Which Transformer architecture fits my data? A vocabulary bottleneck in self-attention

Noam Wies [1] **Yoav Levine** [1] **Daniel Jannai** [1] **Amnon Shashua** [1]

Abstract

After their successful debut in natural language processing, Transformer architectures are now becoming the de-facto standard in many domains. An obstacle for their deployment over new modalities is the architectural configuration: the optimal depth-to-width ratio has been shown to dramatically vary across data types (*e.g.*, 10x larger over images than over language). We theoretically predict the existence of an embedding rank bottleneck that limits the contribution of self-attention width to the Transformer expressivity. We thus directly tie the input vocabulary size and rank to the optimal depth-to-width ratio, since a small vocabulary size or rank dictates an added advantage of depth over width. We empirically demonstrate the existence of this bottleneck and its implications on the depth-to-width interplay of Transformer architectures, linking the architecture variability across domains to the often glossed-over usage of different vocabulary sizes or embedding ranks in different domains. As an additional benefit, our rank bottlenecking framework allows us to identify size redundancies of $25\% - 50\%$ in leading NLP models such as ALBERT and T5.

1. Introduction

Since the introduction of the Transformer as a sequence-to-sequence model for machine translation, its variants have achieved state-of-the-art results in various domains, such as text (Devlin et al., 2019), images (Chen et al., 2020; Dosovitskiy et al., 2021; Jun et al., 2020), audio (Dhariwal et al., 2020; Baevski et al., 2020), video (Weissenborn et al., 2020), mathematical problem solving (Saxton et al., 2019), reinforcement learning (Vinyals et al., 2019; Chen et al., 2021) and bioinformatics (Rives et al., 2019; Rao et al., 2021). While the architecture's operation is mostly unchanged, the chosen ratio between the number of self-attention layers (depth) and the dimension of the internal representation (width) varies greatly across different applications. For example, for a fixed BERT-Base size of 110M parameters, popular architectures range from 12-layered networks to much narrower 128-layered networks.

In language applications, the depth-to-width ratio of Transformer models is relatively consistent: increase in model size is mainly done via widening (Levine et al., 2020), so that the largest self-attention architectures are very wide relative to their depths (Raffel et al., 2020; Brown et al., 2020). In other domains this aspect seems unresolved, even when comparing leading models within the same field. In computer vision, for example, the Vision Transformer (ViT) (Dosovitskiy et al., 2021) sets the state-of-the-art on ImageNet in a transfer learning setup with depth-to-width ratios corresponding to common language models. Conversely, Image GPT (Chen et al., 2020) and Sparse Transformer (Jun et al., 2020) achieve state-of-the-art results in unsupervised learning and density estimation, respectively, by using significantly deeper and narrower models.

Recently, Henighan et al. (2020) perform an ablation study which includes data from different domains, and report empirical evidence for the existence of different "ideal" depth-to-width ratios per data modality. Figure 4 in that study leads the authors to conclude that the depth-to-width ratio of image and math models should be 10x larger than that of language models. A possible take-away can be that the representations of the different data types require different depth-to-width specifications from the architecture. In a contemporary study, Levine et al. (2020) quantify the optimal depth-to-width ratio per self-attention network size. Their results justify the relative shallowness of current attention-based language models, and moreover suggest that further deepening should be logarithmic in widening. Importantly, their theoretical framework pertains to the self-attention architecture expressivity and is agnostic to the input modality.

The two different views above give rise to the following question: **does the optimal depth-to-width ratio depend on the data modality, or is it a consequence of architecture expressivity considerations?** In this paper, we establish architecture expressivity results that explain the ob-

[1] The Hebrew University of Jerusalem. Correspondence to: Noam Wies <noam.wies@cs.huji.ac.il>.

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

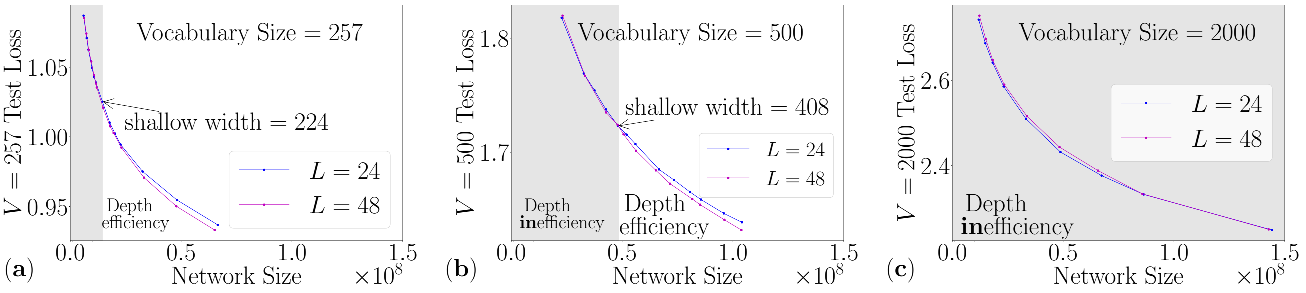

Figure 1. An experimental validation of the effect of vocabulary size on the depth-to-width tradeoff in transformers (full details in section 5.2). The experiments were performed over text, but demonstrate that similarly to images, when using a byte level vocabulary to represent text, deeper networks outperform shallower ones in practical network sizes (BERT-Base is of size $\sim$ 100M). Figures (a) and (b) use similar vocabulary sizes to those used by Henighan et al. (2020) and Jun et al. (2020) over images, respectively. Both studies operated with significantly deeper transformers than commonly used over text. Figure (c) uses a vocabulary size that is larger than the network width, as commonly done for language models, and the advantage of the deeper networks disappears. Note that the loss value depends on the vocabulary size, and thus loss values between figures (a)-(c) are not directly comparable.

served variability in the architecture configurations across domains. By identifying expressivity bottlenecks that have manifested themselves mainly in non-linguistic data modalities, we provide simple *domain independent* practical guidelines for Transformer architecture selection.

Specifically, we identify a *vocabulary bottleneck* in self-attention, proving that the rank of the input embedding matrix caps the network width's contribution to expressivity. Furthermore, we show that when the width exceeds the input embedding rank, deepening is exponentially favorable over widening. We empirically demonstrate the predicted effects of the vocabulary bottleneck. Figure 1 shows experiments in the language domain, demonstrating that when decreasing the vocabulary size below the value of the network width, depth becomes more important than width also in this data modality. This provides empirical support for our expressivity-based interpretation of the variation in the depth-to-width ratio, and counters the modality based interpretation.

As the use of Transformer architectures was extended to different modalities, the effect of this architectural element was overlooked. For example, in the bioinformatics domain, Rives et al. (2019) lead the RaptorX benchmark for protein long-range contact precision (Wang et al., 2017) with a Transformer model that has width 1280, but a vocabulary size 33, equivalent to a character-level model. Our theoretical results, backed by targeted experiments, indicate that this is very sub-optimal, as the width is severely capped by the low rank of the embedding matrix. In vision, the above-mentioned depth-to-width ratio inconsistency across models is commensurate with our findings – the depth-128 Sparse Transformer model has a pixel-intensity vocabulary size of 256, while the depth-12 ViT-Base model enjoys full-rank embedding and can utilize its width of 768.

The above vocabulary bottleneck rarely comes into play when considering language-related applications – the input embedding matrix is commonly a fully-ranked matrix with dimensions of the words vocabulary size ($\sim$ 30K) times the network width ($\sim$ 1K). However, the popular ALBERT model of Lan et al. (2020) has deliberately reduced the rank of the embedding matrix for space efficiency reasons. Our results formalize the negative impact of this approach on model expressivity, and our experiments quantify the performance degradation – the ALBERT approach leads to 25% redundancy in network size, *i.e.*, a low rank network is surpassed by a full rank network with 75% of the parameters.

Another consequence of our framework that applies to leading language models, has to do with the method used in Raffel et al. (2020) for scaling up to their 11B parameter language model, referred to as T5-11B (which holds the state-of-the-art in many NLP benchmarks). Due to hardware related considerations, they elected to keep the representation dimension between layers relatively small, and to invest most of the parameters in the self-attention operation itself. Beyond the vocabulary bottleneck, our rank bottlenecking proof technique applies to bottlenecks created mid-architecture, and specifically we show that a low representation dimension caps the ability to enjoy an excessive parameter increase in the self-attention operation. We validate this prediction empirically, and project T5-11B to be $\sim 45\%$ redundant, *i.e.*, it could achieve its performance with roughly half its size if trained with a regular architecture. Notably, a modified version of T5-11B, dubbed T51.1 XXL (Shazeer, 2020), fixed the above bottleneck in a manner which completely accords with our recommendation. We expand on this modification in section 5.3.

The remainder of this paper is organized as follows. In section 2 we present the analyzed input embedding methods and Transformer architecture. In section 3 we present a measure referred to as a function's separation rank, with which, in section 4, we establish our main results regarding functions realized by Transformer architectures. In section 5, we empirically demonstrate the predicted bottleneck phenomena, summarized by:

1. A degradation when the input embedding rank is smaller than the network width (section 5.1);
2. An advantage of depth over width when the input embedding rank is smaller than the network width (section 5.2);
3. A degradation when the network width is smaller than the internal attention representation (section 5.3).

2. The analyzed Transformer architecture

While the original encoder-decoder based Transformer architecture of Vaswani et al. (2017) is still widely used, variants based solely on its encoder (*e.g.*, BERT (Devlin et al., 2019)) or decoder (*e.g.*, GPT (Radford et al., 2018)) have gained popularity in various domains. For the sake of simplicity, we will analyze such variants. The analyzed Transformer architecture is comprised of an input embedding layer, which we present in section 2.1, followed by L Transformer layers, which we present subsequently in section 2.2.

2.1. The input embedding layer

We will analyze two common methods for translating raw data into an embedding. The first method, *vocabulary* based, is employed when the input is a sequence of N variables that can have one of V discrete values, referred to as vocabulary tokens, *i.e.*, $\left\{w^i\right\}_{i=1}^N$ where $\forall i:\ w^i \in [V]$. Naturally, this method is prevalent in language applications of Transformers (hence the name), but it appears also in other domains such as bioinformatics (Rives et al., 2019) or computer vision (Chen et al., 2020). The second embedding method that we analyze is *convolution* based, used for example over images (Vision Transformer (Dosovitskiy et al., 2021)). This method is applied when the raw input sequence is of real valued vectors $\left\{\mathbf{x}^i\right\}_{i=1}^M$, where $\forall i : \mathbf{x}^i \in \mathbb{R}^{d_{\text{input}}}$, and downsampling is required in order to reduce the sequence length related computation costs.

When using the vocabulary based embedding, since the size of the vocabulary affects storage and runtime, it is common for V to reflect a precomputed compression of the "raw vocabulary" of all possible symbols in the raw data. For example, in language, while character level vocabularies could be used with manageable costs (though as we prove below, these incur severe underutilization of the Transformer's expressive power), when aiming for word-level vocabularies, the number of unique words in web based sources can surpass 1M, and therefore sub-word based encodings such as WordPiece (Schuster & Nakajima, 2012), SentencePiece (Kudo & Richardson, 2018) or BPE (Sennrich et al., 2016) are used to reduce the number of vocabulary tokens to $V \sim 30$K. In computer vision, for ImageGPT, Chen et al. (2020) pre-compute 512 clusters over 3-dimentional RGB pixel values, along with a mapping of each pixel to one of corresponding $V = 512$ vocabulary tokens.

Translating the input sequence $\left\{w^i \in [V]\right\}_{i=1}^N$ into indicators: $\left\{\hat{\mathbf{w}}^i = \hat{e}_{w^i}\right\}_{i=1}^N$, where $\forall i\ :\ \hat{\mathbf{w}}^i\ \in\ \mathbf{V}\ :=\ \mathbb{R}^V$, the output of the embedding layer at position $i \in [N]$ is:

$$\mathbf{y}^{0,i} = M_{\text{V}}\ \hat{\mathbf{w}}^i + \mathbf{p}^i, \tag{1}$$

where $M_{\text{V}} \in \mathbb{R}^{d_x \times V}$ is a learned matrix referred to as the vocabulary matrix (or the embedding matrix). Accordingly, $\mathbf{y}^{0,i}$ is a vector in $\mathbb{R}^{d_x}$, *i.e.*, per location, the input to the first Transformer layer is of dimension d_x, the network width. The added learned position dependent term $\mathbf{p}^i \in \mathbb{R}^{d_x}$ is referred to as the positional embedding.

In the second, convolution based, input embedding method, in order to fit M real valued input vectors into a Transformer layer with an input sequence of size N (such that M is a multiple integer of N), a convolutional kernel $W^{\text{conv}} \in \mathbb{R}^{\frac{M}{N} \times d_x \times d_{\text{input}}}$ is used for computing the ith output of the embedding layer:

$$\mathbf{y}^{0,i} = \sum_{j=1}^{\frac{M}{N}} W_j^{\text{conv}} \mathbf{x}^{\frac{M}{N}\cdot(i-1)+j} + \mathbf{p}^i, \tag{2}$$

where $\mathbf{p}^i$ is the added positional embedding.

In section 4, we will show an expressivity bottlenecking result that depends on r, a measure of rank that corresponds to the employed embedding method. For the vocabulary embedding method, $r \leq \min\{d_x, V\}$ is the rank of the vocabulary matrix:

$$r = \text{rank}\left(M_{\text{V}}\right). \tag{3}$$

For the convolution method, by defining the effective vocabulary dimension to be $V := {}^{M}\!/_{N} \cdot d_{\text{input}}$, we reshape the convolutional kernel W^{conv} into a matrix $\tilde{W}^{\text{conv}} \in \mathbb{R}^{d_x \times V}$ and define $r \leq \min\{d_x, V\}$ as:

$$r = \text{rank}\left(\tilde{W}^{\text{conv}}\right). \tag{4}$$

Importantly, though the data modality influences embedding considerations, there is relative freedom in choosing the embedding method (*e.g.*, both the above methods were employed over images) and controlling the corresponding embedding rank r (via choosing the vocabulary size/rank or the convolution kernel size). Therefore, we will argue that by noticing the rank related expressivity bottleneck, suited input embeddings that allow for full utilization of the Transformer expressivity can be used in different domains.

2.2. The self-attention architecture

Following (Levine et al., 2020), our theoretical analysis will focus on a variant of self-attention in which the attention scores are unnormalized, and which compounds self-attention layers without mixing in element-wise feed-forward layers in between. Accordingly, given an embedding output sequence $\{\mathbf{y}^{0,i}\}_{i=1}^{N}$ (see previous subsection), the function realized by the analyzed H-headed depth-L width-d_x Transformer architecture is recursively written:

$$\mathbf{y}^{l+1,i}\left(\mathbf{y}^{l,1},...,\mathbf{y}^{l,N}\right) \coloneqq \sum_{h=1}^{H} W^{\mathrm{O},l,h} \sum_{j=1}^{N} a_{hj}^{i} W^{\mathrm{V},l,h}\mathbf{y}^{l,j}$$

$$a_{hj}^{i} \coloneqq \left\langle W^{\mathrm{Q},l,h}\mathbf{y}^{l,i}, W^{\mathrm{K},l,h}\mathbf{y}^{l,j}\right\rangle \tag{5}$$

where $\forall l \in [L], h \in [H]$, $W^{\mathrm{K},l,h}, W^{\mathrm{Q},l,h}, W^{\mathrm{V},l,h}$, $\left(W^{\mathrm{O},l,h}\right)^{\top} \in \mathbb{R}^{d_a \times d_x}$ are the Key, Query, Value and Output self-attention learned weights matrices introduced in Vaswani et al. (2017). The attention dimension is usually chosen as $d_a = {}^{d_x}/_{H}$, but in section 4.2.3 we analyze a bottleneck related to relaxing this constraint, and in section 5.3 we project that due to this bottleneck the leading T5 architecture of Raffel et al. (2020) could be almost half of its current size and reach the same performance.

The above relaxations are justified by noting that:

1. Press et al. (2020) train a "self-attention first" network that first performs all of the self-attention operations consecutively, and only then performs all of the position-wise feed-forward operations. This network achieves comparable language modeling performance relatively to the regular approach of interleaving these functionalities. Since the feed-forward operation does not mix different locations, this outcome directly implies that the self-attention mechanism itself provides all of the elaborate input integration, and that the interleaved feed-forward layer is just a performance booster.

2. While removing the attention score normalization via softmax is not in line with an intuitive interpretation of attention as distributing "fractions" of an overall attention budget among inputs, a growing body of work shows that the attention weights distribution does not directly correlate with predictions (Jain & Wallace, 2019; Pruthi et al., 2020; Brunner et al., 2020). Moreover, (Richter & Wattenhofer, 2020) recently point out undesirable traits of the softmax operation, demonstrating that its property of confining the outcome to the convex hull of its inputs unnecessarily limits the expressibility of the self-attention mechanism. The analyzed unnormalized variant retains the actual operation of dynamically linking input and output locations via the Key/Query/Value connectivity of self-attention.

The goal of the above points is not to advocate modifications in the Transformer's non-linearity or normalization operations, but to note that while these are under examination and are susceptible to alteration, the connectivity of self-attention, manifested by eq. (5), is the core mechanism driving its functionality. A reinforcing signal to the above argument is the relevance of conclusions drawn by directly analyzing the self-attention mechanism to experiments in commonly employed self-attention networks, as presented in Levine et al. (2020) regarding depth efficiency regimes as well by us later in section 5 regarding the effects of the embedding rank. These experiments are consistently compatible with theoretical predictions that arise from our framework.

3. A capacity for modeling input dependencies

In this section, we introduce the separation rank of the function realized by a Transformer as a measure that quantifies its ability to model dependencies between subsets of its inputs. We will use this measure in section 4 in order to establish the embedding bottleneck in Transformer architectures. The separation rank, introduced in Beylkin & Mohlenkamp (2002) for high-dimensional numerical analysis, was employed for various applications, *e.g.*, chemistry (Harrison et al., 2003), particle engineering (Hackbusch, 2006), and machine learning (Beylkin et al., 2009). More recently, the separation rank has been established as a measure of dependencies modeled by deep convolutional and recurrent networks w.r.t. their inputs (Cohen & Shashua, 2017; Cohen et al., 2017; Levine et al., 2018a), and tied to quantum entanglement measures for proving that these deep learning architectures can model elaborate many-body quantum particle correlations (Levine et al., 2018b; 2019; Sharir et al., 2020). Our usage of the separation rank directly follows that in Levine et al. (2020), who employed this measure for studying the depth-to-width interplay in self-attention networks.

Let (A, B) be a balanced partition of the input locations, *i.e.*, A and B are equal sized disjoint subsets of $[N]$ whose union gives $[N]$. The separation rank of a function $y(\mathbf{x}^1, \ldots, \mathbf{x}^N)$ w.r.t. a partition (A, B), is the minimal number of summands that together sum up to equal y, where each summand is *multiplicatively separable w.r.t.* (A, B), *i.e.*, is equal to a product of two functions – one that intakes only inputs from one subset $\{\mathbf{x}^i : i \in A\}$, and another that intakes only inputs from the other subset $\{\mathbf{x}^i : i \in B\}$. Formally, the *separation rank* of $y : \mathbf{V}^N \to \mathbb{R}$ w.r.t. the partition (A, B) is defined as follows:

$$\begin{aligned} &sep\,(y, A, B) \coloneqq \min\{R \in \mathbb{N} \cup \{0\} : \\ &\quad \exists g_1, \ldots, g_R : \mathbf{V}^{N/2} \to \mathbb{R}, g'_1, \ldots, g'_R : \mathbf{V}^{N/2} \to \mathbb{R} \\ &\quad y\left(\mathbf{x}^1, \ldots, \mathbf{x}^N\right) = \\ &\quad \sum\nolimits_{r=1}^{R} g_r\left(\{\mathbf{x}^i : i \in A\}\right) g'_r\left(\{\mathbf{x}^i : i \in B\}\right)\} \end{aligned}$$

We will use the separation rank as a quantifier of correlations that can be expressed by the model. If the separation rank of a function w.r.t. an input partition is 1, the function is separable, meaning it cannot take into account consistency between $\{\mathbf{x}^i\}_{i\in A}$ and $\{\mathbf{x}^i\}_{i\in B}$. In a statistical setting, if y is a probability density function such as in Radford et al. (2018), this would mean that $\{\mathbf{x}^i\}_{i\in A}$ and $\{\mathbf{x}^i\}_{i\in B}$ are statistically independent. The higher $sep(y; A, B)$ is, the farther y is from this situation, *i.e.* the more it models dependency between $\{\mathbf{x}^i\}_{i\in A}$ and $\{\mathbf{x}^i\}_{i\in B}$, or equivalently, the stronger the correlation it induces between the inputs indexed by A and those indexed by B.

4. The Vocabulary Bottleneck

In this section, we theoretically establish the vocabulary bottleneck phenomenon in Transformer architectures, and its effect on the depth-to-width interplay. Specifically, we prove an upper bound on the separation rank of the analyzed Transformer architecture that grows exponentially with depth L times *the minimum* between the network width d_x and the embedding rank r, and show it is tight for $L > \log_3 d_x$.

Without considering the embedding rank bottleneck, Levine et al. (2020) have shown that for $L > \log_3 d_x$, both depth and width contribute exponentially to the separation rank, and have provided extensive empirical corroboration of this prediction for a "depth-**in**efficiency" regime of self-attention. For the complementary regime of $L < \log_3 d_x$ they prove a "depth-efficiency" result, by which deepening is favorable over widening. Our results imply that when the embedding rank is lower than the network width, the width related parameters are underutilized and "depth-efficiency" kicks in immediately, even within the more practical $L > \log_3 d_x$ regime. We formalize these notions below, and validate them empirically in the next section.

The following theorems states that the network's capacity to model dependencies is harmed by a low rank embedding:

Theorem 1. *(upper bound on the separation rank) Let $y_p^{i,L,d_x,H,r}$ be the scalar function computing the p^{th} entry of an output vector at position $i \in [N]$ of the H-headed depth-L width-d_x Transformer network*[1] *defined in eq.* (5), *where the embedding rank r is defined by eq.* (3) *(vocabulary embedding) or eq.* (4) *(convolution embedding). Let $sep\left(y_p^{i,L,d_x,H,r}\right)$ denote its separation rank (section 3). Then the following holds:*

$$\log\left(sep(y_p^{i,L,d_x,H,r})\right) = \tilde{O}\left(L \cdot \min\{r, d_x\}\right) \qquad (6)$$

The theorem below simply states that under additional assumptions, the upper bound in the above theorem is asymptotically tight:

Theorem 2. *(lower bound on the separation rank) For $y_p^{i,L,d_x,H,r}$ as defined in in theorem 1, assume that $L > \log_3 d_x$, $H < r$. Furthermore, for the vocabulary embedding case, assume that $N \to \infty$. Then for all values of the network weights but a set of Lebesgue measure zero, the following holds:*

$$\log\left(sep(y_p^{i,L,d_x,H,r})\right) = \tilde{\Omega}\left(L \cdot (\min\{r, d_x\} - H)\right) \quad (7)$$

Note that the architectural assumptions that are added for the lower bound are practically reasonable. (1) $L > \log_3 d_x$: for typical width d_x of 1000s, the log implies that the bound holds for networks of practical depths ~ 8 and above; (2) $H < r$: the number of attention heads per layer H is typically in the order of 10s, while the width and rank are typically 100s and above.

Note that while the $N \to \infty$ assumption in the vocabulary embedding are not practically reasonable, Our proof usage of N is clearly wastefully, and we conjecture the lower bound holds for $N = \Omega\left({}^{r\cdot L}\!/\!_{\log_3 r}\right)$ (see the appendix for detailed discussion). Moreover, (Bhojanapalli et al., 2020) showed that $d_x < N$ limits self-attention expressivity, thus it is unlikely that huge N contribute significantly to the network's capacity to model dependencies.

In the following, we outline the proof sketch for theorem 1 (section 4.1) and then discuss three practical implications of the established vocabulary rank bottleneck (section 4.2).

4.1. Proof sketch for theorem 1

We present below the proof outline for the upper bound in eq. 6, which establishes that the contribution of width to the separation rank is bottlenecked by the input embedding rank. We defer the full proof to the appendix, along with the proof of the lower bound in theorem. 2, which establishes the tightness of the upper bound.

First, notice that each self-attention layer, as defined in eq. (5), is a degree 3 polynomial over its $N \cdot d_x$ scalar inputs. Since both the vocabulary embedding and the convolution embedding are linear mappings (eqs. (1) and (2)), in both cases the whole network is a composition of L degree-3 polynomials. Therefore $y_p^{i,L,d_x,H,r}$ is a degree 3^L polynomial over $N \cdot d_x$ variables. By definition, the

[1] For simplicity, in this theorem we ignored the positional embedding dependencies and defer the full theorem to the appendix.

separation rank of any monomial is 1. In addition, the separation rank of sum of functions is upper bounded by the sum of theirs separation ranks. Thus, we upper bounded $sep(y_p^{i,L,d_x,H,r})$ by the number of its monomials, which is at most $O\left(\left(3^L+N\cdot d_x\right)^{N\cdot d_x}\right)$ (a simple combinatorial bound detailed in the appendix).

The above analysis is agnostic to the first linear embedding layer. However, when $r<d_x$ this layer is important since the $N\cdot d_x$ variables have only $N\cdot r$ degrees of freedom: we can define a set of $N\cdot r$ variables which are a linear combinations of the original variables. Importantly $y_p^{i,L,d_x,H,r}$ is still a degree 3^L polynomial over the new variables, and each monomial of this polynomial has separation rank of 1. By noticing that the summation over monomials is not tight, a more careful analysis, as done in the appendix, shows that for separation rank purposes, the effective degree of freedom is only $O\left(r\right)$, independent of N, thus concluding that $sep(y_p^{i,L,d_x,H,r})$ is upper bounded by $O\left(\left(3^L+r\right)^r\right)$.

4.2. Practical implications

Beyond quantifying the vocabulary bottleneck's effect on the network's ability to model dependencies (via separation rank), theorems 1 and 2 have direct implications on Transformer architecture design. We detail them in the following.

4.2.1. The low rank expressivity bottleneck

Since two functions can be equal only if they have the same separation rank, a network with a low rank embedding $r<d_x$ cannot express the operation of a full rank $r=d_x$ network. As we demonstrate in section 5.1 (figure 2), this result translates into an empirical degradation in performance when $r<d_x$.

A popular low vocabulary rank method is ALBERT, suggested in (Lan et al., 2020); we show in section 5.1 that the low ${}^{r}/_{d_x}={}^{128}/_{4096}$ ratio implemented in their network yields a 25% redundancy in network size, *i.e.*, a low rank network is surpassed by a full rank network with 75% of the parameters. The above vocabulary bottleneck is even more common in non-linguistic domains. For example, Rives et al. (2019) train a Transformer model with ${}^{r}/_{d_x}={}^{33}/_{1280}$. The result in theorems 1 and 2 formalizes the sub-optimality of these settings.

4.2.2. Effect on the depth-to-width interplay

Beyond establishing a degradation in performance for low embedding rank Transformers, theorems 1 and 2 imply an advantage of deepening versus widening beyond the point of $d_x=r$, as deepening contributes exponentially more to the separation rank in this case. As we demonstrate in section 5.2 (figures 1 and 3), when comparing two Transformer architectures of depths $L^{\text{shallow}}<L^{\text{deep}}$ with the same embedding rank r and the same number of parameters, the two networks perform comparably when $d_x^{\text{shallow}}\leq r$, and the deeper network is better when $d_x^{\text{shallow}}>r$.

This implication directly explains the observed depth-to-width ratio differences between language models and vision models. The Sparse Transformer (Child et al., 2019) over images, which is 128 layers deep at the same parameter count of the 12-layered BERT-Base in language, has a small pixel-intensity vocabulary (each pixel is translated into 3 input tokens corresponding to its color channels), which caps the contribution of width. The same vocabulary is used in the ablation of Henighan et al. (2020), which attributes the difference in optimal depth-to-width ratio to the difference in data modalities. The result in theorems 1 and 2, along with the corroborating experiments in section 5.2, implies that this phenomenon is *modality independent*, and is in fact related to architecture expressivity.

4.2.3. A mid-architecture bottleneck – width caps the internal attention dimension

The above implications relate to the vocabulary bottleneck caused by a low input embedding rank. By observing that the upper bound on the separation rank does not depend on the number of attention heads H, we establish a mid-architecture bottleneck related to this architectural aspect, which affects leading Transformer architectures.

Specifically, following the original implementation of Vaswani et al. (2017), most common Transformer architectures set the number of heads to be $H={}^{d_x}/_{d_a}$, *i.e.*, the network width divided by the internal attention dimension (see text below eq. (5)). However, in their effort to drastically increase network size under hardware restrictions, Raffel et al. (2020) trained their unprecedentedly-sized 11B parameter T5 model by decoupling the above: They train a width $d_x=1$K network but increase the number of attention heads per layer to $H=128$ while keeping the attention dimension fixed at $d_a=128$. Thus, T5-11B achieves an internal attention embedding of $H\cdot d_a=16$K before projecting back down to the width value of 1K between layers.

However, since the bounds in theorems 1 and 2 are capped at d_x, our theoretical framework predicts that this form of parameter increase is suboptimal, *i.e.*, adding parameters to the internal attention representation beyond $H\cdot d_a=d_x$ is inferior to simply increasing the width d_x. We verify this conclusion in section 5.3 (figure 4), where we establish that the architectural configuration presented in T5 is indeed highly parameter redundant.

5. Experiments

In the previous sections, we analyzed a simplified version of Transformer networks (described in section 2). For this class, we proved the existence of a low-rank embedding bottleneck that limits the contribution of network width to

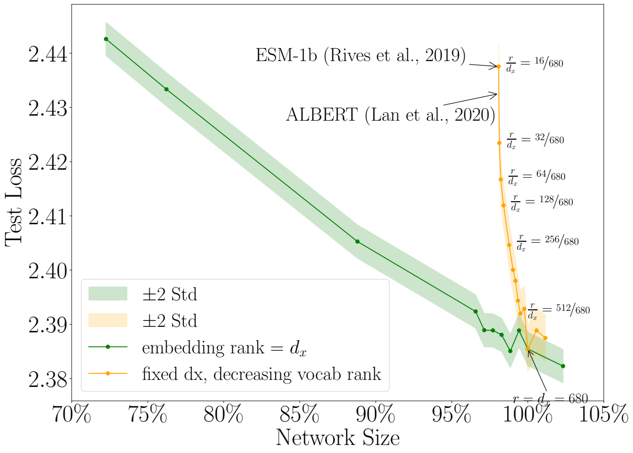

Figure 2. An experimental validation of the low-rank embedding bottleneck. We reduce the network size by decreasing either the width of all layers (green), or by only decreasing the input embedding rank (orange). A rapid degradation is observed when the input embedding rank is decreased. This degradation affects leading models in NLP (ALBERT) and in other domains (ESM-1b).

the Transformer expressivity. In this section, we demonstrate that **our theoretical predictions are manifested in common Transformer networks**; the experiments below were conducted over common architectures which include all operations that were omitted in our theoretical analysis.

Since unidirectional models are more stable to vocabulary variations than bidirectional models (Levine et al., 2021), we trained decoder-only language models, by optimizing the autoregressive log-likelihood of the training examples for $1M$ steps. We perform our experiments over language in order to establish that architecture expressivity causes the variation in the optimal depth-to-width ratio: we demonstrate that the vocabulary bottleneck causes the examined language modality to exhibit the same trends that were attributed to other modalities by Henighan et al. (2020).

Our training set was English Wikipedia, BookCorpus and OpenWebText, with a total size of 60G. We report the loss on a held out test set of 40K sequences. Notably, we estimated the variance of the pretraining and evaluation procedure by rerunning 7 of the trained architectures five times each, and found it to be very low – the reported test loss is stable up to 10^{-3}. The remainder of the training details are given in the appendix.

5.1. Rank bottleneck degrades performance

Theorems 1 and 2 reveal a low-rank embedding-bottleneck phenomenon in Transformer architectures. In this subsection, we demonstrate this phenomenon by factoring the input embedding matrix into two matrices of dimensions $d_x \times r$, $r \times V$, while incrementally decreasing the rank r. This is similar to the approach suggested in ALBERT (Lan et al., 2020) for reducing the parameter count. Specifically, we trained depth-12 networks with embedding ranks that range between 16 and the full rank of $d_x = 680$. Additionally, we trained a baseline of full input embedding rank depth-12 networks of sizes varying between 50M and 75M parameters (full details on the widths are given in the appendix).

In common language model implementations, the input embedding weights are shared with the final classification weights (referred to as weight tying (Press & Wolf, 2017)). In our experiment, we wanted ensure that the performance degradation is caused by the embedding bottleneck and not by the softmax bottleneck that Yang et al. (2018) establish regarding the final classification operation. Therefore, we did not perform the above weight tying, and kept the classification weights matrix a fully ranked $V \times d_x$ matrix, even when we decreased the rank of the input embedding matrix.

Figure 2 shows that when decreasing the input embedding rank, the loss increases much more rapidly than when reducing the network size by decreasing the width d_x. A network with ~ 70 million parameters and an "embedding-rank to width" ratio of ${}^{r}/_{d_x} = {}^{16}/_{680}$, is comparable in performance to a non-bottlenecked network that is 25% smaller. This low-rank ratio is close the the ratio of ALBERT-xxlarge ${}^{r}/_{d_x} = {}^{128}/_{4096}$ as well as to the ratio caused by the $V = 33$ and $d_x = 1280$ of the leading 650M parameter ESM-1b (Rives et al., 2019) protein model, implying that their network could have been strengthened by constructing a larger vocabulary (perhaps a "word-level" equivalent).

5.2. Vocabulary affects the depth-to-width interplay

The results of the previous section directly imply that by limiting the embedding rank, a small vocabulary can harm performance. In this section we verify the second conclusion of theorems 1 and 2, which states that for either $V < d_x$ or $r < d_x$, it is better to use narrower and deeper models.

5.2.1. Small vocabulary size $\mathbf{V} < \mathbf{d_x}$

We compared networks of depths $L^{\text{shallow}} = 24, L^{\text{deep}} = 48$, of sizes ranging between 5M and 110M, when three different BPE (Sennrich et al., 2016) vocabularies of sizes $V = 257, 500, 2000$ are used (full details on the vocabularies and trained architectures are given in the appendix). Figures 1(a)-(c) show a clear trend: the smaller the vocabulary, the sooner the deeper networks begin outperforming the shallow ones. Moreover, at the "depth-efficiency" point, for which the deeper network starts outperforming the shallow one, the width of the shallower network is around the vocabulary size.

In addition, we show in the appendix that this does not occur when the vocabulary size exceeds the network width and

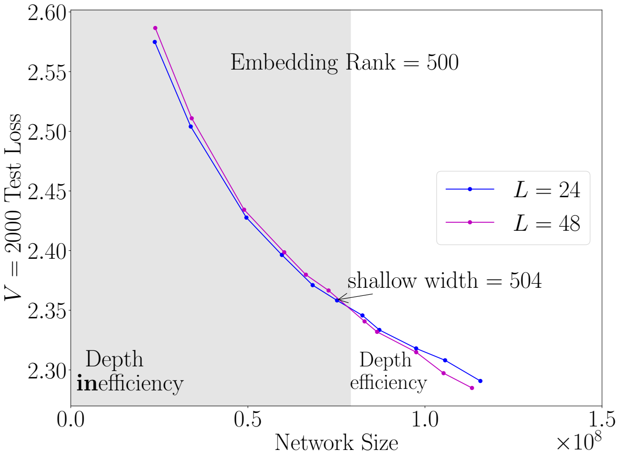

Figure 3. An experiment with the same $V = 2000$ vocabulary of figure 1(c), when the rank of the input embedding matrix is restricted to 500. Deeper networks gain an advantage over shallower ones due to the rank bottleneck.

does not constitute a bottleneck. In this case, the vocabulary size has negligible effect on the "depth-efficiency" point. For example, we show that the "depth-efficiency" point of GPT-2's vocabulary (Radford et al., 2019) remains very close to that of our $V = 2000$ vocabulary, even though the GPT-2 vocabulary is $\sim 25X$ larger. This is directly in line with our prediction when there is no vocabulary bottleneck.

Overall, figure 1 clearly shows that the same phenomenon that occurred over images and mathematical data, of tilting the depth-to-width ratio towards depth, occurs also in the case of language when the vocabulary bottleneck is present. This phenomenon was attributed by Henighan et al. (2020) to the difference in data modalities, but the above outcomes reinforce the network expressivity related interpretation for its origin.

5.2.2. Low input rank $\mathbf{r < d_x}$

Notably, since the input sequence length N is fixed, using a smaller vocabulary implies that the model sees less text. Therefore, the above differences in the depth-to-width tradeoffs could be attributed to the variation in training data rather than to the effect of the rank bottleneck. Figure 3 establishes the rank bottleneck as the influencing factor by comparing $L^{\text{shallow}} = 24, L^{\text{deep}} = 48$ networks with the same vocabulary size of 2000, while limiting the embedding rank to 500. Similarly to the vocabulary varying experiments in figure 1, the deeper network surpassed the shallower one when the latter's width reached its rank.

Interestingly, in this case the transition is very close to the rank of 500, at $d_x^{\text{shallow}} = 504$, while in the $V = 500$ setting of figure 1b the transition occurs earlier, around $d_x^{\text{shallow}} = 408$. We conjecture that when the vocabulary itself is small, not all tokens are equally utilized, whereas for a low rank vocabulary matrix with a larger vocabulary, a better utilization of the rank is achieved. Either way, we see that the advantage of depth is indeed an implication of the embedding rank bottleneck.

5.3. Width bottlenecks the attention dimension

In this section, we demonstrate the effect of the mid-architecture bottleneck identified in section 4.2.3, and show that performance is degraded when the internal attention representation dimension exceeds network width, *i.e.*, when $H \cdot d_a > d_x$.

We trained networks of sizes ranging between 30M and 70M, while varying the bottleneck ratio of $^{H \cdot d_a}/_{d_x}$ within $\{1, 2, 4, 8, 16\}$, where 1 is the baseline value. For each bottleneck ratio and network size, we chose the best depth in $\{12, 18, 24\}$. We provide the results of all the depths per bottleneck ratio in the appendix, showing that the performance difference between values of the bottleneck ratio is much larger than the variation between different depths per bottleneck ratio.

Figure 4 shows that by fixing d_a and increasing H, performance is indeed degraded for $d_x < H \cdot d_a$. Importantly, the degradation is monotone with the $^{H \cdot d_a}/_{d_x}$ bottleneck ratio. The second largest T5 model of Raffel et al. (2020) has distributed its 3B parameters by using width of $d_x = 1\text{K}$ and a bottleneck ratio of $^{H \cdot d_a}/_{d_x} = 4$, corresponding to the yellow line in figure 4. The figure shows that the performance of a network with this ratio can be achieved by a baseline network with $\sim 75\%$ of the parameters (green). The largest T5 model has a width of $d_x = 1\text{K}$ and a bottleneck ratio $^{H \cdot d_a}/_{d_x} = 16$, corresponding to the red line in figure 4. The performance of a network with this ratio can be achieved by a baseline network with $\sim 55\%$ of the parameters, implying that T5-11B could have been trained with $\sim 6\text{B}$ parameters with no degradation.

It is noteworthy that T5.1.1 XL and T5.1.1 XXL (Shazeer, 2020; Xue et al., 2021b;a), more recent, modified, versions of T5-3B and T5-11B, have increased widths of 2K and 4K, respectively, with corresponding number of heads such that $^{H \cdot d_a}/_{d_x} = 1$. Our theoretical analysis and its empirical corroboration in figure 4 highlight the importance of this architectural aspect.

6. Discussion

After observing a variation in the optimal depth-to-width ratios of Transformers when applied to different domains, previous works have concluded that this architectural design aspect depends on the input data modality. Our theoretical framework, reinforced by supporting experiments, indicates the variation in common vocabulary sizes across domains as

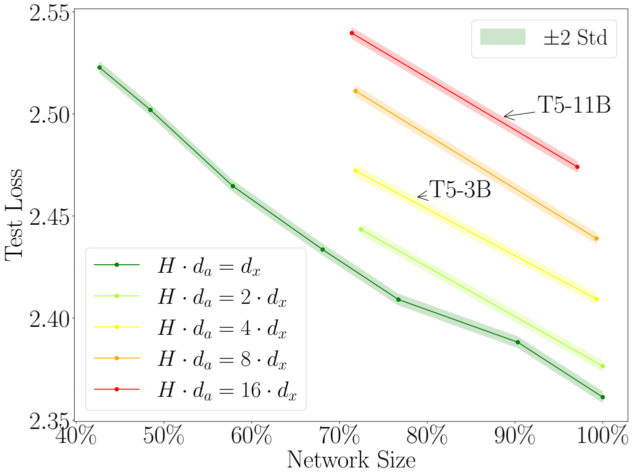

Figure 4. The $d_x < H \cdot d_a$ bottleneck of the T5 architecture degrades performance. As the bottleneck ratio increases, smaller baseline architectures outperform variants that invest parameters in the internal attention representation. T5-11B was trained with $H \cdot d_a = 16d_x$, implying a $\sim 45\%$ parameter redundancy.

the root of the observed depth-to-width variability. Specifically, we prove and validate empirically that when the vocabulary size or rank is smaller than the network width, the contribution of depth to expressivity is boosted and the optimal depth-to-width ratio per Transformer architecture size increases.

Our results provide practical domain independent guidelines for Transformer architecture design. If possible, it is good to increase the input embedding rank such that it surpasses the network width. For example, the largest Vision Transformer, ViT-Huge (Dosovitskiy et al., 2021), has width of 1280 but an input rank of 768, which constitute a bottleneck. This can easily be alleviated if noticed, for example via an additional convolutional layer over the inputs (see eq. (4)). Alternatively, the network depth can be increased at the expense of its width, as previous works propose (without noting the input rank origins).

However, it is important to note the vocabulary independent depth-to-width Transformer expressivity considerations given in (Levine et al., 2020): for a given parameter budget, a network can be too deep. More generally, from either expressivity, optimization, or even engineering considerations, a large width is beneficial (*e.g.*, allows for a more effective parallelization). Therefore, methods for increasing the vocabulary rank, such as an elaborate convolutional embedding, or alternatively, different coding methods, should be considered for training a large width Transformer that does not suffer from the identified vocabulary bottleneck.

Regarding impact on NLP (still the major consumer of Transformers), while the vocabulary size is commonly larger than network width in this field, and does not therefore constitute a bottleneck, Levine et al. (2020) predict that scaling up to 1-Trillion parameter networks and beyond, would require massive widening of $d_x = 30$K and more. This reaches standard vocabulary sizes, so the vocabulary bottleneck should be noted. Moreover, Xue et al. (2021a) concurrently demonstrate that a byte-level vocabulary can work well in Transformer-based LMs. As we show, when proposing architectural modifications or vocabulary size reductions, our theoretically established bottlenecks should be considered, as they translate into practically manifested redundancies (see figures 2 and 4). Overall, our work aims to provide timely theoretical interpretations, to help guide the rapid empirical advances of our field.

Acknowledgments

This research was supported by the ERC (European Research Council) and the ISF (Israel Science Foundation). Experiments were performed with Cloud TPUs and supported by Google's TPU Research Cloud (TRC). Experiments results management done using Neptune and supported by Neptune.ai's research program. Yoav Levine was supported by the Israel Academy of Sciences Adams fellowship.

References

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12435–12446. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf.

Beylkin, G. and Mohlenkamp, M. J. Numerical operator calculus in higher dimensions. *Proceedings of the National Academy of Sciences*, 99(16):10246–10251, 2002.

Beylkin, G., Garcke, J., and Mohlenkamp, M. J. Multivariate regression and machine learning with sums of separable functions. *SIAM Journal on Scientific Computing*, 31(3):1840–1857, 2009.

Bhojanapalli, S., Yun, C., Rawat, A. S., Reddi, S., and Kumar, S. Low-rank bottleneck in multi-head attention models. In *International Conference on Machine Learning*, pp. 864–873. PMLR, 2020.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1876–1900. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Brunner, G., Liu, Y., Pascual, D., Richter, O., Ciaramita, M., and Wattenhofer, R. On identifiability in transformers. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJg1f6EFDB.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *arXiv preprint arXiv:2106.01345*, 2021.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. Generative pretraining from pixels. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1691–1703. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/chen20s.html.

Child, R., Gray, S., Radford, A., and Sutskever, I. Generating long sequences with sparse transformers. *URL https://openai.com/blog/sparse-transformers*, 2019.

Cohen, N. and Shashua, A. Inductive bias of deep convolutional networks through pooling geometry. In *5th International Conference on Learning Representations (ICLR)*, 2017.

Cohen, N., Sharir, O., Levine, Y., Tamari, R., Yakira, D., and Shashua, A. Analysis and design of convolutional networks via hierarchical tensor decompositions. *arXiv preprint arXiv:1705.02302*, 2017.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., and Sutskever, I. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Hackbusch, W. On the efficient evaluation of coalescence integrals in population balance models. *Computing*, 78 (2):145–159, 2006.

Harrison, R. J., Fann, G. I., Yanai, T., and Beylkin, G. Multiresolution quantum chemistry in multiwavelet bases. In *Computational Science-ICCS 2003*, pp. 103–110. Springer, 2003.

Henighan, T., Kaplan, J., Katz, M., Chen, M., Hesse, C., Jackson, J., Jun, H., Brown, T. B., Dhariwal, P., Gray, S., et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

Jain, S. and Wallace, B. C. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1357. URL https://www.aclweb.org/anthology/N19-1357.

Jun, H., Child, R., Chen, M., Schulman, J., Ramesh, A., Radford, A., and Sutskever, I. Distribution augmentation for generative modeling. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5006–5019. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/jun20a.html.

Kudo, T. and Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://www.aclweb.org/anthology/D18-2012.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1eA7AEtvS.

Levine, Y., Sharir, O., Ziv, A., and Shashua, A. Benefits of depth for long-term memory of recurrent networks. *(ICLR 2018) International Conference on Learning Representations workshop*, 2018a.

Levine, Y., Yakira, D., Cohen, N., and Shashua, A. Deep learning and quantum entanglement: Fundamental connections with implications to network design. In *International Conference on Learning Representations*, 2018b. URL https://openreview.net/forum?id=SywXXwJAb.

Levine, Y., Sharir, O., Cohen, N., and Shashua, A. Quantum entanglement in deep learning architectures. *Physical review letters*, 122(6):065301, 2019.

Levine, Y., Wies, N., Sharir, O., Bata, H., and Shashua, A. Limits to depth efficiencies of self-attention. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 22640–22651. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/ff4dfdf5904e920ce52b48c1cef97829-Paper.pdf.

Levine, Y., Lenz, B., Lieber, O., Abend, O., Leyton-Brown, K., Tennenholtz, M., and Shoham, Y. {PMI}-masking: Principled masking of correlated spans. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3Aoft6NWFej.

Press, O. and Wolf, L. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 157–163, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-2025.

Press, O., Smith, N. A., and Levy, O. Improving transformer models by reordering their sublayers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2996–3005, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.270. URL https://www.aclweb.org/anthology/2020.acl-main.270.

Pruthi, D., Gupta, M., Dhingra, B., Neubig, G., and Lipton, Z. C. Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4782–4793, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.432. URL https://www.aclweb.org/anthology/2020.acl-main.432.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Rao, R., Liu, J., Verkuil, R., Meier, J., Canny, J. F., Abbeel, P., Sercu, T., and Rives, A. Msa transformer. *bioRxiv*, 2021.

Richter, O. and Wattenhofer, R. Normalized attention without probability cage. *arXiv preprint arXiv:2005.09561*, 2020.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., and Fergus, R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, 2019. doi: 10.1101/622803. URL https://www.biorxiv.org/content/10.1101/622803v4.

Saxton, D., Grefenstette, E., Hill, F., and Kohli, P. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1gR5iR5FX.

Schuster, M. and Nakajima, K. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5149–5152, 2012. doi: 10.1109/ICASSP.2012.6289079.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/

v1/P16-1162. URL https://www.aclweb.org/anthology/P16-1162.

Sharir, O., Levine, Y., Wies, N., Carleo, G., and Shashua, A. Deep autoregressive models for the efficient variational simulation of many-body quantum systems. *Physical review letters*, 124(2):020503, 2020.

Shazeer, N. Experimental t5 pre-trained model checkpoints. https://github.com/google-research/text-to-text-transfer-transformer/blob/master/released_checkpoints.md, 4 2020. (Accessed on 06/06/2021).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019.

Wang, S., Sun, S., Li, Z., Zhang, R., and Xu, J. Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLOS Computational Biology*, 13(1):1–34, 01 2017. doi: 10.1371/journal.pcbi.1005324. URL https://doi.org/10.1371/journal.pcbi.1005324.

Weissenborn, D., Täckström, O., and Uszkoreit, J. Scaling autoregressive video models. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rJgsskrFwH.

Xue, L., Barua, A., Constant, N., Al-Rfou, R., Narang, S., Kale, M., Roberts, A., and Raffel, C. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *arXiv preprint arXiv:2105.13626*, 2021a.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 483–498, Online, June 2021b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2021.naacl-main.41.

Yang, Z., Dai, Z., Salakhutdinov, R., and Cohen, W. W. Breaking the softmax bottleneck: A high-rank RNN language model. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkwZSG-CZ.