The Lipschitz Constant of Self-Attention

Hyunjik Kim [1] **George Papamakarios** [1] **Andriy Mnih** [1]

Abstract

Lipschitz constants of neural networks have been explored in various contexts in deep learning, such as provable adversarial robustness, estimating Wasserstein distance, stabilising training of GANs, and formulating invertible neural networks. Such works have focused on bounding the Lipschitz constant of fully connected or convolutional networks, composed of linear maps and pointwise non-linearities. In this paper, we investigate the Lipschitz constant of self-attention, a non-linear neural network module widely used in sequence modelling. We prove that the standard dot-product self-attention is *not* Lipschitz for unbounded input domain, and propose an alternative L2 self-attention that *is* Lipschitz. We derive an upper bound on the Lipschitz constant of L2 self-attention and provide empirical evidence for its asymptotic tightness. To demonstrate the practical relevance of our theoretical work, we formulate invertible self-attention and use it in a Transformer-based architecture for a character-level language modelling task.

1. Introduction

Lipschitz continuity is a strong form of continuity for functions. Loosely speaking, a function is *Lipschitz continuous* if changing its input by a certain amount cannot change its output by more than K times that amount. The constant K is a hard constraint on how rapidly the function's output can vary, and the smallest such K is known as the function's *Lipschitz constant*. For example, $f_1(x) = \sqrt{|x|}$ and $f_2(x) = \exp(x)$ for $x \in \mathbb{R}$ are not Lipschitz continuous, because their output can change arbitrarily fast as x approaches 0 and $+\infty$ respectively. On the other hand, $g_1(x) = \tanh(x)$ and $g_2(x) = \alpha x$ are Lipschitz continuous, because their rate of change (derivative) is bounded.

In deep learning, we often use Lipschitz continuity as a constraint for neural networks, to control how much a network's output can change relative to its input. Such Lipschitz constraints are useful in several contexts. For example, Lipschitz constraints can endow models with provable robustness against adversarial pertubations (Cisse et al., 2017; Tsuzuku et al., 2018; Anil et al., 2019), and guaranteed generalisation bounds (Sokolić et al., 2017). Moreover, the dual form of the Wasserstein distance is defined as a supremum over Lipschitz functions with a given Lipschitz constant, hence Lipschitz-constrained networks are used for estimating Wasserstein distances (Peyré & Cuturi, 2019). Further, Lipschitz-constrained networks can stabilise training for GANs, an example being spectral normalisation (Miyato et al., 2018). Finally, Lipschitz-constrained networks are also used to construct invertible models and normalising flows. For example, Lipschitz-constrained networks can be used as a building block for invertible residual networks and hence flow-based generative models (Behrmann et al., 2019; Chen et al., 2019). Additionally, Neural ODEs (Chen et al., 2018; Grathwohl et al., 2019) are typically defined using vector fields parameterized via Lipschitz networks, so that the flow generated by the vector field is guaranteed to exist for all times.

Nonetheless, designing Lipschitz-continuous neural networks and computing (or even upper-bounding) their Lipschitz constant is a hard problem. Previous work mostly focused on fully-connected and convolutional networks, not only because they are common in deep learning, but also because they are relatively simple to analyze, as compositions of linear maps and pointwise non-linearities. Even in this case however, exact evaluation of the Lipschitz constant of fully-connected and convolutional networks is NP-hard (Virmaux & Scaman, 2018) and obtaining a tight upper bound remains a challenging task (Virmaux & Scaman, 2018; Fazlyab et al., 2019; Latorre et al., 2020).

Fully-connected and convolutional networks are not the only neural networks worthy of interest. Recently, *self-attention* (Vaswani et al., 2017) has become a popular alternative to recurrent neural networks. Self-attention is a key component of the Transformer (Vaswani et al., 2017), that has found success as a building block in models of various data modalities, starting with natural-language processing (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020) and extending to computer vision

[1]DeepMind, UK. Correspondence to: Hyunjik Kim <hyunjikk@google.com>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

(Zhang et al., 2019; Parmar et al., 2019), audio generation (Huang et al., 2019), and reinforcement learning (Parisotto et al., 2020). However, so far no previous work has analysed the Lipschitz properties of self-attention, and thus it has been unclear whether self-attention is a viable option in applications that require Lipschitz constraints. In this work, we address this gap in the theory of self-attention by providing a thorough analysis of its Lipschitz properties. In particular, we make the following contributions:

- We prove that the widely used *dot-product self-attention* is *not* Lipschitz, and therefore not suitable to use in applications requiring Lipschitz constraints.
- We formulate *L2 self-attention* as an alternative, and show that it *is* Lipschitz.
- We derive a theoretical upper bound on the Lipschitz constant of L2 self-attention, and provide empirical evidence of the asymptotic tightness of the bound.
- As a practical demonstration of the theory, we use this bound to formulate invertible self-attention, and explore its use in a Transformer architecture for character-level language modelling. We compare its test log-likelihood and stability to dot-product self-attention.

2. Lipschitz Constant of Fully-Connected/Convolutional Layers

We first define the notion of Lipschitz continuity, and proceed to define the Lipschitz constant.

Definition 2.1. Given two metric spaces $(\mathcal{X}, d_{\mathcal{X}})$ and $(\mathcal{Y}, d_{\mathcal{Y}})$, a function $f : \mathcal{X} \to \mathcal{Y}$ is called *Lipschitz continuous* (or K-*Lipschitz*) if there exists a constant $K \geq 0$ such that

$$d_{\mathcal{Y}}(f(\mathbf{x}), f(\mathbf{x}')) \leq K d_{\mathcal{X}}(\mathbf{x}, \mathbf{x}') \quad \text{for all } \mathbf{x}, \mathbf{x}' \in \mathcal{X}. \quad (1)$$

The smallest such K is the *Lipschitz constant* of f, denoted $\mathrm{Lip}(f)$.

In this paper, we focus on the common case where $\mathcal{X} = \mathbb{R}^n$, $\mathcal{Y} = \mathbb{R}^m$, and $d_{\mathcal{X}}, d_{\mathcal{Y}}$ are induced by a p-norm $\|\mathbf{x}\|_p := (\sum_i |x_i|^p)^{1/p}$. We will primarily consider the cases $p = 2$ and $p = \infty$, where $\|\mathbf{x}\|_\infty := \max_i |x_i|$. To emphasise the dependence of the Lipschitz constant on the choice of p-norm, we will often denote it by $\mathrm{Lip}_p(f)$. In this case, it follows directly from Definition 2.1 that the Lipschitz constant is given by

$$\mathrm{Lip}_p(f) = \sup_{\mathbf{x} \neq \mathbf{x}' \in \mathbb{R}^n} \frac{\|f(\mathbf{x}) - f(\mathbf{x}')\|_p}{\|\mathbf{x} - \mathbf{x}'\|_p}. \quad (2)$$

Next, we outline some basic results that are useful for estimating Lipschitz constants, also covered in related works (Virmaux & Scaman, 2018; Behrmann et al., 2019). We describe how these results are used to provide bounds on the Lipschitz constant of fully-connected networks (`FCN`) and convolutional neural networks (`CNN`), using the fact that both are compositions of linear maps and pointwise non-linearities. To begin with, the following theorem suggests a way to bound $\mathrm{Lip}_p(f)$ for a differentiable Lipschitz function f:

Theorem 2.1 (Federer, 1969)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be differentiable and Lipschitz continuous under a choice of p-norm $\|\cdot\|_p$. Let $J_f(x)$ denote its total derivative (Jacobian) at x. Then $\mathrm{Lip}_p(f) = \sup_{\mathbf{x} \in \mathbb{R}^n} \|J_f(\mathbf{x})\|_p$ where $\|J_f(\mathbf{x})\|_p$ is the induced operator norm on $J_f(\mathbf{x})$.*

Hence if f is a linear map represented by a matrix W then

$$\begin{aligned}\mathrm{Lip}_p(f) = \|W\|_p &:= \sup_{\|\mathbf{x}\|_p = 1} \|W\mathbf{x}\|_p \\ &= \begin{cases} \sigma_{\max}(W), & \text{if } p = 2 \\ \max_i \sum_j |W_{ij}| & \text{if } p = \infty \end{cases}\end{aligned}$$

where $\|W\|_p$ is the operator norm on matrices induced by the vector p-norm, and $\sigma_{\max}(W)$ is the largest singular value of W. Under this choice of norm, many common non-linearities (including `relu`, `sigmoid`, `tanh`, `elu`) are 1-Lipschitz. $\|W\|_2 = \sigma_{\max}(W)$ is usually estimated via *power iteration*; we provide details on how this is done in Appendix B.

Since we now know the Lipschitz constants of the components of both `FCN` and `CNN`, we can bound their Lipschitz constants by applying the following lemma:

Lemma 2.1 (Federer, 1969)**.** *Let g, h be two composable Lipschitz functions. Then $g \circ h$ is also Lipschitz with $\mathrm{Lip}(g \circ h) \leq \mathrm{Lip}(g)\,\mathrm{Lip}(h)$.*

Corollary 2.1. For a fully-connected network (`FCN`) or a convolutional neural network (`CNN`) $f = W_K \circ \rho_{K-1} \circ W_{K-1} \circ \ldots \circ \rho_1 \circ W_1$, we have $\mathrm{Lip}_p(f) \leq \prod_k \|W_k\|_p$ under a choice of p-norm with 1-Lipschitz non-linearities ρ_k.

The above bound is not necessarily tight; there are various works that compute tighter bounds for `FCN` and `CNN` (e.g. Virmaux & Scaman, 2018; Fazlyab et al., 2019; Latorre et al., 2020).

3. Lipschitz Constant of Self-Attention

3.1. Dot-product self-attention is *not* Lipschitz

Moving on, we investigate whether self-attention is Lipschitz. We first consider the widely used *(scaled) dot-product multihead self-attention* as formulated by Vaswani et al. (2017). Let $\mathbf{x}_1, \ldots, \mathbf{x}_N$ be a sequence of N elements, where

$\mathbf{x}_i \in \mathbb{R}^D$ for $i = 1, \ldots, N$. We represent this sequence as a matrix X:

$$X = \begin{bmatrix} — & \mathbf{x}_1^\top & — \\ & \vdots & \\ — & \mathbf{x}_N^\top & — \end{bmatrix} \in \mathbb{R}^{N \times D}, \tag{3}$$

Dot-product multihead self-attention (DP−MHA) is a map from $\mathbb{R}^{N\times D}$ to $\mathbb{R}^{N\times D}$ consisting of H 'heads', where H is chosen to divide D. Each head is a map from $\mathbb{R}^{N\times D}$ to $\mathbb{R}^{N\times D/H}$ defined by

$$\begin{aligned} DP(X) &\coloneqq \text{softmax}\left(\frac{XW^Q(XW^K)^\top}{\sqrt{D/H}}\right) XW^V \\ &= PXW^V, \end{aligned}$$

where $W^Q, W^K, W^V \in \mathbb{R}^{D\times D/H}$ are learnable parameters specific to each head, and $P \in \mathbb{R}^{N\times N}$ is the output of the softmax (we suppress the dependence of P on X to reduce clutter below). The input to the softmax is an $N \times N$ matrix of pairwise dot products (hence *dot-product* self-attention), and the softmax is applied to each row of this matrix. Finally, the outputs of all heads are concatenated into an $N \times D$ matrix and are right multiplied by $W^O \in \mathbb{R}^{D\times D}$, thus DP−MHA is defined by

$$MHA_{DP}(X) \coloneqq \left[DP^1(X), \ldots, DP^H(X)\right] W^O. \tag{4}$$

In what follows, we will prove that MHA as defined above is *not* Lipschitz, assuming that the MHA map is non-trivial, i.e. $W^Q, W^K, W^V, W^O \neq 0$. It is sufficient to show that a single head DP is not Lipschitz, since MHA is a linear combination of the outputs of each head. Also note that P is a stochastic matrix, i.e. its entries are non-negative and its rows sum to 1. Since the rows of X are the $\mathbf{x}_i$'s, a linear transformation of each $\mathbf{x}_i$ by some matrix A is equivalent to right multiplication of X by $A^\top$. So right multiplication of X by W^V is a linear map and thus Lipschitz. Therefore, we are interested in the mapping $f(X) = PX$; this is *not* a linear mapping because P itself is a non-linear function of X. In fact, we show that f is *not* Lipschitz, thus proving the first main result of the paper:

Theorem 3.1. DP−MHA *is not Lipschitz for any vector p-norm $\|\cdot\|_p$ with $p \in [1, \infty]$.*

Summary of Proof. We use Theorem 2.1, noting that if the supremum of the norm of the Jacobian is infinite, then the mapping is not Lipschitz. In particular, we show that when $\mathbf{x}_i = \mathbf{0}$ for some i, some elements of the Jacobian of f grow proportionally to the sample variance of $\mathbf{x}_{\neq i}$, which is unbounded.

Proof. We show the proof for the case $D = H = 1$ (i.e. $X \in \mathbb{R}^{N\times 1}$, a column vector, and $x_i \in \mathbb{R}$) for readability. See Appendix C for the general case, which follows the same logic.

The mapping f can be written as

$$f(X) = PX = \text{softmax}\left(aXX^\top\right) X = \begin{bmatrix} f_1(X) \\ \vdots \\ f_N(X) \end{bmatrix} \in \mathbb{R}^{N\times 1},$$

$$\text{where} \quad f_i(X) = \sum_{j=1}^{N} P_{ij} x_j \in \mathbb{R}$$

and $a = W^K W^Q \in \mathbb{R}$ (we assume $a \neq 0$ such that self-attention is non-trivial). Hence f can be interpreted as a map of each x_i to a point in the convex hull of $x_1, ..., x_N$. Since f is a map from $\mathbb{R}^{N\times 1}$ to $\mathbb{R}^{N\times 1}$, its Jacobian is

$$J_f = \begin{bmatrix} J_{11} & \ldots & J_{1N} \\ \vdots & \ddots & \vdots \\ J_{N1} & \ldots & J_{NN} \end{bmatrix} \in \mathbb{R}^{N\times N}, \tag{5}$$

where $J_{ij} = \frac{\partial f_i(X)}{\partial x_j} \in \mathbb{R}$. By taking partial derivatives we can show that

$$J_{ij} = aX^\top P^{(i)} \left[E_{ji}X + \delta_{ij}X\right] + P_{ij}I$$

where

- $E_{ij} \in \mathbb{R}^{N\times N}$ is a binary matrix with zeros everywhere except the (i, j)th entry
- $\delta_{ij} \in \{0, 1\}$ is the Kronecker delta
- $P^{(i)} \coloneqq \text{diag}(P_{i:}) - P_{i:}^\top P_{i:} \in \mathbb{R}^{N\times N}$.

See Appendix A for useful identities in deriving the above Jacobian.

So for $i = j$:

$$J_{ii} = aX^\top P^{(i)} e_{ii} X + aX^\top P^{(i)} X + P_{ii} \tag{6}$$

Let us investigate the scalar $X^\top P^{(i)} X$. We observe that it is in fact a variance of a discrete distribution. Specifically:

$$X^\top P^{(i)} X = \textstyle\sum_k P_{ik} x_k^2 - \left(\sum_k P_{ik} x_k\right)^2 = \text{Var}(\mathbb{X}), \tag{7}$$

where $\mathbb{X}$ is a discrete distribution with support at the inputs $\{x_1, \ldots, x_N\}$ and probability mass function given by their softmax probabilities $\mathbb{P}(\mathbb{X} = x_j) = P_{ij}$. A consequence of this interpretation is that $P^{(i)}$ is *positive semi-definite* (PSD) since $X^\top P^{(i)} X = \text{Var}(\mathbb{X}) \geq 0$, with equality if and only if the x_j are all equal.

We use this observation to show that J_{ii} is unbounded, and so $\|J_f\|_p$ is unbounded, hence DP−MHA is *not* Lipschitz. Consider the case $x_i = 0$. Then

$$P_{i:}^\top = \text{softmax}\left(XAx_i\right) = \frac{1}{N}\mathbb{1},$$

i.e. we have uniform attention regardless of $x_{\neq i}$. The first term of J_{ii} in Equation (6) disappears since $e_{ii}X = [0, \dots, x_i, \dots, 0] = \mathbf{0}$, and the last term becomes $\frac{1}{N}I$. Now consider the second term $aX^\top P^{(i)}X = a\mathrm{Var}(\mathbb{X}_l)$. Note $\mathbb{X}$ is uniformly distributed, since $\mathbb{P}(\mathbb{X} = x_j) = P_{ij} = 1/N$. Hence the second term is equal to a times the sample variance of $x_1, \dots, x_N$, which can be arbitrarily large. Hence J_{ii} can become arbitrarily large, so the full Jacobian J_f is unbounded. □

High-level intuition for proof. At $x_i = 0$, $f_i(X) = \frac{1}{N}\sum_k x_k$, the mean of the inputs. The rate of change of f_i is governed by how fast the softmax saturates when x_i is perturbed, which is determined by how spread out the $x_{\neq i}$ are. The more spread out they are (the higher the sample variance), the greater the rate of saturation of the softmax, and the faster the rate of change of f_i. Since the sample variance of $x_{\neq i}$ can be arbitrarily large, the rate of change of f_i can also be arbitrarily large, i.e. the entries of the Jacobian (and hence its p-norm) can become arbitrarily large. In Appendix D, we show that adding bias terms to $\mathbf{x}_i^\top W^Q$ and $\mathbf{x}_j^\top W^K$ does *not* resolve the issue.

The implications of this result are the following. (1) There can be undesirable behaviour (e.g. training instabilities) for the Transformer when some inputs are close to zero and others have large magnitude. (2) Dot-product self-attention (and hence the standard Transformer) is not a suitable choice when we require a Lipschitz neural network, such as for formulating invertible residual networks (Behrmann et al., 2019). Therefore, to use self-attention and Transformers in such applications, a Lipschitz formulation of self-attention is required, together with an explicit (ideally tight) upper bound to its Lipschitz constant, to quantify how much the output can change with respect to changes in the input.

One method to make dot-product self-attention Lipschitz is by ensuring its inputs are bounded. Indeed, if the input space is compact, e.g. $[0, 1]^{N\times D}$, any continuously differentiable function is Lipschitz, including dot-product self-attention. However, as we further discuss in Section 6, such an approach has its own challenges, since it makes the Lipschitz constant depend on the input range. Instead, in the next section we formulate a version of self-attention that is provably Lipschitz on all of $\mathbb{R}^{N\times D}$, allowing us to derive an upper bound that holds for any subset of $\mathbb{R}^{N\times D}$.

3.2. L2 self-attention: a Lipschitz formulation of self-attention

The pathology in dot-product self-attention arises because the softmax probabilities $P_{i:}$ are constant with respect to $\mathbf{x}_{\neq i}$ when $\mathbf{x}_i = 0$. This behaviour can be undesirable as we want P_{ij} to vary according to $\mathbf{x}_j$, regardless of whether $\mathbf{x}_i$ is zero or not. Hence we propose an alternative form of self-attention based on L2 distance:

$$P_{ij} \propto \exp(L_{ij}) := \exp\left(-\frac{\left\|\mathbf{x}_i^\top W^Q - \mathbf{x}_j^\top W^K\right\|_2^2}{\sqrt{D/H}}\right), \tag{8}$$

with the normalisation constant ensuring that $\sum_j P_{ij} = 1$. We will refer to it as *L2 self-attention*. It is reminiscent of the standard squared-exponential kernel, but with softmax normalisation that ensures that each row of the kernel matrix sums to 1. Normalisation is usually necessary to deal with inputs of varying length N (Wang et al., 2018), hence we keep the softmax for L2 self-attention. Similarly to dot-product self-attention, L2 self-attention can be computed efficiently with matrix operations; see Appendix E for details, with a comparison of wall-clock runtimes between different choices of attention.

We first state the mathematical formulation of L2 multihead self-attention (`L2-MHA`) before proving the main result — the upper bound of its Lipschitz constant with respect to $\|\cdot\|_p$ for $p = 2, \infty$. The full `L2-MHA` map $F : \mathbb{R}^{N\times D} \to \mathbb{R}^{N\times D}$ is defined as

$$F(X) := \left[f^1(X)W^{V,1}, \dots, f^H(X)W^{V,H}\right] W^O$$
$$\text{where} \quad f^h(X) := P^h X A_h.$$

In the above, $W^{V,h} \in \mathbb{R}^{D\times D/H}$, $W^O \in \mathbb{R}^{D\times D}$, P^h is defined as in Equation (8) with $W^{Q,h} = W^{K,h} \in \mathbb{R}^{D\times D/H}$, and $A_h := W^{Q,h}{W^{Q,h}}^\top/\sqrt{D/H} \in \mathbb{R}^{D\times D}$. There are two changes from the usual form of multihead self-attention:

(1) We require $W^{Q,h} = W^{K,h}$ for each head $f^h(X)$ to be Lipschitz. In Lemma F.1 of Appendix F we show that `L2-MHA` is *not* Lipschitz for arbitrary $W^{Q,h}$, $W^{K,h}$, and that tying $W^{Q,h} = W^{K,h}$ is sufficient for `L2-MHA` to be Lipschitz, with intuition for why tying is sufficient.

(2) In each head of the self-attention $f^h(X)$, right multiplication by A_h has been included for the theorem below to hold (details are in the proof). In practice, there is little harm done by this extra linear transformation, since when the heads are combined together in F, each $f^h(X)$ is additionally transformed by $W^{V,h}$, a free parameter.

The second main result of the paper is the following:

Theorem 3.2. `L2-MHA` *is Lipschitz, with the following bound on* $\mathrm{Lip}_\infty(F)$*:*

$$\mathrm{Lip}_\infty(F) \leq \left(4\phi^{-1}(N-1) + \frac{1}{\sqrt{D/H}}\right)\|{W^O}^\top\|_\infty \max_h \|W^{Q,h}\|_\infty \|{W^{Q,h}}^\top\|_\infty \max_h \|{W^{V,h}}^\top\|_\infty$$

and the following bound on $\mathrm{Lip}_2(F)$*:*

$$\mathrm{Lip}_2(F) \leq \frac{\sqrt{N}}{\sqrt{D/H}} \left(4\phi^{-1}(N-1)+1\right) \left(\sqrt{\sum_h \|W^{Q,h}\|_2^2 \|W^{V,h}\|_2^2}\right) \|W^O\|_2$$

where $\phi(x) := x\exp(x+1)$ *is an invertible univariate function on* $x > 0$*, and* N *is the input sequence length.*

Specifically, $\phi^{-1}(N-1) = W_0(\frac{N}{e})$ *where* W_0 *is the Lambert* W*-function, which grows sub-logarithmically as* $O(\log N - \log\log N)$ *(Corless et al., 1996). Hence the above bounds can be simplified to* $O(\log N)$ *for* $p = \infty$ *and* $O(\sqrt{N}\log N)$ *for* $p = 2$*.*

Proof. See Appendix F, which uses the key observation that $X^\top P^{(i)} X$ is a covariance matrix (c.f. Equation (7)) to bound $\|J_F\|_p$, the norm of the Jacobian of F. Appendix G shows how the argument can be modified to prove the analogous result for the case with masking in the self-attention. □

These bounds are complemented by the concurrent work of Vuckovic et al. (2020), which provides a $O(\sqrt{D\log N})$ bound on $\mathrm{Lip}_1(F)$ using measure-theoretic tools.

4. Application: Invertible Self-Attention

4.1. Invertible residual network

Consider the residual function $g(x) := \mathbf{x} + f(\mathbf{x})$. Behrmann et al. (2019) give the following sufficient condition for its invertibility: if f is a *contraction* with respect to some metric, i.e. if $\mathrm{Lip}(f) < 1$, and the metric space on which f is defined is complete, then g is invertible. (A Euclidean space with a metric induced by a p-norm $\|\cdot\|_p$ for $p \in [1,\infty]$ is always complete.) Specifically, the inverse $g^{-1}(\mathbf{y})$ is the unique fixed point of the recursion $\mathbf{x}^{i+1} := \mathbf{y} - f(\mathbf{x}^i)$, since by the definition of the inverse we have $\mathbf{y} = g^{-1}(\mathbf{y}) + f(g^{-1}(\mathbf{y}))$. Because f is a contraction, *Banach's Fixed Point Theorem* guarantees that this fixed point exists and is unique for all $\mathbf{y}$, and that the recursion converges for all initial values $\mathbf{x}^0$ (often set to $\mathbf{y}$ in practice) exponentially fast. Hence the inverse can be computed to arbitrary accuracy (up to numerical precision in practice) by the above fixed-point iteration.

Note that a composition of such invertible residual blocks is also invertible. Behrmann et al. (2019) use this observation to design invertible ResNets: they take f to be a `CNN` normalised by an upper bound on $\mathrm{Lip}(f)$ given by Corollary 2.1, making the resulting function *contractive*. For the 2-norm $\|\cdot\|_2$, a hyperparameter $c < 1$ is chosen and each linear map (convolution) W in the `CNN` is multiplied by $c/\|W\|_2$ if $c < \|W\|_2$ where $\|W\|_2$ is estimated by power iteration (c.f. Appendix B). This multiplicative factor determines the scale of the Lipschitz constant of the normalised function.

4.2. Invertible self-attention

The standard use case of self-attention is with a skip connection inside the Transformer. A Transformer block is composed of residual blocks of multihead self-attention (`MHA`) and fully-connected (`FCN`) layers (Figure 1). Hence similarly to invertible ResNets, we can normalise `L2-MHA` by the upper bounds given in Theorem 3.2 to obtain `Contractive-L2-MHA` f, with which we can obtain invertible self-attention $g(\mathbf{x}) = \mathbf{x} + f(\mathbf{x})$. Since `Dropout` is also part of the residual branch along with `Contractive-L2-MHA`, we should check that it is also contractive. At test time, `Dropout` multiplies inputs by the dropout keep probability $p < 1$, so it is a contraction with Lipschitz constant p at evaluation time. At training time, `Dropout` amounts to setting some inputs to zero, while keeping other inputs constant. This can be expressed as right multiplication by a diagonal binary matrix M, and for such matrices we can verify $\|M\|_p := \sup_{\|x\|_p=1} \|Mx\|_p \leq 1$. Notice that `LayerNorm` is not part of the residual branch, hence its Lipschitz continuity is not relevant for invertibility; rather, we can replace it with an invertible normalisation such as `ActNorm` (Kingma & Dhariwal, 2018). However, architectures that place `LayerNorm` inside the residual branch (termed `pre-LN` as opposed to the traditional `post-LN` in Figure 1) have become more prevalent in the literature (Wang et al., 2019; Xiong et al., 2020), and in this case it makes sense to investigate its Lipschitz continuity. We show that `LayerNorm` is Lipschitz in Appendix N, with a bound on its Lipschitz constant.

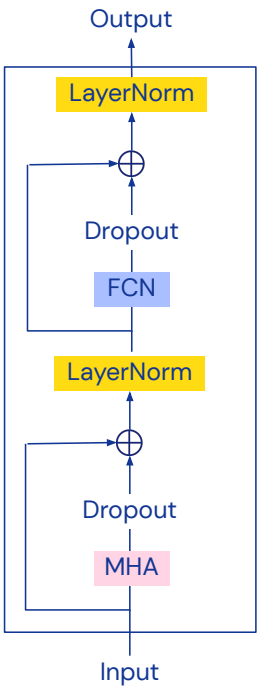

Figure 1. Transformer block.

In the next section, we investigate the properties of invertible self-attention and how it compares with the standard dot-product self-attention; we replace `DP-MHA` in the Transformer with `Contractive-L2-MHA`, hence replacing the residual self-attention module with invertible self-

attention. We are not interested in the modified Transformer per se, but rather in comparing the properties of invertible self-attention to standard self-attention — we only use the Transformer as a testbed for this purpose, since self-attention is commonly used in a Transformer. Given the theoretical focus of the paper, we believe that a more challenging application of invertible self-attention, such as normalising flow-based modelling, would be more suitable as a separate paper focused on that particular application.

5. Experimental Results

5.1. Asymptotic tightness of the upper bound on $\mathrm{Lip}_\infty(F)$

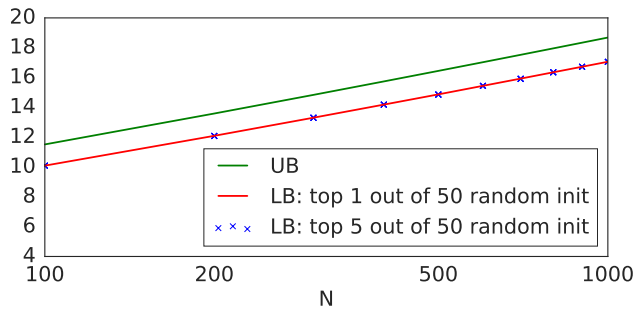

Figure 2. Lower and upper bound on $\mathrm{Lip}_\infty(f)$ for L2-MHA f, with $H = D = 1$ and varying N.

A tight bound on the Lipschitz constant of self-attention is desirable for all listed applications in Section 1; it leads to tighter generalisation bounds, lighter constraints for provable robustness, and better expressiveness in residual flow models. Hence we investigate the tightness of our bound on the Lipschitz constant of `L2-MHA`. The Lipschitz constant is a supremum over the space of inputs $X \in \mathbb{R}^{N\times D}$ (c.f. Equation (2)) and approximating it requires solving an intractable optimisation problem. Hence it is infeasible to estimate accurately in general, especially when X is high-dimensional. However, we may compute a lower bound on the Lipschitz constant by maximising the norm of the Jacobian $\|J_f(X)\|$ with respect to X until convergence. This local optimum will form a lower bound by Theorem 2.1, and we can expect this lower bound to be fairly tight for the low-dimensional case, provided the optimisation is thorough.

We use this observation to provide empirical evidence for the asymptotic tightness of the upper bound on $\mathrm{Lip}_\infty(f)$ in Theorem 3.2. In Figure 2, we show the upper bound as well as the lower bound on $\mathrm{Lip}_\infty(f)$ obtained by optimising $\|J_f(X)\|_\infty$ with respect to X for `L2-MHA` f with 50 different random initialisations of X, with $H = D = 1$ and N varying between 100 and 1000. See Appendix H for further details. Note that we use a log-scale for the x-axis, and recall that the upper bound is $O(\log N - \log\log N)$, dominated by the $O(\log N)$ term for large N. Hence the plot for the upper bound shows a linear trend. We also observe that the slope of the lower bound is very similar, providing empirical evidence that the $O(\log N - \log\log N)$ upper bound is asymptotically tight.

There are at least two possible explanations for the gap between the upper and lower bounds. (1) The lower bound is only a local optimum — the true Lipschitz constant is a global optimum across inputs, which can be difficult to attain especially for high values of N. (2) The multiplicative constant of the upper bound may be loose. Assuming asymptotic tightness, it remains an open question whether the multiplicative constant can be tightened. We show the analogous plot for $\mathrm{Lip}_2(F)$ and discuss the results in Appendix J. Additionally in Appendix K, we show that optimising $\|J_f(X)\|_\infty$ w.r.t. X for `DP-MHA` f causes the norm to diverge, providing empirical verification of Theorem 3.1, that `DP-MHA` is indeed *not* Lipschitz.

5.2. Numerical invertibility of MHA residual map

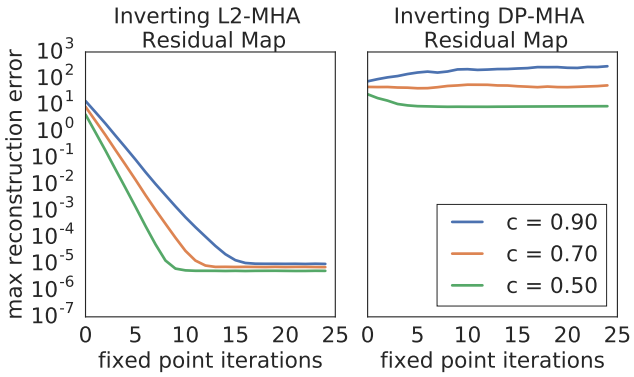

Figure 3. Invertibility of $g(\mathbf{x}) = \mathbf{x} + cf(\mathbf{x})$ where f is L2-MHA (left) and DP-MHA (right).

Recall from Section 4.1 that $g(\mathbf{x}) = \mathbf{x} + f(\mathbf{x})$ is invertible if f is contractive. Hence if f is `Contractive-L2-MHA`, g is necessarily invertible. However, technically we do not disprove the invertibility of `DP-MHA`, since the converse does not hold in general i.e. if f is `DP-MHA`, which we have shown is *not* Lipschitz hence *not* contractive, it may still be the case that g *is* invertible. To verify that `DP-MHA` (with the skip connection) is *not* invertible in practice, we compare the numerical invertibility of the residual map $g(\mathbf{x}) = \mathbf{x}+cf(\mathbf{x})$ between the cases where f is `L2-MHA` and `DP-MHA` in Figure 3. For each, we take `MHA` with 8 heads and randomly initialised weights, and quantify the maximum reconstruction error across a batch of 128 inputs whose outputs are inverted via the fixed-point iteration described in Section 4.1. We use $N = 64$, $D = 64$, and $c \in \{0.5, 0.7, 0.9\}$ (see Appendix I for analogous results for a wider range of N

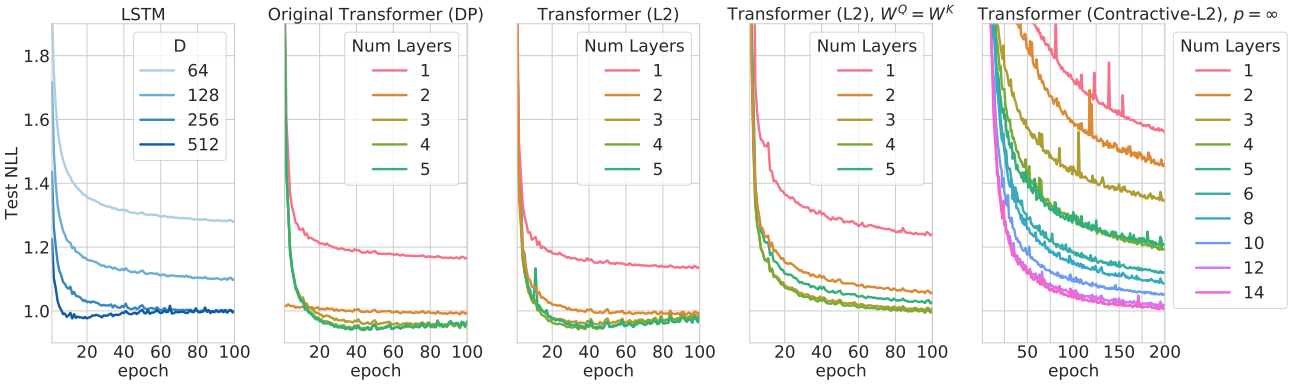

Figure 4. Test NLL curves during training for various LSTM/Transformer models on PTB character level language modelling.

and D and for DP−MHA with trained weights). To highlight the difference between the two types of self-attention, recall in the proof of Theorem 3.1 (showing that DP−MHA is not Lipschitz) that when one of the inputs $\mathbf{x}_i$ is 0, some terms of the Jacobian grow with the sample variance of $\mathbf{x}_{\neq i}$. Hence we check numerical invertibility at a set of N inputs where $\mathbf{x}_i = 0$ and $\mathbf{x}_{\neq i}$ are chosen uniformly at random.

In Figure 3, we see that DP−MHA is *not* invertible whereas L2−MHA *is* invertible for sufficiently small c. This shows how not having the theoretical guarantee of f being contractive can cost us invertibility in practice. We note that the figure shows local invertibility at the sampled inputs, as opposed to global invertibility across the whole input space, yet this clearly highlights the difference between the two choices of self-attention. Experiments with the globally invertible self-attention obtained by normalising with the Lipschitz upper bound are provided in the next section.

5.3. Expressiveness of L2-MHA and invertible self-attention

A natural question to ask is: how does the expressiveness of L2−MHA and Contractive−L2−MHA (that leads to invertible self-attention with the skip connection) compare with the original DP−MHA? We expect that the Lipschitz constraint will limit the expressiveness of the Transformer, and would like to find out by how much. We investigate this by comparing the performance of the original Transformer and the Transformer with invertible self-attention (c.f. Figure 1) at character-level language modelling on the Penn Treebank dataset (Marcus et al., 1993). We compare the test negative log-likelihood (NLL) of a baseline LSTM, the original Transformer (DP−MHA), and a series of models between the original Transformer and the Transformer with invertible self-attention (Contractive−L2−MHA), making one change at a time and tuning the hyperparameters on a validation set. For Contractive−L2−MHA, we normalise $F =$L2−MHA by the bound on $\mathrm{Lip}_\infty(F)$ as it is tighter than the bound on $\mathrm{Lip}_2(F)$. During training we backpropagate through these contractive blocks $F/\mathrm{Lip}_\infty(F)$ (including the denominator) to update the model parameters. We found that only backpropagating through the numerator (i.e. applying stop−gradient to denominator) gave slightly worse performance. See Appendix H for experimental details.

The results are shown in Figure 4. The first plot shows the best performing LSTM reaching a test NLL of around 1.0, and the second plot shows the best performing Transformer reaching a slightly improved performance for 3–5 layers of Transformer blocks. We observe instabilities in training for a higher number of layers, requiring careful tuning of the learning rate schedule for stability at the cost of performance, a commonly observed phenomenon in the literature of deep Transformer architectures (Bapna et al., 2018; Parisotto et al., 2020). The third plot shows results for the Transformer with DP−MHA replaced with L2−MHA but without tying W^Q and W^K, and we observe a very similar test performance. The fourth plot shows the change when we further tie the query and key weights (making $W^Q = W^K$); we see that there is a small degradation in performance. Here the number of trainable parameters has been reduced, but in Appendix L we show that matching parameter count does not help performance, suggesting that the reduction in performance when tying queries and keys is not solely due to having fewer parameters. We note that performance saturates at around 5 layers for each Transformer model so far. On the rightmost plot we show results when further dividing self-attention in each block by the upper bound on $\mathrm{Lip}_\infty(F)$, to obtain invertible self-attention. This does give reduced performance for the same number of layers, but we can attain similar performance with more layers, no longer saturating at 5 layers.

Thus we conclude the following. (1) Replacing the dot-product with the L2 distance incurs hardly any loss in expressiveness. (2) Tying the query and key weights to obtain Lipschitz self-attention incurs a small loss in expressiveness.

Number of Layers	2	4	6	8	10	12	14	16	18
Transformer (**DP**)	1.061	1.032	1.021	**1.017**	1.025	-	-	-	-
Transformer (**L2**), $W^Q = W^K$	1.168	1.040	1.023	1.024	1.019	**1.008**	1.018	1.027	1.034
Transformer (**Contractive-L2**)	1.246	1.135	1.103	1.079	1.072	1.060	1.039	**1.029**	1.031

Table 1. Test NLL for Transformer models trained with **fixed learning rate** on PTB character level language modelling.

(3) Dividing by the upper bound on $\mathrm{Lip}_\infty(F)$ to obtain invertible self-attention incurs a noticeable loss in expressiveness, but also has a stabilising effect on the optimisation of the Transformer, thus allowing one to compensate for the apparent loss in expressiveness by increasing the number of layers.

5.4. Training Stability of DP-MHA vs L2-MHA

In Figure 5, we compare the output variance of trained `L2-MHA` against trained `DP-MHA`, with weights from the one-layer Transformer (L2), $W^Q = W^K$ model and (DP) model used for Figure 4 respectively. We take the same distribution of inputs as used for the numerical invertibility experiment in Section 5.2, and show the histogram of inputs and outputs after flattening the input/output tensors. We see that the range of outputs remains similar to the range of inputs for Lipschitz `L2-MHA`, whereas for `DP-MHA` the outputs have a much wider range, because the Jacobian norm is large for `DP-MHA` at these inputs.

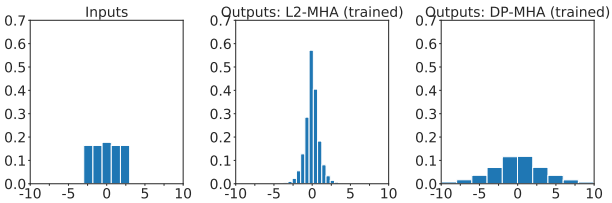

Figure 5. Histogram showing distribution of inputs/outputs of trained L2-MHA and DP-MHA

In practice, this leads to instabilities in training for `DP-MHA`, hence requiring careful tuning of the learning rate schedule for training deeper Transformer models: linear warmup and square root decay, as detailed in Appendix H. We investigate the behaviour of the different Transformer models on the above PTB task when using a **fixed learning rate**. We observe that `DP-MHA` fails to train at all beyond 10 layers, whereas both `L2-MHA` ($W^Q = W^K$) (i.e. Lipschitz L2-MHA but not contractive) and `Contractive-L2-MHA` shows stable training for up to 18 layers (see Appendix M for the training curves). This was the deepest model we could fit on a single GPU, and we expect to be able to train even deeper models with these two. In Table 1 we show the best Test NLL across training for each of the Transformer models. Note that for `DP-MHA` training becomes unstable beyond 10 layers, so we are only able to provide results up to 10 layers. The generalisation performance of the best model for each setting of self-attention is similar.

6. Conclusion and Discussion

We have shown that the widely used dot-product self-attention is *not* Lipschitz, and that the proposed L2 self-attention *is* Lipschitz, by deriving an $O(\log N - \log\log N)$ Lipschitz bound for $p = \infty$ and an $O(\sqrt{N}(\log N - \log\log N))$ bound for $p = 2$, where N is the input sequence length. We also provided empirical evidence of the asymptotic tightness of the bound for $p = \infty$. We demonstrated that Lipschitz-constrained self-attention can be used to formulate invertible self-attention, which we experimentally evaluated on a character-level language modelling task. And finally, we also showed that L2-MHA is more stable during training, allowing the use of fixed learning rate for stable training of deep architectures.

Our approach to Lipschitz self-attention has been to replace the dot-product kernel with an L2 kernel. An alternative would be to constrain the inputs of self-attention to be bounded; if the input space is compact, e.g. $[0,1]^{N\times D}$, *any* continuously differentiable function is Lipschitz, including dot-product self-attention. However, while being simple to implement, this solution has its own difficulties. First, it makes the Lipschitz constant depend on the range of the input, and thus obtaining a tight bound would require non-trivial mathematical work. We stress that a guarantee that the function is Lipschitz does not tell us anything about its Lipschitz constant; without a tight Lipschitz bound, the true Lipschitz constant can be very large, at which point it is unhelpful that the function is Lipschitz. Second, since self-attention is typically applied at multiple layers within a model (e.g. Transformer), the input to each self-attention will live in a different compact set that depends on the parameters of the previous layers, complicating the analysis for subsequent layers. A solution is to constrain the inputs of each layer to be in the same compact set, e.g. by passing them through a sigmoid non-linearity. This however can have undesirable side effects such as vanishing gradients when the sigmoids are saturated. Despite these difficulties, this could be a worthwhile alternative route for obtaining Lipschitz self-attention to explore in the future.

Having a provably Lipschitz self-attention module at our disposal makes it possible to use Transformer-based ar-

chitectures in applications requiring Lipschitz constraints, while enjoying theoretical guarantees. A natural application of Lipschitz self-attention is for residual flows (Behrmann et al., 2019), and for parameterising Neural ODEs (Chen et al., 2018) where a Lipschitz vector field guarantees the existence of a unique solution to the ODE for all times. These models can be used for density estimation and generative modelling of sets. Another interesting direction for future work would be to analyse different variants of self-attention based on kernels other than dot-product and L2, as (Tsai et al., 2019) do from an experimental perspective, for which we believe the mathematical tools developed in this paper may aid the analysis.

Acknowledgements

We would like to thank Adam Kosiorek, Arnaud Doucet, Yee Whye Teh, Michalis Titsias, Emilien Dupont and Theophane Weber for helpful discussion and feedback.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

Anil, C., Lucas, J., and Grosse, R. Sorting out Lipschitz function approximation. In *International Conference on Machine Learning*, pp. 291–301, 2019.

Bapna, A., Chen, M. X., Firat, O., Cao, Y., and Wu, Y. Training deeper neural machine translation models with transparent attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3028–3033, 2018.

Behrmann, J., Grathwohl, W., Chen, R. T. Q., Duvenaud, D., and Jacobsen, J.-H. Invertible residual networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pp. 6571–6583, 2018.

Chen, R. T. Q., Behrmann, J., Duvenaud, D., and Jacobsen, J.-H. Residual flows for invertible generative modeling. In *Advances in Neural Information Processing Systems*, 2019.

Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 854–863, 2017.

Corless, R. M., Gonnet, G. H., Hare, D. E., Jeffrey, D. J., and Knuth, D. E. On the Lambert W function. *Advances in Computational mathematics*, 5(1):329–359, 1996.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4171–4186, 2019.

Fazlyab, M., Robey, A., Hassani, H., Morari, M., and Pappas, G. Efficient and accurate estimation of Lipschitz constants for deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 11423–11434, 2019.

Federer, H. *Geometric Measure Theory*. Classics in Mathematics. Springer Berlin Heidelberg, 1969. ISBN 9783642620102.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Grathwohl, W., Chen, R. T. Q., Betterncourt, J., Sutskever, I., and Duvenaud, D. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations*, 2019.

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Simon, I., Hawthorne, C., Shazeer, N., Dai, A. M., Hoffman, M. D., Dinculescu, M., and Eck, D. Music Transformer. In *International Conference on Learning Representations*, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pp. 10215–10224, 2018.

Latorre, F., Rolland, P., and Cevher, V. Lipschitz constant estimation of neural networks via sparse polynomial optimization. In *International Conference on Learning Representations*, 2020.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. Building a large annotated corpus of English: The Penn

Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

Mises, R. and Pollaczek-Geiringer, H. Praktische verfahren der gleichungsauflösung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 9(2):152–164, 1929.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for Generative Adversarial Networks. In *International Conference on Learning Representations*, 2018.

Parisotto, E., Song, H. F., Rae, J. W., Pascanu, R., Gulcehre, C., Jayakumar, S. M., Jaderberg, M., Kaufman, R. L., Clark, A., Noury, S., Botvinick, M. M., Heess, N., and Hadsell, R. Stabilizing Transformers for reinforcement learning. In *International Conference on Machine Learning*, 2020.

Parmar, N., Ramachandran, P., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. Stand-alone self-attention in vision models. In *Advances in Neural Information Processing Systems*, pp. 68–80, 2019.

Peyré, G. and Cuturi, M. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6): 355–607, 2019.

Sokolić, J., Giryes, R., Sapiro, G., and Rodrigues, M. R. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280, 2017.

Tsai, Y.-H. H., Bai, S., Yamada, M., Morency, L.-P., and Salakhutdinov, R. Transformer dissection: An unified understanding for Transformer's attention via the lens of kernel. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 4344–4353, 2019.

Tsuzuku, Y., Sato, I., and Sugiyama, M. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 6541–6550, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Virmaux, A. and Scaman, K. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems*, pp. 3835–3844, 2018.

Vuckovic, J., Baratin, A., and Tachet des Combes, R. A mathematical theory of attention. *arXiv preprint arXiv:2007.02876*, 2020.

Wang, Q., Li, B., Xiao, T., Zhu, J., Li, C., Wong, D. F., and Chao, L. S. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1810–1822, 2019.

Wang, X., Girshick, R., Gupta, A., and He, K. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, 2018.

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., and Liu, T. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pp. 10524–10533. PMLR, 2020.

Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. Self-attention Generative Adversarial Networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 7354–7363, 2019.