x}{2}\Big), \tag{178}$$

so the maximizer $x^\star > 0$ satisfies

$$x^\star\,\tanh\frac{x^\star}{2} = 2. \tag{179}$$

Using $\tanh\Big(\frac{x^\star}{2}\Big) = \frac{2}{x^\star}$ and

$$\mathrm{sech}^2\frac{x^\star}{2} = \frac{(x^\star)^2 - 4}{(x^\star)^2}. \tag{180}$$

Thus

$$K = g(x^\star) = \frac{1}{2} + \frac{x^\star}{4}. \tag{181}$$

Since $g(-x) = 1 - g(x)$, $x^\star$ gives the global maximum of $|g|$, solving $x^\star\,\tanh\frac{x^\star}{2} = 2$ gives:

$$x^\star \approx 2.3993572805\cdots. \tag{182}$$

Hence:

$$\mathrm{Lip}\,[\mathrm{Siwsh}(x)] = \sup_x |f'(x)| \approx 1.09983932\cdots \tag{183}$$

□

A.5 Proof: Lipschitz constant of GELU

Proof. Let:

$$f(x) = x\,\Phi(x), \tag{184}$$

where:

$$\Phi(x) = \tfrac{1}{2}\big(1 + \mathrm{erf}(x/\sqrt{2})\big) \tag{185}$$

is the standard normal CDF and:

$$\phi(x) = \tfrac{1}{\sqrt{2\pi}} e^{-x^2/2} \tag{186}$$

is its PDF.

Differentiate:

$$g(x) := f'(x) = \Phi(x) + x\,\phi(x). \tag{187}$$

Note the symmetry $\Phi(-x) = 1 - \Phi(x)$ and $\phi(-x) = \phi(x)$, hence

$$g(-x) = 1 - g(x). \tag{188}$$

Compute the critical points:

$$g'(x) = \phi(x) + \phi(x) + x\,\phi'(x) = 2\phi(x) - x^2\phi(x) = \phi(x)\,(2 - x^2). \tag{189}$$

Since $\phi(x) > 0$, we have $g'(x) > 0$ for $|x| < \sqrt{2}$ and $g'(x) < 0$ for $|x| > \sqrt{2}$. Hence g attains its unique global maximum at $x^\star = \sqrt{2}$ and minimum at $-\sqrt{2}$.

Since $g(-x) = 1 - g(x)$, the minimum equals $1 - g(\sqrt{2})$, so indeed $\sup_x |g(x)| = g(\sqrt{2})$.

Evaluate at:

$$x = \sqrt{2} \approx 1.414224\cdots, \tag{190}$$

hence:

$$\begin{aligned}
\mathrm{Lip}\,[\mathrm{GELU}(x)] &= \sup_x |f'(x)| && (191)\\
&= g(\sqrt{2}) && (192)\\
&= \Phi(\sqrt{2}) + \sqrt{2}\,\phi(\sqrt{2}) && (193)\\
&= \tfrac{1}{2}\big(1 + \mathrm{erf}(1)\big) + \frac{e^{-1}}{\sqrt{\pi}} && (194)\\
&\approx 1.128904145. && (195)
\end{aligned}$$

□

A.6 Proof: Lipschitz constant of Softmax

Proof. Let

$$\mathrm{softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}}, \tag{196}$$

and:

$$p := \mathrm{softmax}(z) \in \Delta^{n-1}, \tag{197}$$

where Δ^{n-1} is the probability simplex in $\mathbb{R}^n$ (*i.e.*, the set of all probability vectors of length n):

$$\Delta^{n-1} \;:=\; \big\{\, p \in \mathbb{R}^n \;\big|\; p_i \geq 0,\; \sum_{i=1}^n p_i = 1 \,\big\}. \tag{198}$$

The Jacobian is

$$J(z) = \nabla\,\mathrm{softmax}(z) = \mathrm{diag}(p) - p\,p^\top, \tag{199}$$

which is symmetric positive semidefinite (PSD) and satisfies:

$$J(z)\,\mathbf{1} = 0. \tag{200}$$

Hence the Lipschitz constant K is:

$$K = \sup_{z\in\mathbb{R}^n} \|J(z)\|_2 = \sup_{p\in\Delta^{n-1}} \lambda_{\max}\big(\mathrm{diag}(p) - p\,p^\top\big). \tag{201}$$

where $\lambda_{\max}$ is the largest singular value of $\mathrm{diag}(p) - p\,p^\top$.

For any unit vector $v \in \mathbb{R}^n$,

$$v^\top J(z)\,v = v^\top \mathrm{diag}(p)\,v - (p^\top v)^2 = \sum_{i=1}^n p_i v_i^2 - \Big(\sum_{i=1}^n p_i v_i\Big)^2 = \mathrm{Var}_{i\sim p}[v_i]. \tag{202}$$

Therefore

$$\|J(z)\|_2 = \sup_{\|v\|_2=1} \mathrm{Var}_{i\sim p}[v_i]. \tag{203}$$

By Popoviciu's inequality on variances, for any random variable X supported in $[a, b]$,

$$\mathrm{Var}[X] \le \frac{(b-a)^2}{4}, \tag{204}$$

with equality attained by a two–point distribution at the endpoints.

Applying this to $X = v_i$ under p, we get

$$\mathrm{Var}_{i\sim p}[v_i] \le \frac{\big(\max_i v_i - \min_i v_i\big)^2}{4}. \tag{205}$$

Maximizing the RHS over $\|v\|_2 = 1$ yields $\max_i v_i - \min_i v_i \le \sqrt{2}$, with equality for

$$v = \frac{1}{\sqrt{2}}(e_a - e_b) \quad \text{for some distinct } a, b, \tag{206}$$

so that

$$\sup_{\|v\|_2=1} \mathrm{Var}_{i\sim p}[v_i] \le \frac{(\sqrt{2})^2}{4} = \frac{1}{2}. \tag{207}$$

This upper bound is (arbitrarily) attainable by choosing p supported on the two indices a, b with $p_a = p_b = \frac{1}{2}$ and $p_i \to 0$ for $i \notin \{a, b\}$ (which is approached by softmax logits $z_a = z_b \gg z_{i\notin\{a,b\}}$). In that case,

$$J = \begin{bmatrix} \frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{1}{4} \end{bmatrix} \oplus 0_{n-2}, \tag{208}$$

whose largest eigenvalue is $\frac{1}{2}$.

Hence:

$$\mathrm{Lip}\,[\mathrm{Softmax}(x)] = \sup_z \|J(z)\|_2 = \frac{1}{2}, \tag{209}$$

independent of $n \ge 2$.

□