Attention is not *all* you need: pure attention loses rank doubly exponentially with depth

Yihe Dong [1] **Jean-Baptiste Cordonnier** [2] **Andreas Loukas** [3]

Abstract

Attention-based architectures have become ubiquitous in machine learning. Yet, our understanding of the reasons for their effectiveness remains limited. This work proposes a new way to understand self-attention networks: we show that their output can be decomposed into a sum of smaller terms—or paths—each involving the operation of a sequence of attention heads across layers. Using this path decomposition, we prove that self-attention possesses a strong inductive bias towards "token uniformity". Specifically, without skip connections or multi-layer perceptrons (MLPs), the output converges doubly exponentially to a rank-1 matrix. On the other hand, skip connections and MLPs stop the output from degeneration. Our experiments verify the convergence results on standard transformer architectures.

1. Introduction

The attention mechanism (Bahdanau et al., 2015) was initially developed to better learn long-range sequential knowledge, and found effective use in transformer networks (Vaswani et al., 2017). Since then, attention-based architectures have permeated across data domains machine learning applications, such as in natural language processing (Devlin et al., 2018), speech recognition (Luo et al., 2020), and computer vision (Ramachandran et al., 2019; Bello et al., 2019). As such, it is vital to develop tools to understand the inner workings of transformers and attention in general, both to shed light on existing models, and to design more effective future models.

This work provides new insights about the operation and inductive bias of networks built by stacking multiple self-attention layers. Surprisingly, we find that *pure* self-attention networks (SANs), i.e., transformers with skip connections and multi-layer perceptrons (MLPs) disabled, lose expressive power *doubly exponentially* with respect to network depth. More specifically, we prove that the output converges with a cubic rate to a rank one matrix that has identical rows. While we derive the convergence bounds in part by using properties of stochastic matrices, our results go beyond what one would expect based on standard results. In particular, by leveraging the cascading effects of specifically stacking self-attention modules, we show exponentially faster convergence than what standard theory prescribes. Furthermore, while previous studies have considered the rank of individual self-attention matrices (Wang et al., 2020; Katharopoulos et al., 2020; Cordonnier et al., 2020b), our results are the first to address conditions under which the *entire* network converges to rank *one*.

This raises the question, why do transformers work? Our analysis indicates that skip connections play a key role in mitigating rank collapse, and MLPs can slow down the convergence by increasing their Lipschitz constant. We characterize these counteracting forces by proving upper bounds of this convergence behavior under SAN architectural variants that resemble transformers. Our results reveal a previously unknown vital utility of skip connections, beyond facilitating optimization and gradient flow (He et al., 2016a; Balduzzi et al., 2018).

In the process, we develop a new *path decomposition* to study self-attention networks. Namely, we decompose a SAN into a linear combination of weakly-interdependent *paths*, where each 'path' corresponds to a deep single-head SAN. Intuitively, one can view the self-attention heads in each layer of the original network as different gateways, and a path follows a sequence of gateway choices, one gateway per layer (Figure 1). Coupled with the rank collapse analysis, our results suggest that deep SANs with skip connections should rely more on short paths.

Our main contributions are as follows: (1) We present a systematic study of building blocks of the transformer, revealing opposing impacts between self-attention and the counteracting forces: skip connections and MLP, in con-

[1]Google. yihed@google.com. [2]EPFL. jean-baptiste.cordonnier@epfl.ch. [3]EPFL. andreas.loukas@epfl.ch.
[1,3] Equal contribution.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

tributing and preventing a *rank collapse* in transformers. As a corollary, this reveals a previously unknown vital effect of skip connections beyond facilitating optimization. (2) We propose a new method for analyzing SANs via a *path decomposition*, revealing SANs as an ensemble of shallow networks. (3) We verify our theory with experiments on common transformers architectures.[1]

Notation. In this work, bold-face letters denote vectors (lower-case) and matrices (upper-case). We denote the ℓ_1, ℓ_∞-composite norm of a matrix $\boldsymbol{X}$ as $\|\boldsymbol{X}\|_{1,\infty} = \sqrt{\|\boldsymbol{X}\|_1 \|\boldsymbol{X}\|_\infty}$. We note that $\ell_{1,\infty}$ is not a proper norm as it does not satisfy the triangle inequality, though it is absolutely homogeneous and positive definite. We also use the shorthand notation $[H] = (1, \cdots, H)$.

2. Attention doubly exponentially loses rank

We start by studying self-attention networks (SANs) built exclusively out of multi-head self-attention layers. We prove that SANs converge exponentially (with depth) to a rank-1 matrix that makes all tokens identical.

Our analysis in §2.1 relies on an unconventional way to express the output of a multi-head SAN as a sum of single-head networks. We refer to the latter as *paths*, where each path is denoted by a sequence of attention heads. Intuitively, one can view the attention heads in a transformer layer as different gateways, and a path follows a sequence of gateway choices, one gateway per layer (see Figure 1). A proof sketch of why rank collapse occurs is given in §2.2, whereas the main rank collapse result is presented in §2.3.

2.1. The path decomposition argument

Let $\boldsymbol{X}$ be a $n \times d_{in}$ input tensor consisting of n tokens. An SAN is built out of L multi-head self-attention layers, each having H heads. The output of the h-th self-attention head can be written as

$$\mathrm{SA}_h(\boldsymbol{X}) = \boldsymbol{P}_h \boldsymbol{X} \boldsymbol{W}_{V,h} + \mathbf{1}\boldsymbol{b}_{V,h}^\top,$$

where $\boldsymbol{W}_{V,h}$ is a $d_{in} \times d_v$ value weight matrix and the $n \times n$ *row-stochastic* matrix $\boldsymbol{P}_h$ is given by

$$\begin{aligned}\boldsymbol{P}_h &= \mathrm{softmax}\big(d_{qk}^{-\frac{1}{2}} (\boldsymbol{X}\boldsymbol{W}_{Q,h} + \mathbf{1}\boldsymbol{b}_{Q,h}^\top)(\boldsymbol{X}\boldsymbol{W}_{K,h} + \mathbf{1}\boldsymbol{b}_{K,h}^\top)^\top\big) \\ &= \mathrm{softmax}(d_{qk}^{-\frac{1}{2}} (\boldsymbol{X}\boldsymbol{W}_{QK,h}\boldsymbol{X}^\top + \mathbf{1}\boldsymbol{b}_{Q,h}^\top \boldsymbol{W}_{K,h}^\top \boldsymbol{X}^\top)),\end{aligned}$$

where the key and query weight matrices $\boldsymbol{W}_{K,h}$ and $\boldsymbol{W}_{Q,h}$ are of size $d_{in} \times d_{qk}$, whereas $\boldsymbol{W}_{QK,h} = \boldsymbol{W}_{Q,h}\boldsymbol{W}_{K,h}^\top$. The softmax operates independently on each row of its input. We obtain the final equation by noting that softmax is shift-invariant and disregarding terms that provide a constant contribution across rows (Cordonnier et al., 2020b).

The output of each SAN layer is formed by concatenating the individual outputs of all H attention heads (along the last dimension) and linearly projecting them onto a subspace of appropriate size:

$$\begin{aligned}\mathrm{SA}(\boldsymbol{X}) &= \mathbf{1}[\boldsymbol{b}_{O,1}^\top, \cdots, \boldsymbol{b}_{O,H}^\top] + \\ &\quad [\mathrm{SA}_1(\boldsymbol{X}), \cdots, \mathrm{SA}_H(\boldsymbol{X})]\,[\boldsymbol{W}_{O,1}^\top, \cdots, \boldsymbol{W}_{O,H}^\top]^\top \\ &= \sum_{h\in[H]} \boldsymbol{P}_h \boldsymbol{X} \boldsymbol{W}_h + \mathbf{1}\boldsymbol{b}_O^\top,\end{aligned}$$

where we set $\boldsymbol{W}_h = \boldsymbol{W}_{V,h}\,\boldsymbol{W}_{O,h}^\top$ and $\boldsymbol{b}_O = \sum_h \boldsymbol{b}_{O,h}$ and $[H] = [1, \ldots, H]$.

Let $\boldsymbol{X}^l$ be the output of the l-th layer and fix $\boldsymbol{X}^0 = \boldsymbol{X}$. As is common practice, we let all layers consist of the same number of heads.

Excluding biases $\mathbf{1}\boldsymbol{b}_{O,h}^\top$, the SAN output is given by

$$\begin{aligned}\boldsymbol{X}^L &= \sum_{h\in[H]} \boldsymbol{P}_h^L \boldsymbol{X}^{L-1} \boldsymbol{W}_h^L \\ &= \sum_{h\in[H]} \boldsymbol{P}_h^L \Bigg(\sum_{h'\in[H]} \boldsymbol{P}_{h'}^{L-1} \boldsymbol{X}^{L-2} \boldsymbol{W}_{h'}^{L-1} \Bigg) \boldsymbol{W}_h^L \\ &= \sum_{h_L, h_{L-1}\in[H]^2} \boldsymbol{P}_{h_L}^L \boldsymbol{P}_{h_{L-1}}^{L-1} \boldsymbol{X}^{L-2} \boldsymbol{W}_{h_{L-1}}^{L-1} \boldsymbol{W}_{h_L}^L,\end{aligned}$$

which, after unrolling the recursion backwards, yields:

$$\boldsymbol{X}^L = \sum_{h_1,\ldots,h_L\in[H]^L} (\boldsymbol{P}_{h_L}^L \cdots \boldsymbol{P}_{h_1}^1)\, \boldsymbol{X}\, (\boldsymbol{W}_{h_1}^1 \cdots \boldsymbol{W}_{h_L}^L).$$

The above equations have a clear interpretation if we think of the SAN as a directed acyclic graph, with nodes corresponding to self-attention heads and directed edge connecting heads of consecutive layers.

We formalize this intuition in the following:

Theorem 2.1 (Path decomposition of SAN)**.** *The output of a depth L self-attention network with H heads per layer (including biases and skip connections) is given by*

$$\mathit{SAN}(\boldsymbol{X}) = \sum_{\mathit{path}\in([H]\cup\{0\})^L} \boldsymbol{P}_{\mathit{path}}\, \boldsymbol{X}\, \boldsymbol{W}_{\mathit{path}} + \boldsymbol{1}\boldsymbol{b}^\top, \quad (1)$$

where $\boldsymbol{P}_{\mathit{path}} = \boldsymbol{P}_{h_L}^L \cdots \boldsymbol{P}_{h_1}^1$ is an input-dependent stochastic matrix, whereas $\boldsymbol{W}_{\mathit{path}} = \boldsymbol{W}_{h_1}^1 \cdots \boldsymbol{W}_{h_L}^L$ and $\boldsymbol{b}$ do not depend on the input.

[1] Our code is publicly available at https://github.com/twistedcubic/attention-rank-collapse.

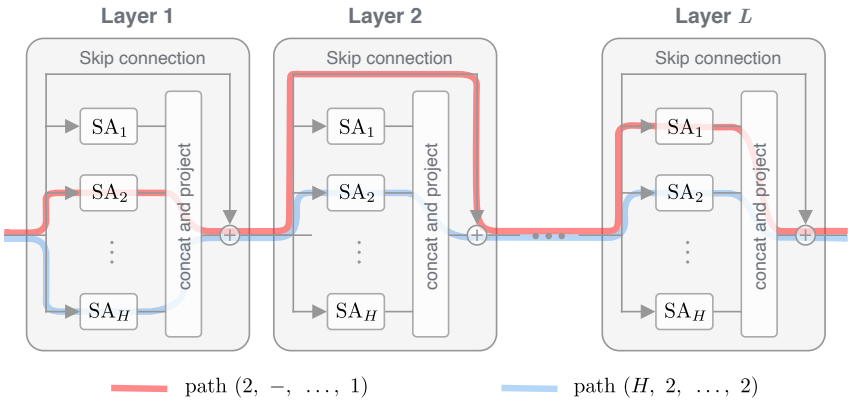


Figure 1: Two paths in a deep Self-Attention Network (SAN) with H heads and L layers. At each layer, a path can go through one of the heads or bypass the layer. Adding an MLP block after each attention layer forms the transformer architecture.

Proof. The proof follows from the fact that the set of row-stochastic matrices is closed under multiplication (i.e., $\boldsymbol{P}^L_{h_L} \cdots \boldsymbol{P}^i_{h_i}$ is row-stochastic) and, moreover, for any row-stochastic matrix $\boldsymbol{P}$, we have $\boldsymbol{P}\mathbf{1} = \mathbf{1}$. □

Each of the terms in (1) then describes a path across heads of different layers:

$$path = (h_1, \ldots, h_L), \quad \text{where} \quad h_l \in (0, 1, \ldots, H).$$

There are a total of $(H+1)^L$ such paths, where each path has length equal to the number of nonzero indices on that path. The path decomposition thus describes the action of a multi-head SAN as the combination of simpler single-head networks. To gain intuition on path interdependence, it helps to split the operations performed into two types: those that act across tokens (multiplication from left) and those that apply independently on each token (multiplication from right). As seen, though paths can interact through token mixing (since P_{path} matrices jointly depend on X), token-wise operations are independent. We can also notice that biases are not particularly meaningful: their total contribution amounts to the single term $\mathbf{1}\boldsymbol{b}^\top$ independently of the number of layers or heads used.

In the following, we show that each path converges rapidly (as a function of length) to a rank-1 matrix with identical rows. This convergence is so dominant so that adding more layers to the SAN does not help: though the number of paths is increased exponentially, each path degenerates doubly exponentially, leading also to a rank-1 output.

2.2. Convergence of single-head SAN

Before tackling the full SAN, it is instructive to consider the behavior of each path separately. We examine, in particular, how the residual

$$\text{res}(\boldsymbol{X}) = \boldsymbol{X} - \mathbf{1}\boldsymbol{x}^\top, \quad \text{where} \quad \boldsymbol{x} = \text{argmin}_{\boldsymbol{x}} \|\boldsymbol{X} - \mathbf{1}\boldsymbol{x}^\top\|$$

changes during the forward pass.

As the following theorem shows, the residual norm of a single path converges to zero surprisingly quickly with respect to its length (doubly exponential with a cubic rate):

Theorem 2.2. *For any single-head SAN consisting of L layers with $\|\boldsymbol{W}^l_{QK}\|_1 \|\boldsymbol{W}^l_V\|_{1,\infty} \le \beta$, without skip connections, we have that*

$$\|res(SAN(\boldsymbol{X}))\|_{1,\infty} \le \left(\frac{4\beta}{\sqrt{d_{qk}}}\right)^{\frac{3^L-1}{2}} \|res(\boldsymbol{X})\|^{3^L}_{1,\infty}, \quad (2)$$

which amounts to a doubly exponential convergence to a rank-1 matrix.

Note that the bound in Eq 2 guarantees $\|\text{res}(\text{SAN}(\boldsymbol{X}))\|_{1,\infty}$ convergence for all inputs of small residual whenever $4\beta < \sqrt{d_{qk}}$. In practice, our experiments imply that the region for convergence is much greater.

The identified cubic rate of convergence is significantly faster than what would be expected when analyzing products of stochastic matrices (linear rate). As a rule of thumb, to achieve a decline of three orders of magnitude, say from 1000 to 1, one could expect a linear rate of convergence to require roughly a dozen iterations, whereas a cubic rate can do so in just two or three iterations. The reason why we get a cubic rate is that the rank of attention matrices depends also on the rank of the input. As we show, the self-attention heads mix tokens faster when formed from a low-rank matrix. This phenomenon becomes stronger as we build deeper SANs, leading to a cascading effect.

Proof sketch. To analyze how the formation of $\boldsymbol{P}_h$ is affected by the rank of the input, we start by writing $\boldsymbol{X} =$

$\mathbf{1}\boldsymbol{x}^\top + \boldsymbol{R}$ for $\boldsymbol{R} = \text{res}(\boldsymbol{X})$ and expanding the attention matrix accordingly:

$$\boldsymbol{X}\boldsymbol{W}_{QK}\boldsymbol{X}^\top = \left(\mathbf{1}\boldsymbol{x}^\top + \boldsymbol{R}\right)\boldsymbol{W}_{QK}\left(\mathbf{1}\boldsymbol{x}^\top + \boldsymbol{R}\right)^\top$$

Invoking once more the shift-invariance property of the softmax operator, the above equation can be simplified to

$$\boldsymbol{P}_h = \text{softmax}(\boldsymbol{R}\frac{\boldsymbol{W}_{QK}}{\sqrt{d_{qk}}}\boldsymbol{R}^\top + \mathbf{1}\boldsymbol{r}^\top),$$

for some appropriate $\boldsymbol{r}$. Observe that if the matrix within the softmax was $\mathbf{1}\boldsymbol{r}^\top$, then $\boldsymbol{P}_h$ would also degenerate to a rank-1 matrix: $\text{softmax}(\mathbf{1}\boldsymbol{r}^\top) = \mathbf{1}\,\boldsymbol{q}^\top$ and the convergence would happen instantly.

The proof builds on this observation by showing that if $\boldsymbol{E} = \boldsymbol{R}\frac{\boldsymbol{W}_{QK}}{\sqrt{d_{qk}}}\boldsymbol{R}^\top$ is small then $\boldsymbol{P}_h$ is almost rank-1:

$$\|\boldsymbol{P}_h - \mathbf{1}\boldsymbol{q}^\top\| \leq 2\,\|\boldsymbol{D}\,\mathbf{1}\,\boldsymbol{q}^\top\|,$$

where $\boldsymbol{D}$ is diagonal and $\boldsymbol{D}_{ii} = \max_j |\boldsymbol{\delta}_i^\top \boldsymbol{E}(\boldsymbol{\delta}_j - \boldsymbol{\delta}_{j'})|$. Thus, we have

$$\boldsymbol{P}_h\boldsymbol{X} = \boldsymbol{P}_h(\mathbf{1}\boldsymbol{x}^\top + \boldsymbol{R}) = \mathbf{1}\boldsymbol{x}^\top + \text{softmax}(\mathbf{1}\boldsymbol{r}^\top + \boldsymbol{E})\boldsymbol{R}$$

and, moreover, $\|\text{res}(\boldsymbol{P}_h\boldsymbol{X})\| \leq 2\,\|\boldsymbol{D}\mathbf{1}\,\boldsymbol{q}^\top\boldsymbol{R}\|$. The proof concludes by bounding the above term and applying the argument recursively over successive layers. □

2.3. Exponential convergence for attention networks

We now move on to analyze the convergence of SANs with *multiple* heads per layer.

Our main result is as follows:

Theorem 2.3. *In a depth-L and width-H self-attention network without skip connections, let $\|\boldsymbol{W}^l_{QK,h}\|_1\|\boldsymbol{W}^l_h\|_{1,\infty} \leq \beta$ for all heads $h \in [H]$ and layers $l \in [L]$, then:*

$$\|res(SAN(\boldsymbol{X}))\|_{1,\infty} \leq \left(\frac{4\,\beta\,H}{\sqrt{d_{qk}}}\right)^{\frac{3^L-1}{2}} \|res(\boldsymbol{X})\|^{3^L}_{1,\infty},$$

which amounts to a doubly exponential rate of convergence.

The bound guarantees convergence of $\text{SAN}(\boldsymbol{X})$ to rank one when $4\beta H < \sqrt{d_{qk}}$. Our experiments show that this is a rather pessimistic estimate, as, in practice, we observe widespread convergence of output to rank-1.

Remark 1. Implications for Xformers. There has been a surge of architectural variants –that we collectively refer to as Xformers– aimed to improve the vanilla transformer (Vaswani et al., 2017) by reducing the quadratic self-attention complexity. The rank collapse result of Theorem 2.3 carries interesting implications for these architectures. One such variant relies on low-rank or kernel-based approximations to the full attention matrix (Katharopoulos et al., 2020; Wang et al., 2020; Choromanski et al., 2020), in which case the paths likely converge even faster to rank one due to the imposed low-rankness. Another variant only computes a subset of the attention matrix entries using particular patterns (Zaheer et al., 2020; Child et al., 2019), such as random patterns, in which case one expects the paths to converge more slowly, as randomization tends to increase the rank of the output.

3. Mechanisms that counteract convergence

Our findings raise a pertinent question—why do attention-based networks work in practice if attention degenerates to a rank-1 matrix doubly exponentially with depth? Aiming to obtain a deeper understanding, we focus on the transformer architecture (Vaswani et al., 2017) and expand our analysis by incorporating the three important components of transformers that SANs lack: *skip connections*, *multi-layer perceptrons*, and *layer normalization*.

We adopt a methodical approach where the modifications to the SAN architecture are introduced one at a time. For each case, we re-derive the convergence bounds and discuss the observed effect.

3.1. Skip connections are crucial

A simple modification to the path decomposition argument for SAN suffices to take into account skip connections. Specifically, we indicate the event that a path has skipped a layer by setting $h = 0$ on the corresponding notation:

$$\begin{aligned}\boldsymbol{X}^L &= \sum_{h\in[H]\cup\{0\}} \boldsymbol{P}^L_h \boldsymbol{X}^{L-1}\boldsymbol{W}^L_h \\ &= \dots \\ &= \sum_{h_1,\dots,h_L\in([H]\cup\{0\})^L} (\boldsymbol{P}^L_{h_L}\cdots\boldsymbol{P}^1_{h_1})\,\boldsymbol{X}\,(\boldsymbol{W}^1_{h_1}\cdots\boldsymbol{W}^L_{h_L}),\end{aligned}$$

where we have fixed $\boldsymbol{P}_0 = \boldsymbol{I}$ and $\boldsymbol{W}_0 = \boldsymbol{I}$.

As observed, skip connections dramatically diversify the path distribution. Denote by $\mathcal{P}_l$ the set of paths of length l. With skip connections enabled, we have

$$|\mathcal{P}_l| = \binom{L}{l} H^l$$

paths of length l (whereas before we had only length L paths). We hypothesize that it is the presence of short paths that stops SAN from degenerating to rank-1. While we

can derive an upper bound for the residual similar to above (which we do in the Appendix for completeness) such an upper bound is vacuously large. Indeed, it is more informative to have a *lower* bound on the residual, to align with practice, where SANs with skip connections do not suffer rank collapse. We present the following simple lower bound:

Claim 3.1. *Consider a depth-L and width-H self-attention network with skip connections. There exist infinitely many parameterizations for which $res(\boldsymbol{X}^L) \geq res(\boldsymbol{X})$. The preceding holds even for $L \to \infty$ and β arbitrarily small.*

The proof is elementary. By the path decomposition, there is always a path that skips all layers, i.e. the path with length 0, preserving the residual. It then follows that, for any parametrization that renders the contribution of the SAN layers orthogonal to the input, we will have $\text{res}(\boldsymbol{X}^L) \geq \text{res}(\boldsymbol{X})$. A simple example of such a parametrization can be recovered by setting $\boldsymbol{W}_V^l = 0$ for every $l \in [L]$, in which case $\|\text{res}(\boldsymbol{X}^L)\| = \|\text{res}(\boldsymbol{X})\|$.

A tight lower bound to the residual in the presence of skip connections is highly nontrivial, and we pose it as an open challenge to the community.

Remark 2. SANs as ensembles of shallow networks. It can be deduced from Theorem 2.3 that SANs with skip connections enabled heavily rely on short paths (since the residual rapidly declines as the path length becomes larger). In other words, SANs behave like ensembles of shallow single-head self-attention networks. The phenomenon was previously identified for ResNets (Veit et al., 2016b) (though the latter study didn't study the rank-collapse phenomenon). Here, the components of this ensemble are inter-dependent, as each attention head participates in many paths of different lengths. Experimental results in §4 support this implication. The supplementary material also provides a study of the path length distribution across several common architectures.

3.2. Multi-layer perceptrons (MLPs) help

We now study how using an MLP affects the residual. In particular, we focus on SANs with layers written as

$$\boldsymbol{X}^{l+1} = f_l\left(\sum_{h\in[H]} \boldsymbol{P}_h \boldsymbol{X}^l \boldsymbol{W}_h\right).$$

Note that, to keep the notation compact, we use f_l to denote both the MLP as well as the output bias.

In our subsequent analysis, we use $\lambda_{l,1,\infty}$ to denote the Lipschitz constant of f_l with respect to $\ell_{1,\infty}$ measure. Note that, though finding the exact constant can be NP-hard even for shallow MLPs (Scaman & Virmaux, 2018), since f_l comprises of linear transformations with Lipschitz non-linearities, f_l is generally Lipschitz.

Corollary 3.2 (SAN with MLP)**.** *Consider a depth-L and width-H SAN with MLP. Moreover, let $\|\boldsymbol{W}_{QK,h}^l\|_1 \|\boldsymbol{W}_h^l\|_{1,\infty} \leq \beta$ for all $h \in [H]$ and $l \in [L]$ and fix $\lambda_{l,1,\infty} \leq \lambda$. We then have that*

$$\|res(\boldsymbol{X}^L)\|_{1,\infty} \leq \left(\frac{4\,\beta\,H\,\lambda}{\sqrt{d_{qk}}}\right)^{\frac{3^L-1}{2}} \|res(\boldsymbol{X})\|_{1,\infty}^{3^L}, \quad (3)$$

which amounts to a doubly exponential rate of convergence.

As seen, though the effect of MLP is less drastic than that of skip connections, the convergence rate in Cor 3.2 can be controlled by the Lipschitz constants $\lambda_{f,1,\infty}$ of the MLPs: the more powerful the MLPs are the slower the convergence becomes. This reveals a tug-of-war between the self-attention layers and the the MLPs, which due to their non-linearity can increase the rank. §4 shows that indeed MLPs counteract convergence in experiments.

However, while increasing the Lipschitz constants slows down residual convergence, it carries the side effect of making the MLPs more sensitive to input perturbations, and thus is often associated with less robust models (Cranko et al., 2018). Furthermore, larger Lipschitz constants pose greater challenges to model optimization, as they lead to larger gradient variance.

3.3. Layer normalization plays no role

Layer normalization is accomplished by rescaling and shifting the input across the feature dimension:

$$\begin{aligned}\text{LN}(\text{SA}(\boldsymbol{X})) &= \text{LN}\left(\sum_{h\in[H]} \boldsymbol{P}_h \boldsymbol{X} \boldsymbol{W}_h + \boldsymbol{1}\boldsymbol{b}_O^\top\right) \\ &= \left(\sum_{h\in[H]} \boldsymbol{P}_h \boldsymbol{X} \boldsymbol{W}_h + \boldsymbol{1}\boldsymbol{b}_O^\top - \boldsymbol{1}\boldsymbol{b}_{LN}^\top\right)\boldsymbol{D}_{LN}^{-1},\end{aligned}$$

where $\boldsymbol{b}_{\text{LN}}$ is the mean of each column $\text{SA}(\boldsymbol{X})$ and $\boldsymbol{D}_{LN}$ is a diagonal matrix with entries corresponding to the (possibly scaled or shifted) standard deviation of each column $\text{SA}(\boldsymbol{X})$.

By setting $\tilde{\boldsymbol{W}}_h = \boldsymbol{W}_h \boldsymbol{D}_{LN}^{-1}$ and $\tilde{\boldsymbol{b}}_O = \boldsymbol{b}_O - \boldsymbol{b}_{LN}$, the above is re-written as

$$\text{LN}(\text{SA}(\boldsymbol{X})) = \sum_{h\in[H]} \boldsymbol{P}_h \boldsymbol{X} \tilde{\boldsymbol{W}}_h + \boldsymbol{1}\tilde{\boldsymbol{b}}_O^\top,$$

which is identical to the equation before layer normalization was applied, though now $\tilde{\boldsymbol{W}}_h$ and $\tilde{\boldsymbol{b}}_O$ are input dependent.

Since right multiplication cannot increase the rank of a matrix, we conclude that layer normalization does not slow down the convergence.

4. Experiments

Our experiments first test the rank collapse results in several well-known transformers architectures (Section 4.1). We then visually illustrate the inductive bias of some architectural variants of transformers with a toy example in §4.2. Additional results can be found in the supplementary material.

4.1. Rank collapse in real architectures

To verify our theoretical predictions, we examine the residual of three well-known transformer architectures: BERT (Devlin et al., 2018), Albert (Lan et al., 2019), and XLNet (Yang et al., 2019). Figure 2 plots the relative residual $\|\text{res}(\text{SAN}(\boldsymbol{X}^l)\|_{1,\infty}/\|\text{SAN}(\boldsymbol{X}^l)\|_{1,\infty}$, of each layer's output before and after the networks have been trained. To compute these ratios we ran the network on 32 samples of 128 tokens excerpts of biographies from Wikipedia (Lebret et al., 2016) and display the mean and standard deviation.

The experiments confirm that, as soon as the skip connections are removed, all networks exhibit a rapid rank collapse. Though MLPs do not seem to help in the mitigation of convergence, we caution that the observation is not an accurate portrayal of how trained transformers behave: removing the skip connections introduces a drastic distribution shift in the MLP input. We expect that the convergence will slow down if the network is retrained.

4.2. Visualizing the bias of different architectures

To empirically investigate the inductive bias of the different components of the transformer architecture, we study the behavior of a single-layer transformer when applied *recurrently* (akin to the universal transformer (Dehghani et al., 2019)) to predict a simple 2D circular sequence. This is designed as a simple task to train different architectural variants from scratch, with visually intuitive results.

Specifically, we train a single-layer transformer to sequentially predict two circular arcs in $\mathbb{R}^2$ of radius 0.3, starting at $(-0.3, 0)$ and $(0.3, 0)$, respectively, each directed counterclockwise and consisting of 1000 points (illustrated as gray arcs in 3). An input sample consists of a sequence of *two opposing points* on the circle, one from the top arc and the other from the bottom arc. We apply teacher-forcing at each step, meaning we give the network the ground truth coordinates of the two current points, and train it to predict the next two points. The model attempts to minimize the MSE loss between the predicted points and the ground truth points on the trajectories. At inference time, we don't apply teacher-forcing, and simply feed the model output as input for the next step.

Since this recurrent application of a single-layer transformer can be reparametrized to be equivalent to a multi-layer transformer without skip connections, *we hypothesize that at inference time the predicted trajectories of the two arcs will converge to the same point (indicating a rank collapse)*, rather than following the training trajectories. Convergence of the two arcs implies rank collapse, as that means the two points in the predicted sequence have become uniform. Note that the setting has also been intentionally constructed to enable training even without skip connections (by using teacher forcing) and thus to disentangle the two distinct benefits of skip connections: their ability to improve optimization and their mitigation of rank collapse.

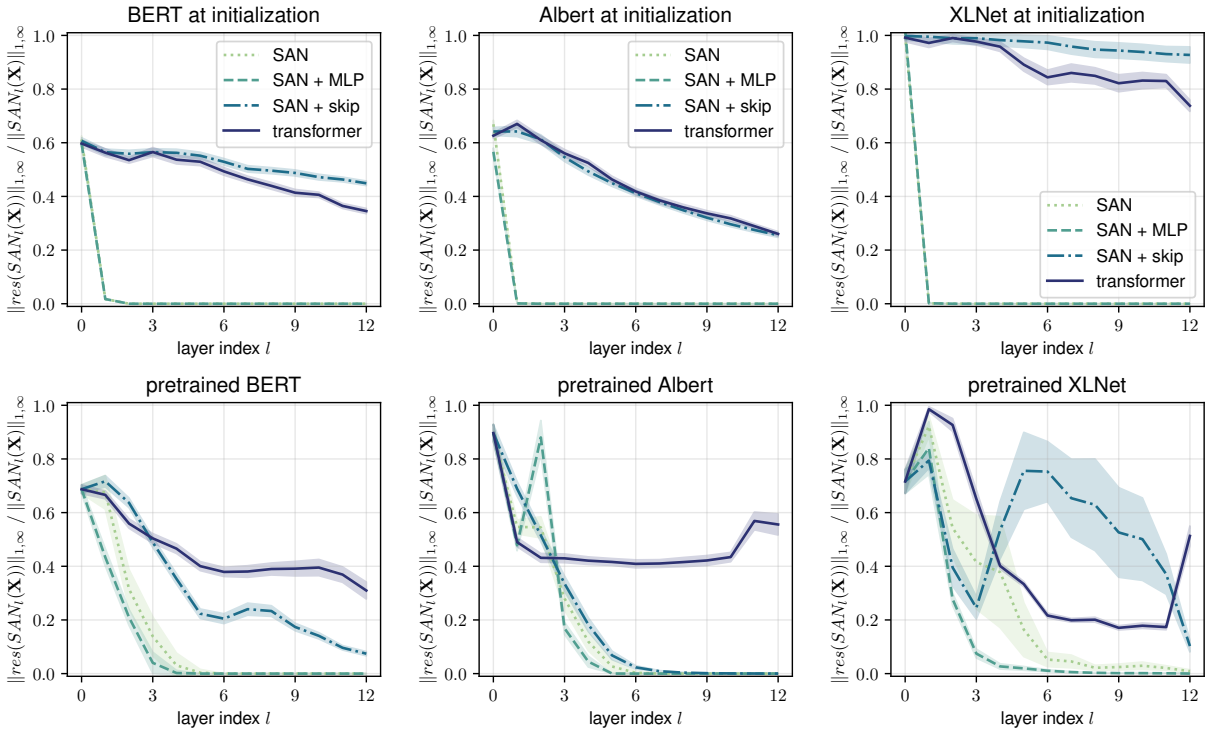
We trained the network until it could perfectly memorize the next step on the circular trajectories with near-zero loss. Figure 3 demonstrates the trajectories predicted at inference time (i.e., without teacher forcing). As seen on the top row, without MLP or skip connections the network exhibits rank collapse. Theorem 2.2 predicts that the convergence slows down when $\beta \geq \|\boldsymbol{W}_{QK}^l\|_1 \|\boldsymbol{W}_V^l\|_{1,\infty}$ increases. Indeed, as the hidden dimension increases from 32 to 128 (leading to larger β), the convergence slows down, becoming hardly observable for dimension 128. The supplementary material contains additional experiments showing that the observed effects are not artifacts of a larger model overfitting the training data, and indeed result from increasing β.

In accordance to our analysis, adding MLP or skip connections either stops or drastically slows down rank collapse. As observed, skip connections tend to slow down points from moving. The latter phenomenon is because in this setting skip connections introduce a bias towards remaining in the same position. On the other hand, adding MLPs does not exhibit the same bias.

4.3. Path effectiveness

SANs can be seen as ensembles of paths of different lengths (from 0 to L), each involving a different sequence of self-attention heads. Our analysis of SAN with skip connections indicates that the expressivity of a path decreases with its length, even if the number of non-linear operations involved increases. To test this hypothesis, we isolate paths of different lengths and evaluate their predictive power.

Tasks. We considered the following three tasks to test path effectiveness with respect to length:



(a) Bert (b) Albert (c) XLNet

Figure 2: Relative norm of the residual along the depth for three models before and after training. Pure attention (SAN) converges rapidly to a rank-1 matrix. Adding MLP blocks and skip connection gives a transformer. Skip connections play a critical role in mitigating rank collapse (i.e., a zero residual).

- *Sequence memorization.* The model learns to memorize a pre-determined mapping from natural language sentences and random label sequences of the same length. We use random tokens to make this purely a test of *expressiveness* of a network by way of *memorizing* training data, rather than confounding effects such as generalization. The cross entropy loss between predicted and the ground truth labels is minimized during training. The training data consist of 500 English sentences from Wikipedia and News sources (Dagan et al., 2006; Wang et al., 2019), tokenized using the SentencePiece tokenizer (Kudo & Richardson, 2018) into a vocabulary of size 30522 with 128 tokens per sequence. Each sequence is labeled with a random binary sequence of the same length.

- *Learning to sort.* Given an input sequence of letters, this task learns to sort the letters in alphabetical ordering (similar tasks have been studied before (Freivalds et al., 2019)). Specifically, the model's output for each input letter is used to determine the position of that letter in the predicted ordering. Each input sequence, of length 8, is created by sampling uniformly randomly, with replacement, from an alphabet of size 10. The training and test sets consist of 1000 and 200 sequences, respectively.

- *Convex hull prediction*. This task was inspired by the work of Vinyals et al. (2015). Given a sequence of N points uniformly distributed in $[0, 1] \times [0, 1]$ and shifted by a random bivariate standard normal, this task predicts the convex hull of these points. Specifically, for each point in the set, the model predicts whether it is part of the convex hull. The training set consists of $10,000$ sequences of points in $[0, 1] \times [0, 1]$, each of length 10.

In all three tasks, we report the test-set per-token label prediction accuracy as the evaluation metric. While we report results for the specified settings, we found that the path effectiveness trends are generally robust with respect to hyperparameter changes, such as the model depth, number of heads, and the difficulty of the task.

Path effectiveness test. We measure the effectiveness of individual paths by a *path disentanglement* procedure that we apply at inference time: the procedure isolates the weights involved and the output of an individual path $(\boldsymbol{P}^L_{h_L} \cdots \boldsymbol{P}^1_{h_1})\, \boldsymbol{X}\, (\boldsymbol{W}^1_{h_1} \cdots \boldsymbol{W}^L_{h_L})$ for *any* given sequence of

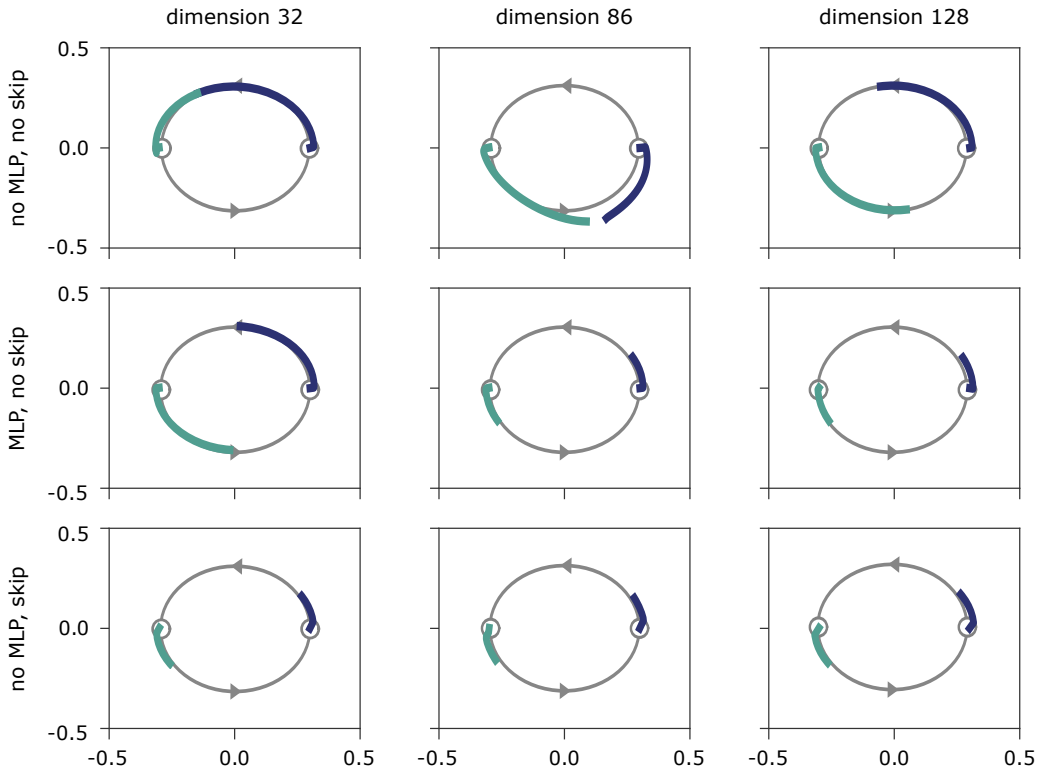


Figure 3: Applying a trained single-layer transformer module recurrently, to models of increasing $\beta \geq \|\boldsymbol{W}_{QK}^l\|_1 \|\boldsymbol{W}_V^l\|_{1,\infty}$ (horizontal direction) and across architectural variants (vertical direction). The two light background paths illustrate the two training trajectories, for which the starting points are $(-0.3, 0)$ and $(0.3, 0)$. Each figure contains the same number of steps. Consistent with the theory in §3, convergence slows down or stops as β increases, as well as when either MLP or skip connections are added.

heads $h_1, \cdots, h_L \in [H \cup 0]^L$. After the attention network has been successfully trained to solve each task, we use this procedure to determine the output of a randomly sampled set of paths of a given length. We then evaluate the task performance based solely on the normalized sum of this subset of paths (rather than from all paths). Note that the training remains unaltered and uses all heads simultaneously, therefore ensuring that each path learns to its full effectiveness.

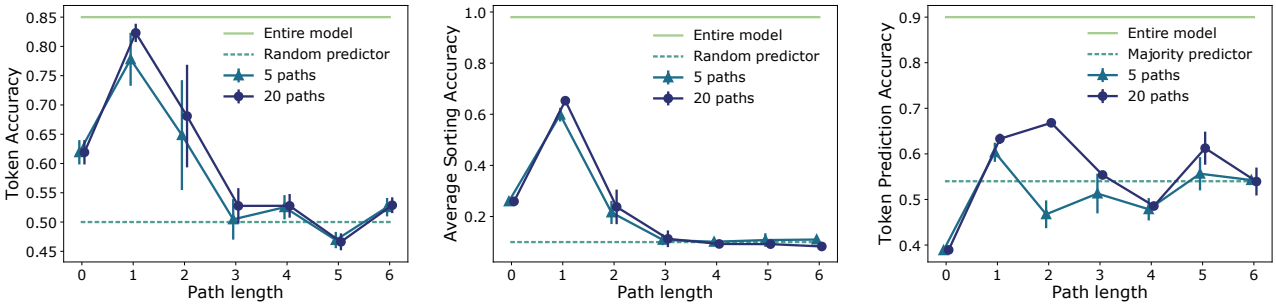
Figure 4 illustrates the resulting performance across all three tasks. We test different subset sizes and report the mean and standard deviation of five repetitions. For reference, we also plot the accuracy of a naive classifier, as well as that of the entire trained model including all paths. As observed, short paths carry predictive power; on the other hand, the output of longer paths is not much better than a random guess. In the convex hull task, since there is a class imbalance, we use a majority class predictor to obtain a random baseline. Though the difference in accuracy between short and long paths is less pronounced for the convex hull task, we observe that the variance of the long paths is noticeably larger.

The depths (L), number of heads (H), and hidden dimensions (d) for the three models are: L:6, H:2, d:250 for memorization, L:6, H:2, d:48 for sorting, and L:6, H:3, d:84 for convex hull. It's important to note that for all three tasks, while higher *peak* accuracies are attainable with increased model capacity and training time, our focus is to study the effects of path length on performance. Indeed, the trend for degenerating performance as path length increases stayed consistent across model sizes in all experiments.

The rapidly diminishing effectiveness of paths with respect to length indicates that attention networks rely almost exclusively on short paths. In other words, attention networks behave like an ensemble of *shallow* networks. Furthermore, the results indicate that there is *underutilized* capacity in long paths, and suggest that one way to make them, and hence the attention network, more effective, is to prevent the long paths from losing rank.

5. Related works

Skip connections were first introduced in ResNets (He et al., 2016a), ever since, it has been used to facilitate optimization in deep networks (He et al., 2016b; Veit et al., 2016a;



(a) Memorization (b) Sorting (c) Convex hull

Figure 4: To determine how much of the expressive power can be attributed to short vs long paths, we examine the performance of subsets of paths of different lengths (rather than of the entire SAN). Performance can be seen to consistently deteriorate with respect to path length, supporting our hypothesis that short paths are responsible for the majority of the expressive power.

Balduzzi et al., 2018). In particular, skip connections tackle the vanishing gradient problem, by allowing the gradient to flow bypass the skipped layers during backpropagation. The original motivation of using skip connections in transformers follow the same reasoning on facilitating optimization (Vaswani et al., 2017). With the paths decomposition for transformers, we discover an additional surprising importance of skip connections: they prevent the transformer output from degenerating to rank one exponentially quickly with respect to network depth.

Veit et al. (2016a) introduced an analogous interpretation for residual networks as a collection of paths of varying lengths, and found that the length of the effective paths in deep residual networks are much shorter than the total network depth, due to the gradients used for parameter updates coming overwhelmingly from these short paths. Our finding suggests that SANs rely on short paths to avoid rank collapse. Daneshmand et al. (2020) proved that batch normalization prevents rank collapse in randomly initialized deep linear networks, with certain assumptions. Interestingly, their work did not find skip connections to have a similar rank stabilizing effect under the settings studied.

Some recent works have approximated the attention matrix with low-rank factorizations (Wang et al., 2020; Tay et al., 2020) or kernel methods (Katharopoulos et al., 2020; Choromanski et al., 2020), to reduce the quadratic self-attention complexity. Are work is orthogonal to these works, by studying the rank of the network's output (rather than of the attention matrix).

There have been other recent advances in understanding the theory behind transformers: (Perez et al., 2019; Dehghani et al., 2019) proved Turing universality, (Cordonnier et al., 2020a) provided necessary and sufficient conditions for attention to simulate convolution. A linearized form of self-attention was also found to exhibit a depth phase transition (Levine et al., 2020); and the Lipschitz constant of self-attention was analyzed by (Kim et al., 2020).

Perhaps the convergence to rank one of a path should come as no surprise: each path component contains row-stochastic matrices as a result of the softmax attention, and (Anthonisse & Tijms, 1977) showed the exponential convergence of products of stochastic matrices to rank one. While the intuition behind stochastic matrices driving convergence still applies, in deep attention networks these matrices interact in more complex ways than what classical analyses consider. As we show, because of these interactions the rank collapses much faster than what would be expected based on classical analyses (cubic vs linear rate).

6. Conclusion

This work exposes competing forces over rank collapse in self-attention networks, namely self-attention vs skip connections and MLPs. In the process, we develop a path decomposition for SANs, which modularizes the study of self-attention and is of independent interest to additional applications. These results open the door for exciting future directions. For instance, how can one leverage the token-uniformity inductive bias revealed to design more effective networks, perhaps better at utilizing long paths? What are some practical implications for width-depth trade-off? How do we prove meaningful lower bounds of residue convergence for transformers? We believe that answering these questions will have broad implications in advancing the state of the art.

Acknowledgments. We would like to thank the anonymous reviewers for constructive feedback that improved the clarity of the paper. Andreas Loukas is supported by the Swiss National Science Foundation in the context of the project "Deep Learning for Graph-Structured Data" (grant #PZ00P2 179981). Jean-Baptiste Cordonnier is supported by the Swiss Data Science Center (SDSC).

References

Anthonisse, J. M. and Tijms, H. Exponential convergence of products of stochastic matrices. In *Journal of Mathematical Analysis and Applications*, 1977.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015.

Balduzzi, D., Frean, M., Leary, L., Lewis, J., Ma, K. W.-D., and McWilliams, B. The shattered gradients problem: If resnets are the answer, then what is the question?, 2018.

Bello, I., Zoph, B., Vaswani, A., Shlens, J., and Le, Q. V. Attention augmented convolutional networks. In *International Conference on Computer Vision*, 2019.

Child, R., Gray, S., Radford, A., and Sutskever, I. Generating long sequences with sparse transformers. 2019. URL arXiv:1904.10509.

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Davis, J., Sarlos, T., Belanger, D., Colwell, L., and Weller, A. Masked language modeling for proteins via linearly scalable long-context transformers. 2020. URL arXiv:2006.03555.

Cordonnier, J.-B., Loukas, A., and Jaggi, M. On the relationship between self-attention and convolutional layers. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=HJlnC1rKPB.

Cordonnier, J.-B., Loukas, A., and Jaggi, M. Multi-head attention: Collaborate instead of concatenate. 2020b. URL arXiv:2006.16362.

Cranko, Z., Kornblith, S., Shi, Z., and Nock, R. Lipschitz networks and distributional robustness. *arXiv preprint arXiv:1809.01129*, 2018.

Dagan, I., Glickman, O., and Magnini, B. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pp. 177–190, 2006.

Daneshmand, H., Kohler, J., Bach, F., Hofmann, T., and Lucchi, A. Batch normalization provably avoids rank collapse for randomly initialised deep networks, 2020.

Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., and Kaiser, Ł. Universal transformers. In *International Conference on Learning Representations*, 2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, 2018. URL arXiv:1810.04805.

Freivalds, K., Ozoliņš, E., and Šostaks, A. Neural shuffle-exchange networks - sequence processing in o(n log n) time. In Wallach, H., Larochelle, H., Beygelzimer, A., Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 6630–6641. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/9001ca429212011f4a4fda6c778cc318-Paper.pdf.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016b.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are RNNs: Fast autoregressive transformers with linear attention. 2020. URL arXiv:2006.16236.

Kim, H., Papamakarios, G., and Mnih, A. The lipschitz constant of self-attention. *arXiv preprint arXiv:2006.04710*, 2020.

Kudo, T. and Richardson, J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Empirical Methods in Natural Language Processing*, 2018.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Lebret, R., Grangier, D., and Auli, M. Generating text from structured data with application to the biography domain. *CoRR*, abs/1603.07771, 2016. URL http://arxiv.org/abs/1603.07771.

Levine, Y., Wies, N., Sharir, O., Bata, H., and Shashua, A. Limits to depth efficiencies of self-attention. *arXiv preprint arXiv:2006.12467*, 2020.

Luo, H., Zhang, S., Lei, M., , and Xie, L. Simplified self-attention for transformer-based end-to-end speech recognition. In *CoRR*, 2020. URL arXiv:2005.10463.

Perez, J., Marinkovic, J., and Barcelo, P. On the turing completeness of modern neural network architectures. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyGBdo0qFm.

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

Scaman, K. and Virmaux, A. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *arXiv preprint arXiv:1805.10965*, 2018.

Tay, Y., Bahri, D., Metzler, D., Juan, D.-C., Zhao, Z., , and Zheng, C. Synthesizer: Rethinking self-attention in transformer models. 2020. URL arXiv:2005.00743.

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

Veit, A., Wilber, M., and Belongie, S. Residual networks behave like ensembles of relatively shallow networks. In *Advances in Neural Information Processing Systems*, 2016a.

Veit, A., Wilber, M., and Belongie, S. Residual networks behave like ensembles of relatively shallow networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 550–558, 2016b.

Vinyals, O., Fortunato, M., and Jaitly, N. Pointer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pp. 2692–2700, 2015.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

Wang, S., Li, B., Khabsa, M., Fang, H., and Ma, H. Linformer: Self attention with linear complexity. 2020. URL arXiv:2006.04768.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32:5753–5763, 2019.

Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., and Ahmed, A. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, 2020. URL arXiv:2007.14062.