INDUCTIVE BIASES AND VARIABLE CREATION IN SELF-ATTENTION MECHANISMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-attention, an architectural motif designed to model long-range interactions in sequential data, has driven numerous recent breakthroughs in natural language processing and beyond. This work provides a theoretical analysis of the inductive biases of self-attention modules, where our focus is to rigorously establish which functions and long-range dependencies self-attention blocks prefer to represent. Our main result shows that bounded-norm Transformer layers create sparse variables: they can represent sparse Lipschitz functions of the input sequence, with sample complexity scaling only logarithmically with the context length. Furthermore, we propose new experimental protocols to support this analysis and to guide the practice of training Transformers, built around the large body of work on provably learning sparse Boolean functions.

1 INTRODUCTION

Self-attention mechanisms have comprised a dramatic paradigm shift in deep learning in recent years, appearing ubiquitously in recent empirical breakthroughs in sequence modeling and unsupervised representation learning. Starting with large-scale natural language processing (Vaswani et al., 2017), self-attention has enjoyed surprising empirical successes in numerous and diverse modalities of data. In many of these settings, self-attention has supplanted traditional recurrent and convolutional architectures, which are understood to incorporate inductive biases about temporal and translational invariances in the data. Self-attention models discard these functional forms, in favor of directly and globally modeling long-range interactions within the input context.

The proliferation of self-attention raises countless mysteries for theorists and empiricists. One fundamental question concerns its statistical properties: *How should we think about the inductive biases of self-attention models?* More specifically, we can ask: *Which functions do self-attention blocks prefer to represent? How many (approximately) distinct functions can they represent?* To this end, this work initiates an analysis of the statistical foundations of self-attention, as it is used in today's state-of-the-art models.

Our main technical contribution is a classical norm-based generalization bound for a Transformer network, which can be extended to related and future architectures via a modular abstraction of attention mechanisms. Combining this with example families of discrete functions representable by bounded-norm Transformers, we describe their inductive bias as what we term *sparse variable creation*: they can fit sparse functions of the input sequence, with sample complexity scaling only logarithmically with the context size.

We accompany this analysis with an experimental verification of the sample complexity scaling laws predicted by the theory. We discuss how to extend and repurpose this experimental protocol of benchmarking long-context sequence models on synthetic "cryptographic" tasks. We find that Transformers trained with gradient descent can learn sparse parities with noise, which may be of independent interest, exposing the empirical study of Transformers to the rich theory established around this problem.

1.1 RELATED WORK

The direct precursors to modern self-attention architectures were recurrent and convolutional networks augmented with attention mechanisms (Bahdanau et al., 2014; Luong et al., 2015; Xu et al.,

2015). Landmark work by Vaswani et al. (2017) demonstrated significantly improved machine translation models via self-attention only; autoregressive language models followed shortly (Liu et al., 2018; Radford et al., 2018; 2019; Brown et al., 2020), as well as self-supervised representation learning (Devlin et al., 2018). More recently, self-attention has demonstrated promise in computer vision (Dosovitskiy et al., 2020), protein folding (Jumper et al., 2021), theorem proving (Polu & Sutskever, 2020), program synthesis (Chen et al., 2021b), and reinforcement learning (Chen et al., 2021a; Janner et al., 2021).

Norm-based generalization bounds. There is a vast body of literature dedicated to establishing statistical guarantees for neural networks, including VC-dimension and shattering bounds (dating back to Anthony et al. (1999)). In recent years, various generalization bounds have been established for various architectures under norm bounds including (but not limited to) Bartlett et al. (2017); Neyshabur et al. (2015; 2017); Long & Sedghi (2019); Chen et al. (2019). Our work complements these bounds by establishing norm-based bounds for attention models.

Theory for attention models. Our work complements various existing theoretical perspectives on attention-based models. Vuckovic et al. (2020) formulate a dynamical system abstraction of attention layers, arriving at similar Lipschitz constant calculations to ours (which are coarser-grained, since they focus on contractivity and stability rather than finite-sample statistical guarantees). Zhang et al. (2019); Snell et al. (2021) study idealizations of optimization problems for self-attention heads. The representational power of Transformers has been studied in the paradigm of universal function approximation and Turing-completeness (Dehghani et al., 2018), and the infinite-width limit (Hron et al., 2020; Yang, 2020). Wei et al. (2021), motivated by the same theme of statistically meaningful approximation of circuits, analyze a margin amplification procedure and apply it to Transformers, focusing on obtaining a mild dependence of the sample complexity on depth (rather than context size). [1]

2 Background and notation

Notation. $\|\cdot\|_2$ denotes the spectral norm for matrices, and $\|\cdot\|_{p,q}$ denotes the (p,q) matrix norm where the p-norm is over columns and q-norm over rows. $\|\cdot\|_p$ denotes the ℓ_p norm for vectors. For ℓ_2-norm, we drop the subscript. B is generally used to quantify bounds on norms of matrices and L for Lipschitz constants.

Covering numbers and uniform generalization bounds. Our main technical contribution is a generalization bound arising from carefully counting the number of functions representable by a Transformer. This main complexity notion we use is covering number.

We will use the following definition of ∞-norm covering number adapted from Zhang (2002)

Definition 2.1. *(Covering Numbers) For a given class of vector-valued functions $\mathcal{F}$, the covering number $\mathcal{N}_\infty(\mathcal{F};\varepsilon;\{z^{(i)}\}_{i=1}^m;\|\cdot\|)$ is the smallest size of a collection (a cover) $\mathcal{C}\subset\mathcal{F}$ such that $\forall f\in\mathcal{F},\exists\widehat{f}\in\mathcal{C}$ satisfying $\max_i\|f(z^{(i)})-\widehat{f}(z^{(i)})\|\le\varepsilon$. Further, define $\mathcal{N}_\infty(\mathcal{F},\varepsilon,m,\|\cdot\|)=\sup_{z^{(1)}\dots z^{(m)}}\mathcal{N}_\infty(\mathcal{F};\varepsilon;z^{(1)},\dots,z^{(m)},\|\cdot\|)$.*

If $\mathcal{F}$ is real-valued (instead of vector-valued), we drop the norm from the notation. Furthermore for functions parameterized by a set of parameters Θ, we exploit the notation to replace $\mathcal{F}$ by Θ.

Recall from Zhang (2002) that for the class of linear functions, $\mathcal{F}_{\text{lin}}=\{x\mapsto w\cdot x\ :\ w\in\mathbb{R}^d,\|w\|_s\le B_W\}$, we have the covering number bound of $\mathcal{N}_\infty(\mathcal{F};\varepsilon;\{x^{(i)}\}_{i=1}^m)\le O(B_X^2B_W^2/\varepsilon^2\cdot\log(B_XB_Wm/\varepsilon)$, where $\|x^{(i)}\|\le B_X$ for $i\in[m]$. Importantly, note that the covering number has a mild dependence on m, only logarithmic; this logarithmic dependence on m will be helpful when we turn our analysis to the capacity of attention mechanisms.

Generalization bounds. This work directly focuses on providing log covering number bounds, which determine the generalization error (via standard techniques involving Rademacher complexity

[1] Our analysis naturally plugs into theirs, upon reconciling their additional assumptions. Quoting the discussion following Theorem 4.1 of (Wei et al., 2021): *"The correct norm-based Rademacher complexity bound to use for Transformers is unclear."*

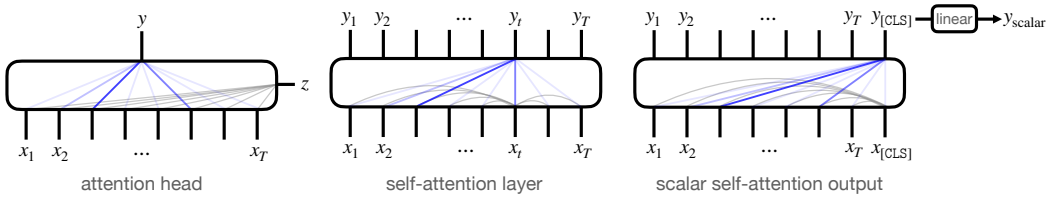

Figure 1: Diagrams of attention modules described in Section 3: alignment scores (grey edges) determine normalized attention weights (blue), which are used to mix the inputs $x_{1:T}$. *Left:* Attention with a general context. *Center:* Stackable self-attention layer, where $x_{1:T}$ is the input as well as the context. *Right:* Auxiliary `[CLS]` token to extract a single scalar from a self-attention layer, providing a real-valued function class for classification or regression tasks.

and chaining). Roughly speaking, the sample complexity for obtaining ε generalization error scales polynomially with the log covering number. See the Appendix A for these standard relationships.

3 Abstractions of attention and self-attention

Attention is not straightforward to define — unlike other architectural components such as convolutions and residual connections, it is a broader model design principle, with numerous variations possible. In this section, we present an abstraction of attention mechanisms, guided by the different manifestations discussed in (Luong et al., 2015), with the goal of enabling a more unified and modular statistical analysis. Subsequently, we show how to represent the Transformer (the predominant attention-based architecture) as a special case of this formulation. Our goal is not necessarily an all encompassing formulation of attention mechanisms, but rather an abstraction that helps to guide general design principles and theoretical analysis.

3.1 Attention

Intuitively, we would like to capture the notion that a variable selects ("attends to") a part of the context on which it will depend, based on a learned function of global interactions (see Figure 1). To this end, we define an *attention head* as a function which maps a collection X of T inputs $\{x_t \in \mathcal{X}\}_{t=1}^T$ (e.g. the tokens in a sentence, pixels in an image, or intermediate activations in a neural network) and a context $z \in \mathcal{Z}$ to an output $y \in \mathcal{Y}$. In this work, we will exclusively consider $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ to be $\mathbb{R}^d$; d is the *embedding dimension*, and we define the matrix of inputs $X = [x_1 x_2 \dots x_T]^\top \in \mathbb{R}^{T \times d}$. An attention head uses z to select the inputs in X to which the output y will "attend", which we formalize below:

Definition 3.1 (Attention head)**.** *An attention head is a function $f : \mathcal{X} \to \mathcal{Y}$, specified by an "alignment score" function* $\mathsf{Score} : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ *parameterized by $\theta_s \in \Theta_s$, normalization function* $\mathsf{Norm} : \mathbb{R}^T \to \Delta^{T-1}$*, and position-wise transformations $\phi_{\text{in}} : \mathcal{X} \to \mathcal{V}, \phi_{\text{out}} : \mathcal{V} \to \mathcal{Y}$ parameterized by $\theta_{\text{in}} \in \Theta_{\text{in}}$ and $\theta_{\text{out}} \in \Theta_{\text{out}}$. The output of an attention head on input $X \in \mathcal{X}^T, z \in \mathcal{Z}$ is*

$$
\begin{aligned}
y &= \phi_{\text{out}}\left(\sum_{t=1}^{T}\Big[\mathsf{Norm}\Big(\mathsf{Score}(x_1, z; \theta_s), \dots \mathsf{Score}(x_T, z; \theta_s)\Big)\Big]_t \phi_{\text{in}}(x_t; \theta_{\text{in}});\ \theta_{\text{out}}\right) \\
&= \phi_{\text{out}}\left(\phi_{\text{in}}(X; \theta_{\text{in}})^\top \mathsf{Norm}\Big(\mathsf{Score}(x_1, z; \theta_s), \dots \mathsf{Score}(x_T, z; \theta_s)\Big);\ \theta_{\text{out}}\right)
\end{aligned}
$$

where $\phi_{\text{in}}(X; \theta) = [\phi_{\text{in}}(x_1; \theta) \dots \phi_{\text{in}}(x_T; \theta)]^\top$ denotes the row-wise application of ϕ_{in}.

The above definition corresponds to left most diagram in Figure 1. Here, $\mathcal{V}$ is a vector space of input representations "mixed" by the normalized alignment scores; in this work, we will set $\mathcal{V} = \mathbb{R}^k$. A function class of attention heads is induced by specifying parameter classes for $\{\Theta_s, \Theta_{\text{in}}, \Theta_{\text{out}}\}$.

3.2 SELF-ATTENTION AND TRANSFORMERS

A self-attention head is a special case of an attention head, in which the context z is one of the inputs x_t themselves: interactions between elements in X are used to select the elements of X on which f depends. In this case, we will use the term *context* to denote X. The focus of this work is to analyze the inductive biases of such a construction. For example, for a self-attention head (see Figure 1 (center)), we would have that the t-th component is:

$$y_t = \phi_{\text{out}}\left(\phi_{\text{in}}(X;\theta_{\text{in}})^\top \mathsf{Norm}(\mathsf{Score}(X, x_t;\theta_s));\theta_{\text{out}}\right),$$

Specifically, for the Transformer architecture, it will be convenient to define all T outputs of this layer at once.

Definition 3.2 (Transformer self-attention layer)**.** *A Transformer attention layer is a collection of T attention heads with outputs $y_1, \ldots, y_T$, specified by the following choices of function classes (with shared parameters between the heads) where the context for y_t is x_t.*

- $\mathsf{Score}(x, x_t; \{W_Q, W_K\}) := x_t^\top W_Q W_K^\top x, \quad W_Q, W_K \in \mathbb{R}^{d\times k}$ *(for output y_t)*
- $\phi_{\text{in}}(x; W_V) := W_V^\top x, \quad W_V \in \mathbb{R}^{d\times k}$
- $\phi_{\text{out}}(x; W_C) := W_C^\top \sigma(x), \quad W_C \in \mathbb{R}^{k\times d}$, *$\rho$-Lipschitz function $\sigma : \mathbb{R} \to \mathbb{R}$ applied position-wise, with $\sigma(0) = 0$.*
- $\mathsf{Norm}(x) := \mathsf{softmax}(x) = \frac{\exp(x)}{\mathbf{1}^\top \exp(x)}$

In summary, defining $Y := [y_1 y_2 \ldots y_T]^\top \in \mathbb{R}^{T\times d}$ and $[\mathsf{RowSoftmax}(M)]_{t,:} := \mathsf{softmax}(M_{t,:})$, we have

$$Y = \sigma\left(\mathsf{RowSoftmax}\left(XW_Q(XW_K)^\top\right)XW_V\right)W_C.$$

Functions from this class $\mathcal{F}_{\text{tf-layer}}$ of Transformer layers map $\mathbb{R}^{T\times d}$ to itself, so that instances from this function class can be composed. Although Definition 3.2 only contains the "self-attention" component, it is not merely a simplified idealization of the full Transformer architecture used in practice. Only a few discrepancies remain (positional embeddings, layer normalization, parallel heads, residual connections, position-wise feedforward networks), which we discuss in Section 4.3 and the appendix.

We introduce one more construction: the canonical way to extract a scalar prediction from the final layer of a Transformer. This is the setup used by the classification modules in BERT (Devlin et al., 2018) and all of its derivatives. For a context of size T, a Transformer layer with $T + 1$ inputs is constructed, with a special input index `[CLS]`. The input at this position is a vector $x_{\texttt{[CLS]}} \in \mathbb{R}^d$ (which can be considered as a constant, a part of the input, or a trainable parameter); the output is a linear function $w^\top y_{\texttt{[CLS]}}$, for a trainable parameter $w \in \mathbb{R}^d$. This defines a class of functions mapping $\mathbb{R}^{T\times d} \to \mathbb{R}$, parameterized by a Transformer layer's parameters and w, which we call the class of *scalar-output Transformers* $\mathcal{F}_{\text{tf-scalar}}$.

4 CAPACITY MEASURES OF ATTENTION MODULES

In this section, we present our main technical results along with an overview of the proofs. Section 4.1 bounds the capacity of a general attention head. Section 4.2 instantiates this bound for the case of a single Transformer self-attention head. Section 4.3 generalizes this bound for full depth-L Transformer networks. Our sample complexity guarantees which scale only *logarithmically* in the context length T, providing rigorous grounding for the intuition that the architecture's inductive bias selects sparse functions of the context. Lastly, in Section 4.4, we complement this capacity analysis by exhibiting examples of functions expressible using low-norm Transformer architectures.

Note: Throughout this section, assume that $\|x_t\|_2 \le B$ for all $t \in [T]$. Note that this allows for the Frobenius norm $\|X\|_F$ to scale with $\sqrt{T}$. The key challenge throughout our analysis is to avoid incurring factors of norms which take a sum over the t dimension, by analyzing the attention parameters in appropriately chosen geometries.

4.1 CAPACITY OF AN ATTENTION HEAD

Let $\mathcal{F}_{\text{head}}$ be the function class corresponding to any particular attention head architecture, that is, $\mathcal{F}_{\text{head}}$ contains functions of the form:

$$f_{\text{head}}(X, z; \theta_s, \theta_{\text{in}}, \theta_{\text{out}}) = w^T \phi_{\text{out}}\left(\phi_{\text{in}}(X; \theta_{\text{in}})^\top \mathsf{Norm}(\mathsf{Score}(X, z; \theta_s)); \theta_{\text{out}}\right)$$

For simplicity, we will focus only on the attention part and assume that ϕ_{out} is a fixed function (no parameters) and w are fixed. It is not hard to handle these even if allowed to be trainable. For the case of Transformers, we handle this more generally (see Appendix A.5)

Assumption 4.1. *We make the following assumptions:*

1. ϕ_{out} *is* L_{out}*-Lipschitz in the* ℓ_2*-norm, that is,* $\|\phi_{\text{out}}(a) - \phi_{\text{out}}(b)\| \le L_{\text{out}}\|a - b\|$ *for all* $a, b \in \mathbb{R}^k$.
2. ϕ_{in} *is* B_{in}*-bounded in* ℓ_2*-norm, that is,* $\|\phi_{\text{in}}(a; \theta_{\text{in}})\| \le B_{\text{in}}\|a\|$ *for all* $a \in \mathbb{R}^d$.
3. Norm *is* L_{Norm}*-Lipschitz in the* ℓ_∞*-norm, that is,* $\|\mathsf{Norm}(a) - \mathsf{Norm}(b)\|_\infty \le L_{\mathsf{Norm}}\|a - b\|_\infty$ *for all* $a, b \in \mathbb{R}^k$
4. $\|w\|$ *is bounded by* B_w.

We prove the following bound on the covering number of the class of general attention heads,

Theorem 4.2 (Attention head capacity)**.** *Under Assumptions 4.1, the covering number of* $\mathcal{F}_{\text{head}}$ *satisfies*

$$\begin{aligned}
&\log \mathcal{N}_\infty\left(\mathcal{F}_{\text{head}}; \varepsilon; \left\{(X^{(i)}, z^{(i)})\right\}_{i=1}^m\right) \\
&\quad \le \inf_{\alpha \in [0,1]} \Bigg[\log \mathcal{N}_\infty\left(\mathcal{F}_{\mathsf{Score}}; \frac{\alpha\varepsilon}{2L_{\mathsf{Norm}} L_{\text{out}} B_w B_{\text{in}}}; \{(x_t^{(i)}, z^{(i)})\}_{i\in[m], t\in[T]}\right) \\
&\qquad\qquad + \log \mathcal{N}_\infty\left(\mathcal{F}_{\text{in}}; \frac{(1-\alpha)\varepsilon}{L_{\text{out}} B_w}; \{x_t^{(i)}\}_{i\in[m], t\in[T]}; \|\cdot\|_2\right)\Bigg],
\end{aligned}$$

where $\mathcal{F}_{\mathsf{Score}} = \{(x, z) \mapsto \mathsf{Score}(x, z; \theta_s) : \theta_s \in \Theta_s\}$*, and* $\mathcal{F}_{\text{in}} = \{x \mapsto \phi_{\text{in}}(x; \theta_{\text{in}}) : \theta_{\text{in}} \in \Theta_{\text{in}}\}$.

Note that the bound is in terms of the $\mathcal{N}_\infty$ covering number of functions that dependent on dimensions d or k and not T. The effect of T only shows up in the number of samples to cover. The $\mathcal{N}_\infty$ number for many classes scales only logarithmically with the number of samples (for eg, linear functions Zhang (2002)). This is exactly what allows us to get a $\log T$ dependence for Transformers.

Proof overview. In order to prove the bound, we first show a Lipschitzness property of $f_{\mathsf{tf\text{-}head}}$. This property allows us to construct the cover by into covering both $\mathcal{F}_{\mathsf{Score}}$ and $\mathcal{F}_{\text{in}}$.

Lemma 4.3 (ℓ_∞-Lipschitzness of $f_{\mathsf{tf\text{-}head}}$)**.** *For any* $\theta_s, \bar{\theta}_s \in \Theta_s, \theta_{\text{in}}, \bar{\theta}_{\text{in}} \in \Theta_{\text{in}}$*; for all* $X \in \mathbb{R}^{T\times d}$ *such that* $\|X\|_{2,\infty} \le 1$*,*

$$\begin{aligned}
&|f_{\text{head}}(X, z; \theta_s, \theta_{\text{in}}, w) - f_{\text{head}}(X, z; \bar{\theta}_s, \bar{\theta}_{\text{in}}, w)| \\
&\quad \le 2L_{\mathsf{Norm}} L_{\text{out}} B_w B_{\text{in}} \left\|\mathsf{Score}(X, z; \theta_s) - \mathsf{Score}(X, z; \bar{\theta}_s)\right\|_\infty \\
&\qquad + L_{\text{out}} B_w \left\|\phi_{\text{in}}(X; \theta_{\text{in}}) - \phi_{\text{in}}(X; \bar{\theta}_{\text{in}})\right\|_{2,\infty}.
\end{aligned}$$

The most crucial part of this proof is to ensure that we do not get a spurious T dependence when accounting for the attention mechanism. The key observation here is that the attention part of the network is computes using Norm, which is a probability distribution. This allows us to move to the maximum (ℓ_∞) error over T tokens instead of sum (ℓ_1), which could potentially incur a T factor. Furthermore, this allows us to combine all samples and tokens and construct a ∞ cover for mT samples.

4.2 CAPACITY OF A TRANSFORMER SELF-ATTENTION HEAD

Let us now look at the case of a Transformer self-attention head and instantiate the covering bound. Recall that a Transformer self-attention head can be described as

$$f_{\mathsf{tf\text{-}head}}(X; W_V, W_{QK}, w) := w^\top \sigma\left(W_V^\top X^\top \mathsf{softmax}\left(X W_{QK}^\top x_\tau\right)\right)$$

which is obtained from the general formulation by setting the context to be x_τ, $\mathsf{Score}(X, x_\tau; W_{QK}) = XW_{QK}^\top x_\tau$, $\mathsf{Norm} = \mathsf{softmax}$ and $\phi_{out} = \sigma$.[2]

Let us define $\mathcal{F}_{\text{tf-head}}$ to be the class of all $f_{\text{tf-head}}$ with parameters satisfying $\|W_V^T\|_{2,\infty} \leq B_{2,\infty}^V$, $\|W_V\| \leq B^V$ and $\|W_{QK}\|_{2,\infty} \leq B_{2,\infty}^{QK}$. Since all these matrices have dimensions dependent on d and k, bounding their norms does not hide a T dependence.

We obtain the following bound on the covering number of $\mathcal{F}_{\text{tf-head}}$ as a corollary of Lemma 4.2:

Corollary 4.4. *For all $X \in \mathbb{R}^{T\times d}$ such that $\|X\|_{2,\infty} \leq 1$, the covering number of $\mathcal{F}_{\text{tf-head}}$ satisfies*

$$\log \mathcal{N}_\infty(\mathcal{F}_{\text{tf-head}}; \varepsilon; X^{(1)}, \ldots, X^{(m)}) \lesssim \frac{\left((kB_{2,\infty}^V L_\sigma B_w)^{\frac{2}{3}} + (\frac{1}{2} d B_{2,\infty}^{QK} L_\sigma B_w B^V)^{\frac{2}{3}}\right)^3 \log(mT)}{\varepsilon^2}$$

The tilde hides logarithmic dependencies on quantities besides m and T.

Our bounds have a logarithmic dependence on T highlighting the inductive bias of the transformer towards selecting sparse functions of the context.

Proof overview. The above result follows from bounding the covering numbers of $\mathcal{F}_{QK} := \{x \to x_\tau^\top W_{QK} x : \|W_{QK}\|_{2,\infty} \leq B_{2,\infty}^{QK}\}$ and $\mathcal{F}_V := \{x \to W_V^\top x : \|W_V^T\|_{2,\infty} \leq B_{2,\infty}^V, \|W_V\| \leq B^V\}$.

Note that $|x_\tau^\top W_{QK} x - x_\tau^\top W_{QK} x| \leq \|W_{QK}x - W_{QK}x\|$ since $\|x_\tau\| \leq 1$, so the covering number of $\mathcal{F}_{QK}$ is at most the covering number of the class of functions of the form $x \mapsto W_{QK}x$. Therefore, a bound on the vector-valued linear function class suffices to handle both covering numbers. We derive the following covering bound which gives the desired result.

Lemma 4.5. *Let $\mathcal{W} : \{W \in \mathbb{R}^{d_1\times d_2} : \|W\|_{2,\infty} \leq B_\infty\}$, and consider the function class $\mathcal{F} : \{x \mapsto Wx : W \in \mathcal{W}\}$. For any $\varepsilon > 0$ and vectors $x_1, \ldots, x_N \in \mathbb{R}^{d_1}$ satisfying $\|x_n\| \leq 1$ for all $n \in [N]$,*

$$\log \mathcal{N}_\infty(\mathcal{F}; \varepsilon; x^{(1)}, \ldots, x^{(N)}; \|\cdot\|_2) \lesssim \frac{c(d_2 B_\infty)^2 \log(N)}{\varepsilon^2}$$

for some constant c.

The proof of Lemma 4.5 actually proves a somewhat stronger bound for the function class $\{W \in \mathbb{R}^{d_1\times d_2} : \|W\|_{2,\infty} \leq B_\infty, \|W\|_{1,\infty} \leq B_1\}$, with $d_2 B_\infty B_1$ in the numerator instead of $(d_2 B_\infty)^2$, but we have kept the latter formulation for simplicity of presentation.

Positional embeddings. In practice, the permutation-invariant symmetry of a Transformer network is broken by adding a *positional embedding* matrix $P \in \mathbb{R}^{T\times d}$ to the input X at the first layer. Often in practice, the embedding matrix is fixed and non-trainable. Our results extend to this setting in a straightforward way (see Appendix A.4 for an exposition). In case these matrices are to be trained, they can add a dependence on T. However, assuming further constraints on these embeddings (in terms of norm) can potentially be mitigated.

Layer normalization. State-of-the-art Transformer networks are trained with layer normalization modules (Ba et al., 2016), which is generally understood to aid optimization. In fact, it is proves to be useful in the analysis of full attention blocks (see Appendix A.5) as it keeps the norm of the embedding of each token bounded. Removing these layers would lead to a worse dependence on the spectral norm of the matrices.

4.3 Capacity of full attention blocks

In this section, we will extend our results for L-layer transformer blocks. Denote the weights of layer i by $W^{(i)} := \left\{W_Q^{(i)}, W_K^{(i)}, W_V^{(i)}, W_C^{(i)}\right\}$ such that $\left\|W_V^{(i)}\right\|_2, \left\|W_K^{(i)} {W_Q^{(i)}}^\top\right\|_2, \left\|W_C^{(i)}\right\|_2 \leq C_2$,

[2] We have collapsed $W_Q W_K^\top$ to a single matrix for ease of presentation, and for simplicty we will consider w fixed. See Appendix 4.3 for a more general analysis.

$\left\|W_V^{(i)}\right\|_{2,\infty}, \left\|W_K^{(i)} W_Q^{(i)\top}\right\|_{2,\infty}, \left\|W_C^{(i)}\right\|_{2,\infty} \leq C_\infty$ and $\|w\| \leq C_w$.[3] Further denote the set of weights up to layer i by $W^{1:i} = (W^{(1)}, \ldots, W^{i-1})$. Denote the input representation of layer i by $g(X; W^{1:i})$. We can now inductively define g starting with $g(X; W^{1:1}) = X$ (the input):

$$g\left(X; W^{1:i+1}\right) = \Pi_{norm}\left(\sigma\left(\Pi_{norm}\left(f\left(g\left(X; W^{1:i}\right); W^{(i)}\right)\right)\right) W_C^{(i)}\right) \text{ with}$$
$$f\left(Z; \{W_Q, W_K, W_V, \cdot\}\right) = \mathsf{RowSoftmax}\left(ZW_Q\left(ZW_K\right)^\top\right) ZW_V,$$

where Π_{norm} denotes layer normalization[4] applied to each row. We use a slightly modified version of LayerNorm where instead of normalizing to norm 1, we project it to the unit ball. The final output value is a scalar $g_{out}(X; W^{1:L+1}, w) = w^\top g\left(X; W^{1:i+1}\right)$ `[CLS]` computing using the `[CLS]` token.

Theorem 4.6 (Simplified version of Theorem A.14)**.** *Let $\mathcal{F}^{(L)}$ represent the class of functions of L-layer transformer network satisfying the norm bounds (as above). Then, we have*

$$\log \mathcal{N}_\infty(\mathcal{F}^{(L)}; \varepsilon; X^{(1)}, \ldots, X^{(m)}) \lesssim \frac{(C_2 L_\sigma)^{O(L)} d^2 L C_w C_\infty^2 \log(TM)}{\varepsilon^2}$$

Note that the dependence on T is only logarithmic even for deeper networks. The dependence on embedding dimension and $(2, \infty)$ norms of the weight matrices is quadratic. As long as the spectral norms of the matrices are bounded by 1 and σ is 1-Lipschitz (holds for sigmoids and ReLUs), the exponential dependence on L can be avoided.

4.4 Sparse Variable Creation: An inductive bias for representing sparse Boolean functions

The above analysis shows that Transformer function classes are "small" in terms of the context size. In this section, we answer the converse question: *which functions of interest can they express?* We show that bounded-norm Transformers can represent sparse Boolean functions. To state our claims, we must first choose a correspondence between Boolean strings and Transformer inputs. Motivated by position-independent token interactions in NLP (and amenability to composition), we choose a collection of pairwise approximately orthogonal vectors $\{v_1, \ldots, v_T\}$, then use the following bijection:

"Does $b_t = 1$ in the Boolean string?" $\leftrightarrow$ "Does v_t appear in the Transformer's context?"

Concretely, given a function $f : \{0,1\}^T \to \mathbb{R}$, a matrix $V \in \mathbb{R}^{T \times d}$, and $\varepsilon > 0$, our notion of representability is the existence of a function $\widehat{f} \in \mathcal{F}_{\text{tf-scalar}}$ which ε-uniformly approximates f:

$$\left|\widehat{f}(V\mathrm{diag}(b)) - f(b)\right| \leq \varepsilon, \qquad \forall b \in \{0,1\}^T.$$

We present overviews informal statements of these approximation results below, and develop the notation and precise statements in Appendix B.

As a warmup, we show that a single self-attention head can implement a family of functions which are monotonic in the number of ones in an arbitrary subset of s indices. Combined with a threshold classifier, this can be used to implement the s-sparse AND and OR gates.

Proposition 4.7 (Representing a sparse "counting" function; informal)**.** *Let $\mathcal{I} \subseteq [T]$, and $|\mathcal{I}| = s \geq 1$. There is a function $f_{\mathcal{I}}(b)$ which only depends on $r := \sum_{i \in \mathcal{I}} b_i$, and is increasing in r, such that a scalar-output Transformer can ε-uniformly approximate f, with weight norms satisfying*

$$\|W_Q\| \leq \log\left(\frac{4Ts}{\varepsilon}\right), \quad \|W_K\| \leq 2, \quad \|W_V\| \leq \sqrt{2}s, \quad \|W_C\| \leq s, \quad \|w\| = 1.$$

[3] See Appendix A.5 for the general statement and analysis with norm bounds dependent on each layer

[4] LayerNorm allows for the norms of the outputs of each token in each layer to remain bounded by 1. Note that the norm of the entire input can still have a dependence on T. Our results would go through with a worse dependence on the spectral norms if we were to remove layer-norm.

Combining this construction with a standard result for universal function approximation using feed-forward networks, we show how the full Transformer architecture combines sparsity -promoting attention heads with the ability to memorize arbitrary functions of the sparsified inputs. We interpret this as a *sparse variable creation* mechanism, promoting the sample-efficient learning of arbitrary sparse Boolean functions.

Proposition 4.8 (Sparse variable creation; informal)**.** *Let $f(b) : \{0,1\}^T \to \mathbb{R}$ be a function which only depends on a set of inputs $\mathcal{I}$, with $|\mathcal{I}| \le s$. A scalar-output Transformer classifier with a 3-layer ReLU network can represent f exactly, with weight norms satisfying*

$$\|W_Q\| \lesssim \log T, \quad \|W_K\| \le 2, \quad \|W_V\| \le \sqrt{2}s, \quad \|W_C\| \le s,$$

and MLP weight matrices bounded by

$$\|W_1\|, \|W_2\| \lesssim s2^s, \quad \|w\| \lesssim \max_{b \in \{0,1\}^T} |f(b)|.$$

Again, note that the dependence on T is only logarithmic; the exponential in s dependence is information theoretically necessary (due to the ability to *create* complex functions on these s variables). The intuition for both constructions is to find low-norm attention matrices which allow the softmax function to approximate the uniform mixture on the inputs from the sparse index set $\mathcal{I}$. For Proposition 4.7, it suffices to apply the Transformer layer's position-wise linear function to this mixture. For Proposition 4.8, we extend this gadget to map the input to a projection of the s-dimensional hypercube onto the simplex, and use a standard construction for a ReLU network to memorize all of f's outputs. The logarithmic factors arise from analyzing the "leakage" caused by the softmax and approximate orthogonality.

5 Empirical scaling laws for learning Boolean gates

Our theoretical analysis has shown that Transformers can represent sparse Boolean functions, with sample complexity scaling mildly with the context size. In this section, we present an empirical study of whether Transformer architectures (as they are trained and used in state-of-the-art language modeling) exhibit these scalings in practice.

We introduce a rigorous benchmark for probing the empirical sample complexity of a Transformer: attribute-efficient learning of a planted sparse Boolean function. We choose a family of distinct distributions $\{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$ on $\{0,1\}^T \times \{0,1\}$, corresponding to supervised learning problems, such that the feature distributions are identical, and the uniform mixture $\frac{1}{N}\sum_{i=1}^N \mathcal{D}_i$ is invariant under all permutations of the indices $1, \ldots, T$. We then select an $i^* \in [N]$ uniformly at random, train a Transformer binary classifier on m samples from $\mathcal{D}_{i^*}$, then evaluate the generalization error via cross-validation. Since the architecture, initialization, training algorithm, and training data are permutation-invariant, at least $\Omega(\log N)$ samples are required to learn this distribution: one sample can only reveal a single bit of information about i^*, via its binary label. We are interested in the empirical scaling of the sufficient sample size m to solve this problem, in terms of N.

Learning sparse AND gates. The simplest instantiation of this experimental framework is a hidden s-way AND under the uniform distribution (i.e. T i.i.d. random bits). The Transformer must learn which of the $N = \binom{T}{s}$ combinations of inputs determines the label, which requires $m \ge \Omega(s \log T)$ samples. The theory predicts the sample complexity of learning a bounded-norm Transformer should match this scaling in T. With hyperparameters typical of Transformer setups used for natural data, we indeed observe that the empirical sample complexity scales sublinearly with T. Figure 2 summarizes our findings; details are provided in Appendix C.1.

Learning sparse parities. We can also replace the sparse AND operation with XOR: the label is the parity of a hidden subset of input bits. This variant emphasizes the "cryptographic" nature of this experimental setup, due to its known computational hardness. $\Omega(T^s)$ statistical queries (thus, batch gradient descent steps) are necessary (Kearns, 1998); in the presence of noise, the fastest known algorithms for learning parities with noise require $T^{\Omega(s)}$ time (Valiant, 2012). This computational-statistical gap introduces entanglements that make it difficult to design an informative scaling-law experiment for deep learning models. However, the fact that Transformer models can fit sparse

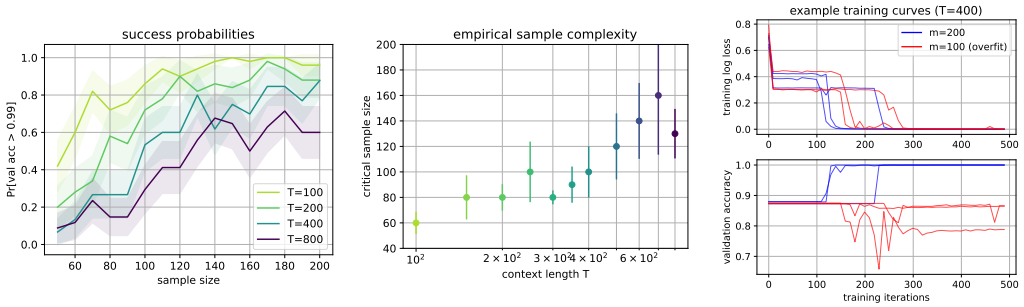

Figure 2: Statistically probing a Transformer by training it on a 3-way AND of a hidden subset of i.i.d. random bits. *Left: Center:* Sublinear scaling of the empirical sample complexity. *Right:* Example training curves in the {overfitting, correct} regimes: $T = 400, m = \{100, 200\}$.

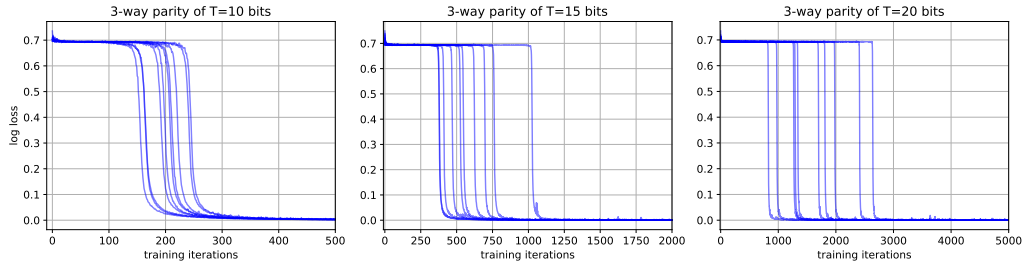

Figure 3: A curious empirical finding: Transformers can learn sparse parities. 10 loss curves (with online batches) are given for this setup with $s = 3, T = \{10, 15, 20\}$, showing abrupt phase transitions from random guessing to perfect classification.

parities at all raises an intriguing question: if theory suggests that "exhaustive search-like" methods are computationally necessary for this problem, why does local search (i.e. gradient-based training) succeed? We leave this *computational* (as opposed to statistical) mystery as an open direction for future work.

6 Conclusion and Future Work

We have presented a theoretical analysis of the inductive biases of self-attention models, finding that they can learn sparse Boolean functions with sample complexity scaling logarithmically in the context length. We call this phenomenon *sparse variable creation*. Our analysis is accompanied by new empirical probes involving training Transformers on sparse Boolean functions, where we corroborate this scaling of the sample complexity in practice. We believe our capacity bounds are improvable (we have only sought to obtain an optimal dependence on T). Incorporating aspects of computation and depth remains a perennial challenge in this line of inquiry.

An exciting line of empirical work has made progress on understanding and interpreting state-of-the-art Transformer language models by examining the activations of their attention mechanisms (Clark et al., 2019; Tenney et al., 2019; Rogers et al., 2020). In some cases, these works have found instances in which transformers seem to have learned (sparse) features that are reminiscent of sparse hand-crafted features used in natural language processing. Reconciling our theoretical foundations work with this area of *BERTology* is an avenue for future synthesis.

Building further upon the principles of attention constitutes an vibrant frontier of empirical research (Tolstikhin et al., 2021; Lee-Thorp et al., 2021; Jaegle et al., 2021b;a; d'Ascoli et al., 2021). We hope that the theoretical foundations and experimental probes presented in this paper will assist in further expanding the breadth of applications of self-attention, and developing more compute-efficient, data-efficient, controllable, and reliable algorithmic interventions.

REFERENCES

Martin Anthony, Peter L Bartlett, and Peter L Bartlett. *Neural network learning: Theoretical foundations*, volume 9. cambridge university press Cambridge, 1999.

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

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Peter Bartlett, Dylan J Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1706.08498*, 2017.

Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002. URL `http://www.jmlr.org/papers/v3/bartlett02a.html`.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *arXiv preprint arXiv:2106.01345*, 2021a.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harri Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021b.

Minshuo Chen, Xingguo Li, and Tuo Zhao. On generalization bounds of a family of recurrent neural networks. *arXiv preprint arXiv:1910.12947*, 2019.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does BERT look at? an analysis of BERT's attention. *arXiv preprint arXiv:1906.04341*, 2019.

Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv preprint arXiv:2103.10697*, 2021.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Richard M Dudley. The sizes of compact subsets of hilbert space and continuity of gaussian processes. *Journal of Functional Analysis*, 1(3):290–330, 1967.

Jiri Hron, Yasaman Bahri, Jascha Sohl-Dickstein, and Roman Novak. Infinite attention: Nngp and ntk for deep attention networks. In *International Conference on Machine Learning*, pp. 4376–4386. PMLR, 2020.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021a.

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. *arXiv preprint arXiv:2103.03206*, 2021b.

Michael Janner, Qiyang Li, and Sergey Levine. Reinforcement learning as one big sequence modeling problem. *arXiv preprint arXiv:2106.02039*, 2021.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

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

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*, 2021.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018.

Philip M Long and Hanie Sedghi. Generalization bounds for deep convolutional neural networks. *arXiv preprint arXiv:1905.12600*, 2019.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

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

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

Michael A Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, 2015.

Ryan O'Donnell. Analysis of boolean functions. *arXiv preprint arXiv:2105.10386*, 2021.

Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*, 2020.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020.

Charlie Snell, Ruiqi Zhong, Dan Klein, and Jacob Steinhardt. Approximating how single head attention learns. *arXiv preprint arXiv:2103.07601*, 2021.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pp. 11–20. IEEE, 2012.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

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

Colin Wei, Yining Chen, and Tengyu Ma. Statistically meaningful approximation: a case study on approximating turing machines with transformers. *arXiv preprint arXiv:2107.13163*, 2021.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pp. 2048–2057. PMLR, 2015.

Greg Yang. Tensor programs ii: Neural tangent kernel for any architecture. *arXiv preprint arXiv:2006.14548*, 2020.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank J Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *arXiv preprint arXiv:1912.03194*, 2019.

Tong Zhang. Covering number bounds of certain regularized linear function classes. *Journal of Machine Learning Research*, 2(Mar):527–550, 2002.

A PROOFS OF CAPACITY BOUNDS

In this section we present the full proofs (including the omitted proofs) of our capacity bounds. We also cover relevant background and useful technical lemmas.

A.1 RADEMACHER COMPLEXITY AND GENERALIZATION BOUNDS

Here we briefly review Rademacher complexity and corresponding generalization bounds. We refer the reader to Bartlett & Mendelson (2002) for a more detailed exposition.

Definition A.1 (Empirical Rademacher complexity)**.** *For a given class of functions $\mathcal{F} = \{f : \mathcal{X} \to \mathbb{R}\}$ and $\{z^{(i)} \in \mathcal{X}\}_{i=1}^m$, the empirical Rademacher complexity $\widehat{\mathcal{R}}(\mathcal{F}; z^{(1)}, \ldots, z^{(m)})$ is defined as*

$$\widehat{\mathcal{R}}(\mathcal{F}; z^{(1)}, \ldots, z^{(m)}) = \frac{1}{m}\mathbb{E}_\varepsilon \left[\sup_{f\in\mathcal{F}} \sum_{i=1}^m \varepsilon_i f(z^{(i)})\right],$$

where ε is a vector of m i.i.d. Rademacher random variables ($\Pr[\varepsilon_i = 1] = \Pr[\varepsilon_i = -1] = 1/2$).

It is standard to obtain a generalization guarantee from the Rademacher complexity of a function class:

Theorem A.2 (Bartlett & Mendelson (2002))**.** *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathbb{R}$ and let $\ell : \mathbb{R} \times \mathbb{R}$ be a b-bounded loss function that is L-Lipschitz in its first argument. For a given function class $\mathcal{F}$, let $\mathsf{risk}(f; \mathcal{D}) := \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(f(x), y)]$ and $\widehat{\mathsf{risk}}\left(f; (z^{(i)}, y^{(i)})_{i=1}^m\right) := \frac{1}{m}\sum_{i=1}^m \ell(f(z^{(i)}), y^{(i)})$. Then for any $\delta > 0$, with probability at least $1 - \delta$, simultaneously for all $f \in \mathcal{F}$,*

$$\left|\mathsf{risk}(f; \mathcal{D}) - \widehat{\mathsf{risk}}\left(f; (z^{(i)}, y^{(i)})_{i=1}^m\right)\right| \le 4L\,\widehat{\mathcal{R}}\left(\mathcal{F}; z^{(1)}, \ldots, z^{(m)}\right) + 2b\sqrt{\frac{\log(1/\delta)}{2m}}.$$

In order to bound the Rademacher complexity, we use a modified version of Dudley's metric entropy.

Lemma A.3 (Dudley (1967); modified)**.** *Consider a real-valued function class $\mathcal{F}$ such that $|f| \le A$ for all $f \in \mathcal{F}$. Then*

$$\widehat{\mathcal{R}}(\mathcal{F}; z^{(1)}, \ldots, z^{(m)}) \lesssim \inf_{\delta\ge 0}\left(\delta + \int_\delta^A \sqrt{\frac{\log \mathcal{N}_\infty(\mathcal{F}; \varepsilon; z^{(1)}, \ldots, z^{(m)})}{m}}\, d\varepsilon\right)$$

Proof sketch. The original statement is for 2-norm covering number, but with a few simplifying tweaks the proof applies for ∞-norm covering number as well; alternatively, the latter case reduces to the former, since $N_2(\cdot) \leq N_\infty(\cdot)$. The original statement also fixes $\delta = 0$ rather than taking an infimum. To fix this, in the proof, replace the exponentially decreasing sequence $\{\alpha_j\}_{j=0}^N$ with $\{\delta\} \cup \{\delta + \alpha_j\}_{j=0}^N$. Finally, skipping the last inequality in the proof ensures that the integral goes to A instead of ∞. □

We focus on only proving covering number bounds, using standard machinery as mentioned above, these can be converted to generalization bounds with dependence scaling polynomially with the covering number.

A.2 Useful lemmas

Lemma A.4. *For discrete distributions P, Q, $\|P - Q\|_1 = 2\|P - Q\|_\infty$.*

Proof. Note that the total variation distance δ between two probability distributions P, Q over a finite sample space can be defined in two ways:

$$\delta(P, Q) = \max_i |P_i - Q_i| = \frac{1}{2}\|P - Q\|_1 \implies \|P - Q\|_1 = 2\|P - Q\|_\infty.$$

□

Lemma A.5. *Given θ, θ', for all i, $|\mathsf{softmax}(\theta)_i - \mathsf{softmax}(\theta')_i| \leq \frac{1}{4}|\theta_i - \theta'_i|$.*

Proof. Observe that $\nabla_{w_i}\mathsf{softmax}(w)_i = \mathsf{softmax}(w)_i(1 - \mathsf{softmax}(w)_i)$. By the mean value theorem, for some $\widehat{\theta} = \alpha\theta + (1 - \alpha)\theta'$ with $\alpha \in [0, 1]$,

$$|\mathsf{softmax}(\theta)_i - \mathsf{softmax}(\theta')_i| = \mathsf{softmax}(\widehat{\theta})_i(1 - \mathsf{softmax}(\widehat{\theta})_i)|\theta_i - \theta'_i| \leq \frac{1}{4}|\theta_i - \theta'_i|.$$

□

Lemma A.6. *For $\alpha_i, \beta_i \geq 0$, the solution to the following optimization*

$$\min_{x_1,\ldots,x_n} \sum_{i=1}^n \frac{\alpha_i}{x_i^2}$$
$$\textit{subject to} \sum_{i=1}^n \beta_i x_i = C$$

is $\frac{\gamma^3}{C^2}$ and is achieved at $x_i = \frac{C}{\gamma}\left(\frac{\alpha_i}{\beta_i}\right)^{1/3}$ where $\gamma = \sum_{i=1}^n \alpha_i^{1/3}\beta_i^{\frac{2}{3}}$.

Proof. The proof follows by a standard Lagrangian analysis. □

Lemma A.7 (Contraction of Π_{norm})**.** *Let Π_{norm} be the projection operator onto the unit norm ball. For any vectors u, v, we have $\|\Pi_{norm}(u) - \Pi_{norm}(v)\| \leq \|u - v\|$.*

Proof. If u, v are both in the unit ball then this follows trivially. Let us assume that $\|u\| \geq \|v\|$ and $\|u\| \geq 1$ WLOG. First suppose $\|v\| \leq 1$. Let $B_V^{(1)} = \alpha u$ be the projection of v in the direction of u, and let $B_V^2 = v - B_V^{(1)}$. Then

$$\begin{aligned}
\|\Pi_{norm}(u) - \Pi_{norm}(v)\|^2 &= \|u/\|u\| - v\|^2 \\
&= \|u/\|u\| - (\alpha u + B_V^2)\|^2 \\
&= \|(\|u\|^{-1} - \alpha)u - B_V^2\|^2 \\
&= (\|u\|^{-1} - \alpha)^2\|u\|^2 + \|B_V^2\|^2 \\
&\leq (1 - \alpha^2)\|u\|^2 + \|B_V^2\|^2 && \text{since } \|u\|^{-1} < \alpha < 1 \\
&= \|u - (\alpha u + B_V^2)\|^2 \\
&= \|u - v\|^2
\end{aligned}$$

If $\|v\| > 1$, then

$$\|\Pi_{norm}(u) - \Pi_{norm}(v)\| = \|\Pi_{norm}(u/\|v\|) - \Pi_{norm}(v/\|v\|)\| \leq \|u/\|v\| - v/\|v\|\| < \|u - v\|.$$

where the second-to-last inequality follows from the $\|v\| < 1$ case. □

Lemma A.8 (Zhang (2002) [Theorem 4]). *] Let $\mathcal{V} : \{v : v \in \mathbb{R}^{d_1}, \|v\| \leq B_1\}$ and $\mathcal{F}_{linear} = \{x \mapsto v^T x : v \in \mathcal{V}\}$. For any $\delta > 0$ and $x^{(1)}, \ldots, x^{(N)}$ satisfying $\|x^{(i)}\| \leq B_2$ $\forall i$,*

$$\log \mathcal{N}_\infty(\mathcal{F}_{linear}; \varepsilon; x^{(1)}, \cdots, x^{(N)}) \lesssim \frac{B_1^2 B_2^2 \log((1 + B_1 B_2/\varepsilon)N)}{\varepsilon^2}.$$

A.3 Omitted Proofs

Proof of Lemma 4.3. Observe that,

$$\begin{aligned}
&|f_{\text{head}}(X, z; \theta_s, \theta_{\text{in}}, w) - f_{\text{head}}(X, z; \bar{\theta}_s, \bar{\theta}_{\text{in}}), w)| \\
&= \left|w^\top \left(\phi_{\text{out}}\left(\phi_{\text{in}}(X; \theta_{\text{in}})^\top \mathsf{Norm}(\mathsf{Score}(X, z; \theta_s))\right) - \phi_{\text{out}}\left(\phi_{\text{in}}(X; \bar{\theta}_{\text{in}})^\top \mathsf{Norm}(\mathsf{Score}(X, z; \bar{\theta}_s))\right)\right)\right| \\
&\leq \|w\| \left\|\phi_{\text{out}}\left(\phi_{\text{in}}(X; \theta_{\text{in}})^\top \mathsf{Norm}(\mathsf{Score}(X, z; \theta_s))\right) - \phi_{\text{out}}\left(\phi_{\text{in}}(X; \bar{\theta}_{\text{in}})^\top \mathsf{Norm}(\mathsf{Score}(X, z; \bar{\theta}_s))\right)\right\|
\end{aligned}$$

By L_{out}-Lipschitzness of ϕ_{out} and bound on $\|w\|$:

$$\leq L_{\text{out}} B_w \left\|\phi_{\text{in}}(X; \theta_{\text{in}})^\top \mathsf{Norm}(\mathsf{Score}(X, z; \theta_s)) - \phi_{\text{in}}(X; \bar{\theta}_{\text{in}})^\top \mathsf{Norm}(\mathsf{Score}(X, z; \bar{\theta}_s))\right\|$$

By triangle inequality:

$$\begin{aligned}
&\leq L_{\text{out}} B_w \left\|\phi_{\text{in}}(X; \theta_{\text{in}})^\top \left(\mathsf{Norm}(\mathsf{Score}(X, z; \theta_s)) - \mathsf{Norm}(\mathsf{Score}(X, z; \bar{\theta}_s))\right)\right\| \\
&\quad + L_{\text{out}} B_w \left\|\left(\phi_{\text{in}}(X; \theta_{\text{in}}) - \phi_{\text{in}}(X; \bar{\theta}_{\text{in}})\right)^\top \mathsf{Norm}(\mathsf{Score}(X, z; \bar{\theta}_s))\right\|
\end{aligned}$$

Using $\|A^\top v\| \leq \|A^\top\|_{2,\infty} \|v\|_1$ and B_{in}-boundedness of ϕ_{in}:

$$\begin{aligned}
&\leq L_{\text{out}} B_w B_{\text{in}} \left\|\mathsf{Norm}(\mathsf{Score}(X, z; \theta_s)) - \mathsf{Norm}(\mathsf{Score}(X, z; \bar{\theta}_s))\right\|_1 \\
&\quad + L_{\text{out}} B_w \left\|\left(\phi_{\text{in}}(X; \theta_{\text{in}}) - \phi_{\text{in}}(X; \bar{\theta}_{\text{in}})\right)^\top\right\|_{2,\infty} \left\|\mathsf{Norm}(\mathsf{Score}(X, z; \bar{\theta}_s))\right\|_1
\end{aligned}$$

By Lemma A.4, for discrete distributions P, Q; $\|P\|_1 = \|Q\|_1 = 1$ and $\|P - Q\|_1 = 2\|P - Q\|_\infty$:

$$\begin{aligned}
&\leq 2 L_{\text{out}} B_w \left\|\phi_{\text{in}}(X; \theta_{\text{in}})^\top\right\|_{2,\infty} \left\|\mathsf{Norm}(\mathsf{Score}(X, z; \theta_s)) - \mathsf{Norm}(\mathsf{Score}(X, z; \bar{\theta}_s))\right\|_\infty \\
&\quad + L_{\text{out}} B_w \left\|\left(\phi_{\text{in}}(X; \theta_{\text{in}}) - \phi_{\text{in}}(X; \bar{\theta}_{\text{in}})\right)^\top\right\|_{2,\infty}
\end{aligned}$$

By L_{Norm}-Lipschitzness in the ℓ_∞ sense of Norm:

$$\leq 2 L_{\mathsf{Norm}} L_{\text{out}} B_w B_{\text{in}} \left\|\mathsf{Score}(X, z; \theta_s) - \mathsf{Score}(X, z; \bar{\theta}_s)\right\|_\infty + L_{\text{out}} B_w \left\|\left(\phi_{\text{in}}(X; \theta_{\text{in}}) - \phi_{\text{in}}(X; \bar{\theta}_{\text{in}})\right)^\top\right\|_{2,\infty}$$

□

Proof of Theorem 4.2. Our goal is to show that for every $\varepsilon > 0$, collection of inputs $(X^{(1)}, z^{(1)}), \ldots, (X^{(m)}, z^{(m)})$, and fixed w, there is a cover $\mathcal{C}_{\text{head}}$ such that for all $\theta_s \in \Theta_s, \theta_{\text{in}} \in \Theta_{\text{in}}$, there is some $(\widehat{\theta}_s, \widehat{\theta}_{\text{in}}) \in \mathcal{C}_{\text{head}}$ such that $\max_i |f_{\text{head}}(X^{(i)}, z^{(i)}; \theta_s, \theta_{\text{in}}, w) - f_{\text{head}}(X^{(i)}, z^{(i)}; \widehat{\theta}_s, \widehat{\theta}_{\text{in}})| \leq \varepsilon$.

Observe that for all $\theta_s, \bar{\theta}_s$,

$$\max_{i \in [m]} \|\mathsf{Score}(X^{(i)}, z^{(i)}; \theta_s) - \mathsf{Score}(X^{(i)}, z^{(i)}; \bar{\theta}_s)\|_\infty = \max_{i \in [m], t \in [T]} \left|\mathsf{Score}(x_t^{(i)}, z^{(i)}; \theta_s) - \mathsf{Score}(x_t^{(i)}, z^{(i)}; \bar{\theta}_s)\right|.$$

Similarly, for all $\theta_{\text{in}}, \bar{\theta}_{\text{in}}$,

$$\max_{i \in [m]} \left\|\left(\phi_{\text{in}}(X^{(i)}; \theta_{\text{in}}) - \phi_{\text{in}}(X^{(i)}; \bar{\theta}_{\text{in}})\right)^\top\right\|_{2,\infty} = \max_{i \in [m], t \in [T]} \left\|\phi_{\text{in}}(x_t^{(i)}; \theta_{\text{in}}) - \phi_{\text{in}}(x_t^{(i)}; \bar{\theta}_{\text{in}})\right\|.$$

This crucially allows us to aggregate over the i and t dimensions together.[5] Therefore, we can consider $\mathcal{N}_\infty$ covers for the above to bound the overall covering number.

Let $\mathcal{C}_{\mathsf{Score}}$ be the $\varepsilon_{\mathsf{Score}}$-cover ($\infty$) for $\mathcal{F}_{\mathsf{Score}}$ over inputs $\left\{(x_t^{(i)}, z^{(i)})\right\}_{i\in[m],t\in[T]}$ of size $\mathcal{N}_\infty\left(\mathcal{F}_{\mathsf{Score}}; \varepsilon_{\mathsf{Score}}; \{(x_t^{(i)}, z^{(i)})\}_{i\in[m],t\in[T]}\right)$. Also, Let $\mathcal{C}_{\mathrm{in}}$ be the $\varepsilon_{\mathrm{in}}$-cover ($\infty$) for $\mathcal{F}_{\mathrm{in}}$ over inputs $\{x_t^{(i)}\}_{i\in[m],t\in[T]}$ of size $\mathcal{N}_\infty\left(\mathcal{F}_{\mathrm{in}}; \varepsilon_{\mathrm{in}}; \{x_t^{(i)}\}_{i\in[m],t\in[T]}; \|\cdot\|_2\right)$.

We are ready to construct the cover for $\mathcal{F}_{\mathrm{head}}$. Set $\mathcal{C}_{\mathrm{head}} = \{f_{\mathrm{head}}(\cdot; \widehat{\theta}_s, \widehat{\theta}_{\mathrm{in}}, w))_{i\in[m]} : \widehat{\theta}_s \in \mathcal{C}_{\mathsf{Score}}, \widehat{\theta}_{\mathrm{in}} \in \mathcal{C}_{\mathrm{in}}\}$. Then for any $\theta_s \in \Theta_s, \theta_{\mathrm{in}} \in \Theta_{\mathrm{in}}$, there exists $\widehat{\theta}_s, \widehat{\theta}_{\mathrm{in}} \in \mathcal{C}_{\mathrm{head}}$, such that for all $i \in [m]$

$$|f_{\mathrm{head}}(X^{(i)}, z^{(i)}; \theta_s, \theta_{\mathrm{in}}, w) - f_{\mathrm{head}}(X^{(i)}, z^{(i)}; \widehat{\theta}_s, \widehat{\theta}_{\mathrm{in}}, w)|$$

Using Lemma 4.3:

$$\leq 2L_{\mathsf{Norm}} L_{\mathrm{out}} B_w B_{\mathrm{in}} \varepsilon_{\mathsf{Score}} + L_{\mathrm{out}} B_w \varepsilon_{\mathrm{in}}.$$

The size of the cover we have constructed is,

$$\begin{aligned}\log|\mathcal{C}_{\mathrm{head}}| &= \log|\mathcal{C}_{\mathsf{Score}}| + \log|\mathcal{C}_{\mathrm{in}}| \\ &= \log\mathcal{N}_\infty\left(\mathcal{F}_{\mathsf{Score}}; \varepsilon_{\mathsf{Score}}; \{(x_t^{(i)}, z^{(i)})\}_{i\in[m],t\in[T]}\right) + \log\mathcal{N}_\infty\left(\mathcal{F}_{\mathrm{in}}; \varepsilon_{\mathrm{in}}; \{x_t^{(i)}\}_{i\in[m],t\in[T]}; \|\cdot\|_2\right)\end{aligned}$$

and we are done. □

Proof of Lemma 4.4. By Lemma 4.2, the covering number of $\mathcal{F}_{\mathsf{tf\text{-}head}}$ satisfies

$$\begin{aligned}&\log\mathcal{N}_\infty\left(\mathcal{F}_{\mathsf{tf\text{-}head}}; \varepsilon; \left\{(X^{(i)}, z^{(i)})\right\}_{i=1}^m\right) \\ &\leq \inf_{\alpha\in[0,1]}\Bigg[\log\mathcal{N}_\infty\left(\mathcal{F}_{QK}; \frac{2\alpha\varepsilon}{L_\sigma B_w B_V}; \{(x_t^{(i)}, z^{(i)})\}_{i\in[m],t\in[T]}\right) \\ &\quad + \log\mathcal{N}_\infty\left(\mathcal{F}_V; \frac{(1-\alpha)\varepsilon}{L_\sigma B_w}; \{x_t^{(i)}\}_{i\in[m],t\in[T]}; \|\cdot\|_2\right)\Bigg].\end{aligned}$$

where we have used the fact that for a scalar-output Transformer layer:

- L_{Norm} is the Lipschitz constant of $\mathsf{softmax}$: $1/4$.
- L_{out} is the Lipschitz constant of σ: L_σ.
- B_{in} is a bound on the norm of $W_V^\top x$: B^V.

By Lemma 4.5, for any $\varepsilon_{QK}, \varepsilon_V > 0$:

$$\log\mathcal{N}_\infty\left(\mathcal{F}_{QK}; \varepsilon_{QK}; \{(x_t^{(i)}, z^{(i)})\}_{i\in[m],t\in[T]}\right) \lesssim \frac{(dB_{2,\mathrm{inf}}^{QK})^2 \log(mT)}{\varepsilon_{QK}^2}$$

$$\log\mathcal{N}_\infty\left(\mathcal{F}_V; \varepsilon_V; \{(x_t^{(i)}, z^{(i)})\}_{i\in[m],t\in[T]}; \|\cdot\|_2\right) \lesssim \frac{(k(B_{2,\mathrm{inf}}^V)^2 \log(mT)}{\varepsilon_V^2}$$

since $W_{QK} \in \mathbb{R}^{d\times d}, W_V \in \mathbb{R}^{d\times k}$. We want to choose ε_{QK} and ε_V to minimize the sum of the above two terms, subject to

$$\frac{1}{2} L_\sigma B_w B^V \varepsilon_{QK} + L_\sigma B_w \varepsilon_V \leq \varepsilon.$$

[5] In the case of the Transformer self-attention mechanism, we will obtain ∞-norm covering numbers for Score and ϕ_{in} that have only logarithmic dependence on the number of examples. Because of this aggregation trick, the resulting covering number for the whole layer will have merely logarithmic dependence on the context length T.

By Lemma A.6, the solution to this optimization leads to an optimal bound of:

$$\log \mathcal{N}_\infty(\mathcal{F}_{\text{tf-head}}; \varepsilon; X^{(1)}, \ldots, X^{(M)}) \lesssim \frac{\left((kB_{2,\infty}^V L_\sigma B_w)^{\frac{2}{3}} + (\frac{1}{2} d B_{2,\infty}^{QK} L_\sigma B_w B^V)^{\frac{2}{3}}\right)^3 \log(mT)}{\varepsilon^2}.$$

□

Proof of Lemma 4.5. Let B_∞ be an upper bound on $\|W\|_{2,\infty}$ and B_1 be an upper bound on $\|W\|_{2,1}$. (In the worst case $B_1 \leq d_2 B_\infty$.)

The approach will be to cover each of the columns of W independently, treating each as specifying a linear function from $\mathbb{R}^{d_1} \to \mathbb{R}$.

By Lemma A.8, letting $\mathcal{V}(b) : \{v : v \in \mathbb{R}^{d_1}, \|v\| \leq b\}$ and $\mathcal{F}_{\text{linear}}(b) = \{x \mapsto v^T x : v \in \mathcal{V}(b)\}$, for any $\delta > 0$

$$\log \mathcal{N}_\infty(\mathcal{F}_{\text{linear}}(b); \delta; x^{(1)}, \cdots, x^{(N)}) \leq \frac{cb^2 \log((1 + b/\delta)N)}{\delta^2}.$$

given that $\|x^{(i)}\| \leq 1$ for all i.

In fact the cover, which we denote by $\widehat{\mathcal{F}}_{\text{linear}}(b; \delta)$, is proper: $\widehat{\mathcal{F}}_{\text{linear}}(b; \delta) = \{x \mapsto \widehat{v}^T x : \widehat{v} \in \widehat{V}\}$ for some finite subset $\widehat{V} \subset \mathcal{V}(b)$.

Let

$$\mathcal{S} = \{(k_1, k_2, \ldots, k_{d_2}) : k_i \in \{0, 1, \ldots, d_2\} \text{ for all } i \text{ and } 0 \leq \sum_{i=1}^{d_2} k_i \leq d_2\}$$

Given any $W \in \mathcal{W}$, let v_i denote the ith column of W. For each i, let

$$k_i^* = \left\lfloor \frac{d_2}{B_\infty} \|v_i\| \right\rfloor$$

Let $S_W = (k_1^*, \ldots, k_{d_2}^*)$. Then $S_W \in \mathcal{S}$, and

$$\frac{B_\infty}{d_2} k_i^* \leq \|v_i\| \leq \frac{B_\infty}{d_2}(k_i^* + 1) \quad \forall i.$$

For every tuple $S = (k_1, k_2, \ldots, k_{d_2}) \in \mathcal{S}$, let

$$\widehat{\mathcal{W}}_S = \left\{ [\widehat{v}_1 \widehat{v}_2 \ldots \widehat{v}_{d_2}]^T : \widehat{v}_i \in \widehat{\mathcal{F}}_{\text{linear}}\left(B_\infty(k_i + 1)/d_2; \varepsilon\sqrt{(k_i + 1)/(2d_2)}\right) \text{ for all } i \right\}.$$

Our cover will be $\widehat{\mathcal{F}} : \{x \mapsto \widehat{W} x : \widehat{W} \in \bigcup_{S \in \mathcal{S}} \widehat{\mathcal{W}}_S\}$. Note that

$$\begin{aligned}
|\widehat{\mathcal{F}}| &\leq \sum_{S \in \mathcal{S}} \prod_{i=1}^{d_2} \mathcal{N}_\infty\left(\mathcal{F}_{\text{linear}}(B_\infty(k_i + 1)/d_2); \varepsilon\sqrt{(k_i + 1)/(2d_2)}; x^{(1)}, \cdots, x^{(N)}\right) \\
&\lesssim \sum_{S \in \mathcal{S}} \exp\left(\sum_{i=1}^{d_2} \frac{B\infty^2 (k_i + 1)^2 \log(N)}{d_2^2 \varepsilon^2 (k_i + 1)/(2d_2)}\right) \\
&= \sum_{S \in \mathcal{S}} \exp\left(\frac{2B_\infty^2 \log(N)}{d_2 \varepsilon^2} \sum_{i=1}^{d_2} (k_i + 1)\right) \\
&\leq |\mathcal{S}| \exp\left(\frac{4B_1 B_\infty \log(N)}{\varepsilon^2}\right)
\end{aligned}$$

where in the last step we used the fact that

$$\sum_{i=1}^{d_2} k_i \leq \frac{d_2}{B_\infty} \sum_{i=1}^{d} \|v_i\| = d_2 B_1 / B_\infty$$

Since $|\mathcal{S}| \leq \binom{2d_2}{d_2} = \exp(O(d_2 \log d_2))$, we obtain

$$\log |\widehat{\mathcal{F}}| \lesssim \frac{d_2 B_1 B_\infty \log(N)}{\varepsilon^2}$$

as desired. In particular, we obtain the more concise, but looser, bound from the theorem statement by using the fact that $B_1 \leq d_2 B_\infty$

For a particular $W = [v_1 \dots v_{d_2}] \in \mathcal{W}$ it is guaranteed that there is a matrix $\widehat{W} \in \widehat{\mathcal{W}}_{S_W}$, $\widehat{W} = [\widehat{v}_1 \dots \widehat{v}_{d_2}]$, such that for each $i \in [d_2]$,

$$|v_i^\top x_n - \widehat{v}_i^\top x_n| \leq \varepsilon \sqrt{\frac{k_i + 1}{2d_2}} \quad \forall n \in [N].$$

where k_i is the ith element of S_W. We then obtain the desired covering property:

$$\begin{aligned}
\max_{n \in [N]} \left\| W x_n - \widehat{W} x_n \right\| &= \max_{n \in [N]} \sqrt{\sum_{i=1}^{d_2} (v_i^\top x_n - \widehat{v}_i^\top x_n)^2} \\
&\leq \sqrt{\sum_{i=1}^{d_2} \varepsilon^2 \left(\frac{k_i + 1}{2d_2} \right)} \\
&= \varepsilon \sqrt{\frac{d_2 + \sum_{i=1}^{d_2} k_i}{2d_2}} \quad \leq \varepsilon
\end{aligned}$$

□

A.4 Positional Embeddings

Since the transformer architecture is permutation invariant for all $t \neq \tau$, often we add positional embeddings (fixed or trainable) to distinguish the different positions of the tokens. These positional embeddings are matrices $P \in \mathbb{R}^{T \times d}$ such that $P = [p_1 \dots p_T]^\top$ for $p_i \in \mathbb{R}^d$. Accounting for the positional embeddings as input, the single attention head can be expressed as:

$$f_{\mathsf{tf-pos}}(X, P; W_V, W_{QK}, w) := w^\top \sigma \left(W_V^\top (X + P)^\top \mathsf{softmax} \left((X + P) W_{QK}^\top (x_\tau + p_\tau) \right) \right).$$

Lemma A.9. *For any $W_K, \bar{W}_K, W_V, \bar{W}_V, W_Q, \bar{W}_Q \in \mathbb{R}^{d \times k}$, for all $X \in \mathbb{R}^{T \times d}$ such that $\|X^\top\|_{2,\infty} \leq 1$ and $P \in \mathbb{R}^{T \times d}$,*

$$\begin{aligned}
&|f_{\mathsf{tf-pos}}(X, P; W_Q, W_K, W_V, w) - f_{\mathsf{tf-pos}}(X, P; \bar{W}_Q, \bar{W}_K, \bar{W}_V, w)| \\
&\leq \frac{1}{2} L_\sigma \|w\| \, \|W_V\|_2 \, (1 + \left\|P^\top\right\|_{2,\infty})^2 \left\| \left(W_Q W_K^\top - \bar{W}_Q \bar{W}_K^\top \right) (X + P)^\top \right\|_{2,\infty} \\
&\quad + L_\sigma \|w\| \left\| (W_V - \bar{W}_V)^\top (X + P)^\top \right\|_{2,\infty}.
\end{aligned}$$

Proof. The proof follows along the same lines as Lemma 4.3, with the only modification of $X \to X + P$. Since the $\left\|(X + P)^\top\right\|_{2,\infty}$ is no longer bounded by 1, we need to account for a dependence on $\left\|P^\top\right\|_{2,\infty}$. □

Therefore our bounds go through for fixed positional embeddings. If we were to train the embeddings, we would need a much finer cover on the embeddings which could incur a T dependence.

A.5 General deep Transformers

We will consider an L-layer transformer. Let us denote the weights of layer i by $W^{(i)} := \left\{ W_Q^{(i)}, W_K^{(i)}, W_V^{(i)}, W_C^{(i)} \right\}$ such that $\left\| W_K^{(i)} {W_Q^{(i)}}^\top \right\|_2 \leq B_{QK}^{(i)}$, $\left\| W_V^{(i)} \right\|_2 \leq B_V^{(i)}$, $\left\| W_C^{(i)} \right\|_2 \leq B_C^{(i)}$ and $\left\| W_K^{(i)} {W_Q^{(i)}}^\top \right\|_{2,\infty} \leq B_{QK}^{\infty\,(i)}$, $\left\| W_V^{(i)} \right\|_{2,\infty} \leq B_V^{\infty\,(i)}$ and $\left\| W_C^{(i)} \right\|_{2,\infty} \leq B_C^{\infty\,(i)}$. Let us further

denote the set of weights up to layer i by $W^{1:i} = (W^{(1)}, \ldots, W^{i-1})$. Let the input representation of layer i be $g(X; W^{1:i})$. We inductively define g with $g(X; W^{1:1}) = X$

$$g\left(X; W^{1:i+1}\right) = \Pi_{norm}\left(\sigma\left(\Pi_{norm}\left(f\left(g\left(X; W^{1:i}\right); W^{(i)}\right)\right)\right) W_C^{(i)}\right) \text{ with}$$
$$f\left(Z; \{W_Q, W_K, W_V, W_C\}\right) = \mathsf{RowSoftmax}\left(ZW_Q\left(ZW_K\right)^\top\right) ZW_V,$$

where Π_{norm} is applied row-wise. Our final output is $g_{out}(X; W^{1:L+1}, w) = w^\top g\left(X; W^{1:i+1}\right)$ `[CLS]` for $\|w\| \leq B_w$.

In order to construct a cover, we will first bound the distance between the function g with different weight parameters $W^{1:L+1}$ and $\widehat{W}^{1:L+1}$. This bound will depend on the closeness of the parameters which will allow us to construct a cover of the network in an iterative fashion by constructing covers of each layer.

A.5.1 Lipschitzness of the Network

To bound the Lipschitzness of the network, we will first bound the distance between f with different weights and inputs.

Lemma A.10 (Instantiation of Lemma 4.3)**.** *For any $W_K, \widehat{W}_K, W_V, \widehat{W}_V, W_Q, \widehat{W}_Q \in \mathbb{R}^{d\times k}$, for all $Z \in \mathbb{R}^{T\times d}$ such that $\|Z^\top\|_{2,\infty} \leq 1$,*

$$\left\|\left(f\left(Z; \{W_Q, W_K, W_V, \cdot\}\right) - f\left(Z; \{\widehat{W}_Q, \widehat{W}_K, \widehat{W}_V, \cdot\}\right)\right)^\top\right\|_{2,\infty}$$
$$\leq \frac{1}{2}\|W_V\|_2 \left\|\left(W_Q W_K^\top - \widehat{W}_W \widehat{W}_K^\top\right) Z^\top\right\|_{2,\infty} + \left\|(W_V - \widehat{W}_V)^\top Z^\top\right\|_{2,\infty}$$

Proof. Consider a fixed row τ of the output of the functions,

$$\|f\left(Z; \{W_Q, W_K, W_V, \cdot\}\right)[\tau] - f\left(Z; \{W_Q, W_K, W_V, \cdot\}\right)[\tau]\|$$
$$= \left\|W_V^\top Z^\top \mathsf{softmax}\left(ZW_K W_Q^\top z_\tau\right) - \widehat{W}_V^\top Z^\top \mathsf{softmax}\left(Z\widehat{W}_K \widehat{W}_Q^\top z_\tau\right)\right\|$$

By triangle inequality:

$$\leq \left\|W_V^\top Z^\top \left(\mathsf{softmax}\left(ZW_K W_Q^\top z_\tau\right) - \mathsf{softmax}\left(Z\widehat{W}_K \widehat{W}_Q^\top z_\tau\right)\right)\right\|$$
$$+ \left\|(W_V - \widehat{W}_V)^\top Z^\top \mathsf{softmax}\left(Z\widehat{W}_K \widehat{W}_Q^\top z_\tau\right)\right\|$$

Using $\|P^\top v\| \leq \|P^\top|_{2,\infty}\|v\|_1$:

$$\leq \left\|W_V^\top Z^\top\right\|_{2,\infty} \left\|\mathsf{softmax}\left(ZW_K W_Q^\top z_\tau\right) - \mathsf{softmax}\left(Z\widehat{W}_K \widehat{W}_Q^\top z_\tau\right)\right\|_1$$
$$+ \left\|(W_V - \widehat{W}_V)^\top Z^\top\right\|_{2,\infty} \left\|\mathsf{softmax}\left(Z\widehat{W}_K \widehat{W}_Q^\top z_\tau\right)\right\|_1$$

By Lemma A.4 and $\|Z^\top\|_{2,\infty} \leq 1$:

$$\leq 2\|W_V\|_2 \left\|\mathsf{softmax}\left(ZW_K W_Q^\top z_\tau\right) - \mathsf{softmax}\left(Z\widehat{W}_K \widehat{W}_Q^\top z_\tau\right)\right\|_\infty + \left\|(W_V - \widehat{W}_V)^\top Z^\top\right\|_{2,\infty}$$

By $\frac{1}{4}$-Lipschitzness of each entry of softmax:

$$\leq \frac{1}{2}\|W_V\|_2 \left\|Z\left(W_K W_Q^\top - Z\widehat{W}_K \widehat{W}_Q^\top\right) z_\tau\right\|_\infty + L_\sigma\|w\| \left\|(W_V - \widehat{W}_V)^\top Z^\top\right\|_{2,\infty}$$

Since $\|z_\tau\| \leq 1$:

$$\leq \frac{1}{2}\|W_V\|_2 \left\|\left(W_Q W_K^\top - \widehat{W}_Q \widehat{W}_K^\top\right)(X+P)^\top\right\|_{2,\infty} + \left\|(W_V - \widehat{W}_V)^\top Z^\top\right\|_{2,\infty}.$$

□

Lemma A.11. *For any $W_K, W_V, W_Q \in \mathbb{R}^{d\times k}$, for all $Z, \widehat{Z} \in \mathbb{R}^{T\times d}$ such that $\|Z^\top\|_{2,\infty} \leq 1, \|\widehat{Z}^\top\|_{2,\infty} \leq 1$,*

$$\left\|\left(f\left(Z;\{W_Q,W_K,W_V,\cdot\}\right) - f\left(\widehat{Z};\{W_Q,W_K,W_V,\cdot\}\right)\right)^\top\right\|_{2,\infty}$$
$$\leq \|W_V\|_2\left(1+\left\|W_K W_Q^\top\right\|_2\right)\left\|(Z-\widehat{Z})^\top\right\|_{2,\infty}.$$

Proof. Consider a fixed row τ of the output of the functions,

$$\left\|f\left(Z;\{W_Q,W_K,W_V,\cdot\}\right)[\tau] - f\left(\widehat{Z};\{W_Q,W_K,W_V,\cdot\}\right)[\tau]\right\|$$
$$= \left\|W_V^\top Z^\top \mathsf{softmax}\left(ZW_KW_Q^\top z_\tau\right) - W_V^\top \widehat{Z}^\top \mathsf{softmax}\left(\widehat{Z}W_KW_Q^\top \widehat{z}_\tau\right)\right\|$$

By triangle inequality:

$$\leq \left\|W_V^\top\left(Z-\widehat{Z}\right)^\top \mathsf{softmax}\left(ZW_KW_Q^\top z_\tau\right)\right\| + \left\|W_V^\top\widehat{Z}^\top\left(\mathsf{softmax}\left(ZW_KW_Q^\top z_\tau\right) - \mathsf{softmax}\left(\widehat{Z}W_KW_Q^\top\widehat{z}_\tau\right)\right)\right\|$$

Using $\|P^\top v\| \leq \|P^\top\|_{2,\infty}\|v\|_1$:

$$\leq \left\|W_V^\top\left(Z-\widehat{Z}\right)\right\|_{2,\infty}\left\|\mathsf{softmax}\left(ZW_KW_Q^\top z_\tau\right)\right\|_1$$
$$+ \left\|W_V^\top\widehat{Z}^\top\right\|_{2,\infty}\left\|\mathsf{softmax}\left(ZW_KW_Q^\top z_\tau\right) - \mathsf{softmax}\left(\widehat{Z}W_KW_Q^\top\widehat{z}_\tau\right)\right\|_1$$

By Lemma A.4, $\|Z^\top\|_{2,\infty} \leq 1$ and $\|PQ\|_{2,\infty} \leq \|P^\top\|_{2,\infty}\|Q\|_2$:

$$\leq \|W_V\|_2\left\|(Z-\widehat{Z})^\top\right\|_{2,\infty} + 2\|W_V\|_2\left\|\mathsf{softmax}\left(ZW_KW_Q^\top z_\tau\right) - \mathsf{softmax}\left(\widehat{Z}W_KW_Q^\top\widehat{z}_\tau\right)\right\|_\infty$$

By 1/4-Lipschitzness of each entry of softmax:

$$\leq \|W_V\|_2\left\|(Z-\widehat{Z})^\top\right\|_{2,\infty} + \frac{1}{2}\|W_V\|_2\left\|ZW_KW_Q^\top z_\tau - \widehat{Z}W_KW_Q^\top\widehat{z}_\tau\right\|_\infty$$

By triangle inequality:

$$\leq \|W_V\|_2\left\|(Z-\widehat{Z})^\top\right\|_{2,\infty} + \frac{1}{2}\|W_V\|_2\left(\left\|(Z-\widehat{Z})W_KW_Q^\top z_\tau\right\|_\infty + \left\|\widehat{Z}W_KW_Q^\top\left(z_\tau-\widehat{z}_\tau\right)\right\|_\infty\right)$$

Since $\|Z^\top\|_{2,\infty} \leq 1$:

$$\leq \|W_V\|_2\left(1+\left\|W_KW_Q^\top\right\|_2\right)\left\|(Z-\widehat{Z})^\top\right\|_{2,\infty}.$$

□

With the above lemmas, we are ready to prove the effect of change of weights on g.

Lemma A.12. *For any $W_1^{i+1}, \widehat{W}_1^{i+1}$ satisfying the norm constraints,*

$$\left\|\left(g^{W^{1:i+1}}(X) - g^{\widehat{W}^{1:i+1}}(X)\right)^\top\right\|_{2,\infty} \leq \left\|\left(W_C^{(i)} - \widehat{W}_C^{(i)}\right)^\top \sigma\left(\Pi_{norm}\left(f\left(g\left(X;\widehat{W}^{1:i}\right);\widehat{W}^{(i)}\right)\right)\right)^\top\right\|_{2,\infty}$$
$$+ L_\sigma B_C^{(i)} B_V^{(i)}\left(1+B_{QK}^{(i)}\right)\left\|\left(g\left(X;W^{1:i}\right) - g\left(X;\widehat{W}^{1:i}\right)\right)^\top\right\|_{2,\infty}$$
$$+ \frac{L_\sigma}{2}B_C^{(i)}B_V^{(i)}\left\|\left(W_Q^{(i)}W_K^{(i)\top} - \widehat{W}_Q^{(i)}\widehat{W}_K^{(i)\top}\right)g\left(X;\widehat{W}^{1:i}\right)^\top\right\|_{2,\infty}$$
$$+ L_\sigma B_C^{(i)}\left\|(W_V - \widehat{W}_V)^\top g\left(X;\widehat{W}^{1:i}\right)^\top\right\|_{2,\infty}.$$

Proof. Unrolling one layer, we have

$$\left\| \left(g\left(X; W^{1:i+1}\right) - g\left(X; \widehat{W}^{1:i+1}\right)\right)^\top \right\|_{2,\infty}$$
$$= \left\| \left(\Pi_{norm}\left(\sigma\left(\Pi_{norm}\left(f\left(g\left(X; W^{1:i}\right); W^{(i)}\right)\right)\right) W_C^{(i)}\right) - \Pi_{norm}\left(\sigma\left(\Pi_{norm}\left(f\left(g\left(X; \widehat{W}^{1:i}\right); \widehat{W}^{(i)}\right)\right)\right) \widehat{W}_C^{(i)}\right)\right)^\top \right\|_{2,}$$

Using Lemma A.7 for each row:

$$\leq \left\| {W_C^{(i)}}^\top \sigma\left(\Pi_{norm}\left(f\left(g\left(X; W^{1:i}\right); W^{(i)}\right)\right)\right)^\top - {\widehat{W}_C^{(i)}}^\top \sigma\left(\Pi_{norm}\left(f\left(g\left(X; \widehat{W}^{1:i}\right); \widehat{W}^{(i)}\right)\right)\right) \right\|_{2,\infty}$$

By triangle inequality for each row:

$$\leq \underbrace{\left\| {W_C^{(i)}}^\top \left(\sigma\left(\Pi_{norm}\left(f\left(g\left(X; W^{1:i}\right); W^{(i)}\right)\right)\right) - \sigma\left(\Pi_{norm}\left(f\left(g\left(X; \widehat{W}^{1:i}\right); \widehat{W}^{(i)}\right)\right)\right)\right)^\top \right\|_{2,\infty}}_{(A)}$$
$$+ \left\| \left(W_C^{(i)} - \widehat{W}_C^{(i)}\right)^\top \sigma\left(\Pi_{norm}\left(f\left(g\left(X; \widehat{W}^{1:i}\right); \widehat{W}^{(i)}\right)\right)\right)^\top \right\|_{2,\infty}.$$

Let us focus on term (A).

Bounding the norm per row:

$$(A) \leq \left\| W_C^{(i)} \right\|_2 \left\| \sigma\left(\Pi_{norm}\left(f\left(g\left(X; W^{1:i}\right); W^{(i)}\right)\right)\right)^\top - \sigma\left(\Pi_{norm}\left(f\left(g\left(X; \widehat{W}^{1:i}\right); \widehat{W}^{(i)}\right)\right)\right)^\top \right\|_{2,\infty}$$

Since σ is L_σ-Lipschitz and $\left\| W_C^{(i)} \right\|_2 \leq B_C^{(i)}$, for each row:

$$\leq L_\sigma B_C^{(i)} \left\| \Pi_{norm}\left(f\left(g\left(X; W^{1:i}\right); W^{(i)}\right)\right)^\top - \Pi_{norm}\left(f\left(g\left(X; \widehat{W}^{1:i}\right); \widehat{W}^{(i)}\right)\right)^\top \right\|_{2,\infty}$$

Using Lemma A.7 for each row:

$$\leq L_\sigma B_C^{(i)} \left\| f\left(g\left(X; W^{1:i}\right); W^{(i)}\right)^\top - f\left(g\left(X; \widehat{W}^{1:i}\right); \widehat{W}^{(i)}\right)^\top \right\|_{2,\infty}$$

By triangle inequality:

$$\leq L_\sigma B_C^{(i)} \left\| f\left(g\left(X; W^{1:i}\right); W^{(i)}\right)^\top - f\left(g\left(X; \widehat{W}^{1:i}\right); W^{(i)}\right)^\top \right\|_{2,\infty}$$
$$+ L_\sigma B_C^{(i)} \left\| f\left(g\left(X; \widehat{W}^{1:i}\right); W^{(i)}\right)^\top - f\left(g\left(X; \widehat{W}^{1:i}\right); \widehat{W}^{(i)}\right)^\top \right\|_{2,\infty}$$

By Lemma A.10 and A.11 and norm bounds on the matrices:

$$\leq L_\sigma B_C^{(i)} B_V^{(i)} \left(1 + B_{QK}^{(i)}\right) \left\| g\left(X; W^{1:i}\right)^\top - g\left(X; \widehat{W}^{1:i}\right)^\top \right\|_{2,\infty}$$
$$+ \frac{L_\sigma}{2} B_C^{(i)} B_V^{(i)} \left\| \left(W_Q^{(i)} {W_K^{(i)}}^\top - \widehat{W}_Q^{(i)} {\widehat{W}_K^{(i)}}^\top g\left(X; \widehat{W}^{1:i}\right)^\top \right) \right\|_{2,\infty} + L_\sigma B_C^{(i)} \left\| (W_V - \widehat{W}_V)^\top g\left(X; \widehat{W}^{1:i}\right)^\top \right\|_{2,\infty}.$$

Combining the above gives us the desired result. □

Lastly, we take account of the last linear weight and observe that,

Lemma A.13. *For any $W^{1:L+1}, \widehat{W}^{1:L+1}$ and $w, \widehat{w}$,*

$$\left| g_{out}\left(X; W^{1:L+1}, w\right) - g_{out}\left(X; \widehat{W}^{1:L+1}, \widehat{w}\right) \right|$$
$$\leq \|w\| \left\| g\left(X; W^{1:L+1}\right) \texttt{[CLS]} - g\left(X; \widehat{W}^{1:L+1}\right) \texttt{[CLS]} \right\| + \left| (w - \widehat{w})^\top g\left(X; \widehat{W}^{1:L+1}\right) \texttt{[CLS]} \right|.$$

Proof. Observe that,

$$\left|g_{out}\left(X;W^{1:L+1},w\right)-g_{out}\left(X;\widehat{W}^{1:L+1},\widehat{w}\right)\right|$$
$$=\left|w^\top g\left(X;W^{1:L+1}\right)\texttt{[CLS]}-\widehat{w}^\top g\left(X;\widehat{W}^{1:L+1}\right)\texttt{[CLS]}\right|$$

By triangle inequality:

$$\leq\left|w^\top\left(g\left(X;W^{1:L+1}\right)\texttt{[CLS]}-g\left(X;\widehat{W}^{1:L+1}\right)\texttt{[CLS]}\right)\right|+\left|(w-\widehat{w})^\top g\left(X;\widehat{W}^{1:L+1}\right)\texttt{[CLS]}\right|$$

Bounding the inner product by norms:

$$\leq\|w\|\left\|g\left(X;W^{1:L+1}\right)\texttt{[CLS]}-g\left(X;\widehat{W}^{1:L+1}\right)\texttt{[CLS]}\right\|+\left|(w-\widehat{w})^\top g\left(X;\widehat{W}^{1:L+1}\right)\texttt{[CLS]}\right|.$$

□

A.5.2 Constructing the Cover

Theorem A.14. *Let $\mathcal{F}^{(L)}$ represent the class of functions of L-layer transformer network satisfying the norm bounds (specified before). Then, we have*

$$\log\mathcal{N}_\infty(\mathcal{F}^{(L)};\varepsilon;X^{(1)},\ldots,X^{(m)})\lesssim\quad\frac{\log(mT)}{\varepsilon^2}\times$$
$$\left((B_WL_\sigma B_w)^{\frac{2}{3}}+\sum_{i=1}^{L}\alpha_i^{\frac{2}{3}}\left(d^{\frac{2}{3}}B_C^{\infty(i)\frac{2}{3}}+d^{\frac{2}{3}}\left(L_\sigma B_C^{(i)}B_V^{(i)}B_{QK}^{\infty(i)}\right)^{\frac{2}{3}}+k^{\frac{2}{3}}\left(L_\sigma B_C^{(i)}B_V^{\infty(i)}\right)^{\frac{2}{3}}\right)\right)^3$$

where $\alpha_i=\prod_{j<i}L_\sigma B_C^{(j)}B_V^{(j)}(1+B_{QK}^{(j)})$.

Proof. Our goal is to show that for every $\varepsilon>0$, collection of inputs $X^{(1)},\ldots,X^{(m)}$, there is a cover $\mathcal{C}$ of vectors in $\mathbb{R}^{(m)}$ such that for all $W^{1:L+1},w$ satisfying the norm bounds, there is some $v\in\mathcal{C}$ such that $\max_i|g_{out}(X^{(i)};W^{1:L+1},w)-B_V^{(i)}|\leq\varepsilon$.

In each layer of the transformer, $W_Q^{(i)}$ and $W_K^{(i)}$ always appear together in the form $W_K^{(i)}{W_Q^{(i)}}^\top$. Therefore, we will overload notation and define $W^{(i)}=\{W_V^{(i)},W_{QK}^{(i)},W_C^{(i)}\}$ as the function computed by an attention head with $W_K^{(i)}{W_Q^{(i)}}^\top$ replaced by $W_{QK}^{(i)}$. Our cover $\mathcal{C}$ will be proper and consist of vectors of the form $(g_{out}(X^{(i)};W^{1:L+1},w))_{i\in[m]}$. We will build the cover iteratively by finding finite collections of set of matrices $\widehat{\mathcal{W}}^{1:i}$ for each layer.

First observe that for any collection of $Z^{(1)},\ldots,Z^{(m)}\in\mathbb{R}^{T\times d_1}$, and any $W,\widehat{W}\in\mathbb{R}^{d_1\times d_2}$,

$$\max_{i\in[m]}\left\|W^\top{Z^{(i)}}^\top-\widehat{W}^\top{Z^{(i)}}^\top\right\|_{2,\infty}=\max_{i\in[m],t\in[T]}\left\|W^\top z_t^{(i)}-\widehat{W}^\top z_t^{(i)}\right\|.$$

This crucially allows us to aggregate over the samples and context length. In particular, we can apply Lemma 4.5 with the input vectors $(z_t^{(i)})_{i\in[m],t\in[T]}$; a total of mT input vectors. Specifically, for any ε and $\mathcal{W}(d_1,d_2,\alpha):=\{W\in\mathbb{R}^{d_1\times d_2}\mid\|W^\top\|_{2,\infty}\leq\alpha\}$ with fixed $Z^{(i)}$ satisfying $\left\|{Z^{(i)}}^\top\right\|_{2,\infty}\leq 1$, Lemma 4.5 gives us such a cover.

First let us build a cover for one transformer layer with input $Z^{(1)},\ldots,Z^{(m)}$. We will begin with creating an ε_V-cover $\widehat{\mathcal{W}}_V$ for $\mathcal{W}_V:\{W\in\mathbb{R}^{d\times k},\left\|W^\top\right\|_{2,\infty}\leq\alpha,\|W\|_2\leq s\}$ and ε_{QK}-cover $\widehat{\mathcal{W}}_{QK}$ for $\mathcal{W}_{QK}:=\{W\in\mathbb{R}^{d\times d},\|W\|_\infty\leq\beta,\|W\|_2\leq r\}$ and inputs $Z^{(1)},\ldots,Z^{(m)}$. For each pair of $\widehat{W}_V\in\widehat{\mathcal{W}}_V$ and $\widehat{W}_{QK}\in\widehat{\mathcal{W}}_{QK}$, we construct ε_C-cover $\widehat{\mathcal{W}}_C(\widehat{W}_V,\widehat{W}_{QK})$ for $\mathcal{W}_C:\{W\in\mathbb{R}^{k\times d},\|W\|_{2,\infty}\leq\gamma,\|W\|_2\leq c\}$ and inputs $\left\{\sigma\left(\Pi_{norm}\left(f\left(Z^{(i)};\widehat{W}_V,\widehat{W}_{QK}\right)\right)\right)\right\}_{i=1}^{(m)}$. Our final cover is $\widehat{\mathcal{W}}:=\left\{(\widehat{W}_V,\widehat{W}_{QK},\widehat{W}_C):\widehat{W}_V\in\widehat{\mathcal{W}}_V,\widehat{W}_V\in\widehat{\mathcal{W}}_V,\widehat{W}_C\in\widehat{\mathcal{W}}_C(\widehat{W}_V,\widehat{W}_{QK})\right\}$.

Using Lemma A.12, we can show that $\widehat{\mathcal{W}}$ is an ε-cover for $g(\cdot; \{W_V, W_{QK}, W_C\})$ and inputs $Z^{(1)}, \ldots, Z^{(m)}$ where

$$\varepsilon = \varepsilon_C + \frac{L_\sigma}{2} cs\varepsilon_{QK} + L_\sigma c\varepsilon_V.$$

Using Lemma 4.5, the size of the cover is,

$$|\widehat{\mathcal{W}}| \leq |\widehat{\mathcal{W}}_V||\widehat{\mathcal{W}}_{QK}| \max_{\substack{\widehat{W}_V \in \widehat{\mathcal{W}}_V \\ \widehat{W}_{QK} \in \widehat{\mathcal{W}}_{QK}}} \left|\widehat{\mathcal{W}}_C(\widehat{W}_V, \widehat{W}_{QK})\right|$$

$$\implies \log|\widehat{\mathcal{W}}| \lesssim \left(\frac{k\alpha^2}{\varepsilon_V^2} + \frac{k\beta^2}{\varepsilon_{QK}^2} + \frac{d\gamma^2}{\varepsilon_C^2}\right) \log(mT).$$

We are now ready to inductively construct a cover for the deeper network. Suppose we have a $\varepsilon^{(i)}$-cover $\widehat{\mathcal{W}}^{1:i}$ for $g(\cdot; W^{1:i})$ on $X^{(1)}, \cdots, X^{(m)}$. We show how to construct an ε^{i+1}-cover for $g(\cdot; W^{1:i+1})$. For every element $\widehat{W}^{1:i} \in \widehat{\mathcal{W}}^{1:i}$ we construct a $\left(\varepsilon_C^{(i)} + \frac{L_\sigma}{2} B_C^{(i)} B_V^{(i)} \varepsilon_{QK}^{(i)} + L_\sigma B_C^{(i)} \varepsilon_V^{(i)}\right)$-cover $\widehat{\mathcal{W}}_i(\widehat{W}^{1:i})$ for the transformer layer (as above) on inputs $\left\{g(X^{(j)}; \widehat{W}^{1:i})\right\}_{j=1}^{(m)}$. Consider the cover, $\widehat{\mathcal{W}}^{1:i+1} := \left\{(\widehat{W}^{1:i}, \widehat{W}^{(i)}) : \widehat{W}^{1:i} \in \widehat{\mathcal{W}}^{1:i}, \widehat{W}^{(i)} \in \widehat{\mathcal{W}}_i(\widehat{W}^{1:i})\right\}$. By Lemma A.12, this gives,

$$\varepsilon^{i+1} = L_\sigma B_C^{(i)} B_V^{(i)} (1 + B_{QK}^{(i)}) \varepsilon^{(i)} + \varepsilon_C^{(i)} + \frac{L_\sigma}{2} B_C^{(i)} B_V^{(i)} \varepsilon_{QK}^{(i)} + L_\sigma B_C^{(i)} \varepsilon_V^{(i)}.$$

The size of the cover is.

$$|\widehat{\mathcal{W}}^{1:i+1}| \leq |\widehat{\mathcal{W}}^{1:i}| \max_{\widehat{W}^{1:i} \in \widehat{\mathcal{W}}^{1:i}} \left|\widehat{\mathcal{W}}_i(\widehat{W}^{1:i})\right|.$$

Inductively applying this, we get

$$\begin{aligned}
\varepsilon^{L+1} &= \sum_{i=1}^{L} \left(\prod_{j<i} L_\sigma B_C^{(j)} B_V^{(j)} (1 + B_{QK}^{(j)})\right) \left(\varepsilon_C^{(i)} + \frac{L_\sigma}{2} B_C^{(i)} B_V^{(i)} \varepsilon_{QK}^{(i)} + L_\sigma B_C^{(i)} \varepsilon_V^{(i)}\right) \\
&= \sum_{i=1}^{L} \alpha_i \left(\varepsilon_C^{(i)} + \frac{L_\sigma}{2} B_C^{(i)} B_V^{(i)} \varepsilon_{QK}^{(i)} + L_\sigma B_C^{(i)} \varepsilon_V^{(i)}\right)
\end{aligned}$$

where $\alpha_i = \prod_{j<i} L_\sigma B_C^{(j)} B_V^{(j)} (1 + B_{QK}^{(j)})$.

The size of the cover is

$$\log\left(|\widehat{\mathcal{W}}^{1:L+1}|\right) \leq \sum_{i=1}^{L} \left(\frac{k^2 {B_V^{\infty}}^{(i)^2}}{{\varepsilon_V^{(i)}}^2} + \frac{d^2 {B_\infty^F}^{(i)^2}}{{\varepsilon_{QK}^{(i)}}^2} + \frac{d^2 {B_C^{\infty}}^{(i)^2}}{{\varepsilon_C^{(i)}}^2}\right) \log(mT).$$

Notice that the layer-norm maintains the norm bound on the inputs. Lastly, we need to cover the linear layer on the `[CLS]` token and compose it with the cover of $g^{1:L}$ (as before). Using Lemma A.8 and A.13, we can get the final ε-cover $\mathcal{C}$ with

$$\varepsilon = B_w \sum_{i=1}^{L} \alpha_i \left(\varepsilon_C^{(i)} + \frac{L_\sigma}{2} B_C^{(i)} B_V^{(i)} \varepsilon_{QK}^{(i)} + L_\sigma B_C^{(i)} \varepsilon_V^{(i)}\right) + B_w \varepsilon_w$$

and size

$$\log|\mathcal{C}| \leq \frac{B_W^2 L_\sigma^2 \log(m)}{\varepsilon_w^2} + \sum_{i=1}^{L} \left(\frac{k^2 {B_V^{\infty}}^{(i)^2}}{{\varepsilon_V^{(i)}}^2} + \frac{d^2 {B_{QK}^{\infty}}^{(i)^2}}{{\varepsilon_{QK}^{(i)}}^2} + \frac{d^2 {B_C^{\infty}}^{(i)^2}}{{\varepsilon_C^{(i)}}^2}\right) \log(mT)$$

Using Lemma A.6, the size of the cover for fixed ε gives us the desired result. □

B PROOFS FOR SPARSE FUNCTION REPRESENTATION

B.1 SETUP AND USEFUL LEMMAS

To define the reduction from Boolean circuits to Transformers, we will fix a set of approximately orthonormal vectors $\{v_i : \|v_i\| = 1\}_{i=1}^{T'}$ in $\mathbb{R}^d$ (which can be thought of as "word embeddings" in a d-dimensional space): for each $i \neq j$, we have $|v_i^\top v_j| \leq \Delta$. When $\Delta = 0$, the maximal T' for which such a set exists is d; for $\Delta \in (0, \frac{1}{2})$, the Johnson-Lindenstrauss lemma implies that the maximal set of is of size $\exp(\Theta(d\Delta^2))$. For a given choice of d, Δ and a maximal $\{v_1, \ldots, v_{T'}\}$, this construction for embedding Boolean function learning problems is well-defined for contexts of length $T \leq T'$. (We will also require a special vector $v_{\texttt{[CLS]}}$ which is exactly orthogonal to all the v_i; this just requires one extra dimension.)

As outlined in the main paper, for a context length of T, we fix a Δ-orthonormal set $v_{1:T}$, and stack $v_{1:T}$ into a matrix $V \in \mathbb{R}^{T \times d}$; further, we assume there is a unit vector $v_{\texttt{[CLS]}}$ such that $v_{\texttt{[CLS]}}^\top v_j = 0$ for each $j \in [T]$. We map binary strings of length T to contexts as follows: $v_i^\top$ is included as a row of the input X if and only if $b_i = 1$. Then, we will state our Boolean function approximation results using the scalar classification head convention: for a context length T, we will specify a scalar-output Transformer $f_{\mathsf{tf\text{-}scalar}}$ with inputs $X \in \mathbb{R}^{(T+1) \times d}$, parameters W_Q, W_K, W_V, W_C, w, fixed auxiliary-token input $x_{\texttt{[CLS]}} := v_{\texttt{[CLS]}}$, and a linear activation $\sigma(x) = x$. For some $\varepsilon > 0$ and a Boolean function $f : \{0,1\} \to \mathbb{R}$, $\widehat{f} \in \mathcal{F}_{\mathsf{tf\text{-}scalar}}$ which ε-uniformly approximates f if

$$\left| \widehat{f}(V \mathrm{diag}(b)) - f(b) \right| \leq \varepsilon, \qquad \forall b \in \{0,1\}^T.$$

Recall the functional form of $f \in \mathcal{F}_{\mathsf{tf\text{-}scalar}}$ (choosing $x_{\texttt{[CLS]}} = v_{\texttt{[CLS]}}$ to be a constant):

$$f(X; W_Q, W_K, W_V, W_C, w) = \mathsf{softmax}\left(v_{\texttt{[CLS]}}^\top W_Q W_K^\top X^\top\right) X W_V W_C w.$$

Finally, we state some useful lemmas.

We will use a construction which approximates a "hard selection" of s indices using the softmax mixture; for this, we will need to quantify the approximation error when the inputs to the softmax function are bounded.

Lemma B.1 (Softmax truncation)**.** *Let $z \in (\mathbb{R} \cup \{-\infty\})^T$ such that $z_t \geq R$ for each $1 \leq t \leq s$, and $z_t \leq 0$ for each $s+1 \leq t \leq T$. Define $z' \in (\mathbb{R} \cup \{-\infty\})^T$ so that $z'_t = z_t$ for $1 \leq t \leq s$, and $z_t = -\infty$ for $s+1 \leq t \leq T$. Then, letting $e^{-\infty} = 0$ in the definition of $\mathsf{softmax}(\cdot)$, we have*

$$\|\mathsf{softmax}(z') - \mathsf{softmax}(z)\|_1 \leq 2\frac{T-s}{s\exp(R)} < \frac{2T}{\exp(R)}.$$

Proof. We have

$$\|\mathsf{softmax}(z') - \mathsf{softmax}(z)\|_1 = \sum_{t=1}^{s} \exp(z_t)\left(\frac{1}{\mathbf{1}^\top \exp(z')} - \frac{1}{\mathbf{1}^\top \exp(z)}\right) + \sum_{t=s+1}^{T} \frac{\exp(z_t)}{\mathbf{1}^\top \exp(z)}.$$

The first summation is equal to

$$1 - \frac{\mathbf{1}^\top \exp(z')}{\mathbf{1}^\top \exp(z)} \leq \frac{T-s}{s\exp(R)},$$

while the same upper bound holds for the second summation, since each term is at most $\frac{1}{s\exp(R)}$. □

We state a very simple Boolean function approximation result, choosing 3-layer ReLU networks for simplicity.

Lemma B.2 (Exact Boolean function representation with a 3-layer ReLU net)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$, and let $x_1, \ldots, x_n \in \mathbb{R}^d$ such that $\|x_i\|_\infty \leq B$ for each $i \in [n]$, $\|x_i - x_j\|_\infty \geq 4\delta$ for each $i \neq j \in [n]$. Then, there are choices of $W_1 \in \mathbb{R}^{h_1 \times d}, b_1 \in \mathbb{R}^d, W_2 \in \mathbb{R}_2^h \times h_1, b_2 \in \mathbb{R}^{h_1}, w \in \mathbb{R}_2^h, b \in \mathbb{R}$ such that*

$$w^\top \mathsf{ReLU}\left(W_2 \mathsf{ReLU}(W_1(x_i + z) + b_1) + b_2\right) + b = f(x_i)$$

for all $i \in [n]$ *and* $\|z\|_\infty \leq \delta$*, where* $\mathsf{ReLU}(x) := x_+ = \max(0, x)$ *is applied entrywise, with*

$$h_1 = 4nd, \quad h_2 = n,$$

$$\|W_1\|_\infty, \|b_1\|_\infty \leq \frac{1}{\delta}, \quad \|W_2\|_\infty, \|b_2\|_\infty \leq 1, \quad \|w\|_\infty, |b| \leq 4 \max_{i \in [n]} |f(x_i)|.$$

Proof. (Brief sketch; the technique is standard (Nielsen, 2015).) One can use 2 ReLU functions to represent a threshold function with a ramp of width $1/\delta$: $\nu(x) := (x/\delta)_+ - (x/\delta + 1)_+$, and thus a bump function $\psi(x) := \nu(x) - \nu(2\delta - x)$, whose value is 1 on the interval $[0, 2\delta]$, and 0 on $(-\infty, -\delta]$ and $[3\delta, \infty)$. The first layer W_1, b_1 is constructed so that the activations can be combined by W_2 to represent the bump functions at each coordinate of each x_i. W_2, b_2 are constructed so that w can represent the bump function which indicates $\|x - x_i\| \leq \delta$. w is then selected to multiply each such bump function with $f(x_i)$, and take a sum over $i \in [n]$. □

B.2 Representing a symmetric monotone Boolean function

In this section, we complete the proof that a scalar-output Transformer can represent a particular class of real-valued Boolean functions $f_{\mathcal{I}} : \{0,1\}^T \to \mathbb{R}$, defined for each context length T and non-empty index set $\mathcal{I} \subseteq [T]$. We will construct this function such that it is sparse (its values only depend on $\mathcal{I}$), symmetric (its values only depend on $\sum_{i \in \mathcal{I}} b_i$), and monotonic (its values are strictly increasing in $\sum_{i \in \mathcal{I}} b_i$). This serves as a warm-up for the more general function approximation result for Transformer layers composed with feedforward networks, but also elucidates the representational power of an attention head alone: the output of this function can then be composed with a threshold classifier to obtain the AND and OR functions on the same index set $\mathcal{I}$.

The function we will approximate is defined as follows:

$$f_{\mathcal{I}}(r) := \frac{r - s}{r + 1}, \quad \text{where } r = \sum_{i \in \mathcal{I}} b_i, s = |\mathcal{I}|.$$

Note that this function is non-positive and strictly increasing in r in the range of bit counts $r = 0, \ldots, s$.

Proposition [. *4.7] Fix a context size* T *and* Δ*-orthogonal basis set of* $T + 1$ *elements* $v_1, \ldots, v_T, v_{\texttt{[CLS]}} \in \mathbb{R}^d$*. Let* $\mathcal{I} \subseteq [T], |\mathcal{I}| = s \geq 1$*, and assume* $s < \frac{1}{2\Delta}$*. For any* $\varepsilon > 0$*, there is a choice of* $\widehat{f}_{\mathcal{I}} \in \mathcal{F}_{\text{tf-scalar}}$ *such that* $\widehat{f}_{\mathcal{I}}$ $(\varepsilon + s\Delta)$*-uniformly approximates* $f_{\mathcal{I}}$*, with weight norms satisfying*

$$\|W_Q\| \leq \frac{\log\left(\frac{4Ts}{\varepsilon}\right)}{1 - 2s\Delta}, \quad \|W_K\| \leq 2, \quad \|W_V\| \leq \sqrt{2}s, \quad \|W_C\| \leq s, \quad \|w\| = 1.$$

Proof. Let $e_1^{(k)}, \ldots, e_k^{(k)}$ be the standard basis in $\mathbb{R}^k$; let $e_1^{(T)}, \ldots, e_T^{(T)}, e_{\texttt{[CLS]}}^{(T)}$ be the standard basis in $\mathbb{R}^{T+1}$ with the special `[CLS]` index. Let $n(i)$ be a unique bijection between $\mathcal{I}$ and $[s]$. The construction is as follows (obtaining norm bounds via the triangle equality on these summations):

- $W_Q := R v_{\texttt{[CLS]}} e_1^{(k)\top}$, for a positive real number R we will pick later.
- $W_K := (\sum_{i \in \mathcal{I}} v_i + v_{\texttt{[CLS]}}) e_1^{(k)\top}$.
- $W_V := \sum_{i \in \mathcal{I}} v_i e_{n(i)}^{(k)\top} - v_{\texttt{[CLS]}} \left(\sum_{i \in \mathcal{I}} e_{n(i)}^{(k)}\right)^\top$.
- $W_C := \sum_{i \in \mathcal{I}} e_{n(i)}^{(k)} v_1^\top$.
- $w := v_1$.

We will first argue that when $\Delta = 0$, and in the limit as $R \to \infty$, this construction captures the desired function exactly. We have

$$v_{\texttt{[CLS]}}^\top W_Q W_K^\top = R \left(\sum_{i \in \mathcal{I}} v_i + v_{\texttt{[CLS]}}\right)^\top$$

If $\Delta = 0$, then

$$v_{\texttt{[CLS]}}^\top W_Q W_K^\top X^\top = R\left(\sum_{i\in\mathcal{I}} b_i e_i^{(T)} + e_{\texttt{[CLS]}}^{(T)}\right)^\top.$$

As $R \to \infty$, the softmax distribution converges to the uniform distribution on the indices $i \in \mathcal{I}$ such that $b_i = 1$ and the `[CLS]` index:

$$\mathsf{softmax}\left(v_{\texttt{[CLS]}}^\top W_Q W_K^\top X^\top\right) \to \frac{1}{r+1}\left(\sum_{i\in\mathcal{I}} b_i e_i^{(T)} + e_{\texttt{[CLS]}}^{(T)}\right),$$

where $r = \sum_{i\in\mathcal{I}} b_i$. Calling this limit u, we have

$$y_{\texttt{[CLS]}} = u^\top X W_V W_C w = \frac{1}{r+1}\left(\sum_{i\in\mathcal{I}} b_i - \sum_{i\in\mathcal{I}} 1\right) = \frac{1}{r+1}(r-s) = f_{\mathcal{I}}(b),$$

as claimed. Now, let us propagate the approximation errors in the case of finite R and nonzero Δ. Since the scalar factor in $W_K^\top v_j = (\sum_{i\in\mathcal{I}} v_i + v_{\texttt{[CLS]}})^\top v_j e_1^{(k)}$ is a sum over s inner products between the v_j, we have $|W_K^\top v_i - b_i| \le s\Delta$. The input to the softmax $v_{\texttt{[CLS]}}^\top W_Q W_K^\top X^\top$ is thus

$$R\left(\sum_{i\in\mathcal{I}} b_i e_i^{(T)} + e_{\texttt{[CLS]}}^{(T)} + \xi\right)^\top,$$

where $\|\xi\|_\infty \le s\Delta$. Applying Lemma B.1 to

$$R\left(\sum_{i\in\mathcal{I}} b_i e_i^{(T)} + e_{\texttt{[CLS]}}^{(T)} + \xi - s\Delta\mathbf{1}\right)^\top,$$

we conclude that

$$\|\tilde{u} - u\|_1 \le \frac{2T}{\exp\left((1-2s\Delta)R\right)},$$

letting $\tilde{u}$ denote $\mathsf{softmax}\left(v_{\texttt{[CLS]}}^\top W_Q W_K^\top X^\top\right)$.

Finally, by the same argument as for the inexact inner products in XW_K, we have

$$XW_V W_C = \left(\sum_{i\in\mathcal{I}}(b_i - 1) + \xi'\right) v_1^\top$$

for $\|\xi'\|_\infty \le s\Delta$. Thus, we have

$$\begin{aligned} y_{\texttt{[CLS]}} &= \tilde{u}^\top\left(\sum_{i\in\mathcal{I}}(b_i-1) + \xi'\right) = u^\top\left(\sum_{i\in\mathcal{I}}(b_i-1) + \xi'\right) + (\tilde{u}-u)^\top\left(\sum_{i\in\mathcal{I}}(b_i-1) + \xi'\right) \\ &= f_{\mathcal{I}}(b) + u^\top\zeta' + (\tilde{u}-u)^\top\left(\sum_{i\in\mathcal{I}}(b_i-1) + \xi'\right). \end{aligned}$$

Notice that $u^\top\zeta' \le s\Delta$, and the absolute value of the last term is bounded by

$$\|\tilde{u} - u\|_1 \left\|\left(\sum_{i\in\mathcal{I}}(b_i-1) + \xi'\right)\right\|_\infty \le \frac{2Ts(1+\Delta)}{\exp\left((1-2s\Delta)R\right)} \le \frac{4Ts}{\exp\left((1-2s\Delta)R\right)}.$$

The final result is concluded by making this term at most ε, by choosing

$$R = \frac{\log\left(\frac{4Ts}{\varepsilon}\right)}{1-2s\Delta}.$$

□

B.3 APPROXIMATING ARBITRARY SPARSE BOOLEAN FUNCTIONS

We now proceed to use the ideas from the proof of Proposition 4.7 to approximate arbitrary Boolean functions. To do this, we will augment the original scalar-output Transformer class with a 3-layer feedforward ReLU network of hidden layer. We call this function class $\mathcal{F}^{(\mathrm{mlp})}_{\text{tf-scalar}}$, with functions of the form

$$f(X; W_Q, W_K, W_V, W_C, W_1, W_2, w) = w^\top \mathsf{ReLU}\left(\left(\mathsf{ReLU}\left((\phi + b_1)W_1\right) + b_2\right) W_2\right) + b,$$

where

$$\phi := \mathsf{softmax}\left(v_{\texttt{[CLS]}}^\top W_Q W_K^\top X^\top\right) X W_V W_C,$$

and $\mathsf{ReLU}(x) = \max(0, x)$ is applied entrywise. Here, $W_h \in \mathbb{R}^{d\times h}$, and $w \in \mathbb{R}^h$.

Proposition [. *4.8] Fix a context size T and Δ-orthogonal basis set of $T+1$ elements $v_1, \ldots, v_T, v_{\texttt{[CLS]}} \in \mathbb{R}^d$. Let $\mathcal{I} \subseteq [T], |\mathcal{I}| = s \geq 1$; assume $s \leq d$. Let $f : \{0,1\}^T \to \mathbb{R}$ be an arbitrary function whose value only depends on the bits at positions $i \in \mathcal{I}$, and $|f(b)| \leq B$ for all $b \in \{0,1\}^T$. Then, there is a choice of $\widehat{f} \in \mathcal{F}_{\text{tf-scalar}}$ such that $\widehat{f}$ ε-uniformly approximates f, with weight norms satisfying*

$$\|W_Q\| \leq \frac{\log\left(\frac{2T}{\Delta}\right)}{1 - 2s\Delta}, \quad \|W_K\| \leq 2, \quad \|W_V\| \leq \sqrt{2}s, \quad \|W_C\| = s,$$

$$\|W_1\|, \|b_1\|, \|W_2\|, \|b_2\| \leq \frac{4s2^s}{\Delta}, \quad \|w\|, b \leq B.$$

Proof. We use the same construction as in Proposition 4.7 for the attention component:

- $W_Q := R v_{\texttt{[CLS]}} e_1^{(k)\top}$, for a positive real number R we will pick later.
- $W_K := (\sum_{i\in\mathcal{I}} v_i + v_{\texttt{[CLS]}}) e_1^{(k)\top}$.
- $W_V := \sum_{i\in\mathcal{I}} v_i e_{n(i)}^{(k)\top} - v_{\texttt{[CLS]}} \left(\sum_{i\in\mathcal{I}} e_{n(i)}^{(k)}\right)^\top$.

Instead of projecting the output of the mixing step, we use W_C to map it to the hypercube. Let $e_1^{(d)}, \ldots, e_k^{(d)}$ denote the standard basis in $\mathbb{R}^d$; then we set

- $W_C := \sum_{i\in\mathcal{I}} e_{n(i)}^{(k)} e_{n(i)}^{(d)\top}$.

In the exact case of $\Delta = 0, R \to \infty$, for each $b \in \{0,1\}^T$, the image of

$$\tilde{\phi}(b) := \mathsf{softmax}\left(v_{\texttt{[CLS]}}^\top W_Q W_K^\top X^\top\right) X W_V W_C, \quad X = V\mathrm{diag}(b)$$

is

$$\phi_{\text{ideal}}(b) := \frac{1}{r+1} \sum_{i\in\mathcal{I}} (b_i - 1) e_{n(i)}^{(d)\top},$$

where $r = \sum_{i\in\mathcal{I}} b_i$; note that these are distinct for each b. Thus, we can select the MLP weights W_h, w using a standard universal function approximation argument. Before doing so, let us propagate the approximation error. We again have

$$\|\tilde{u} - u\|_1 \leq \frac{2T}{\exp\left((1 - 2s\Delta)R\right)},$$

where $\tilde{u}$ denotes $\mathsf{softmax}\left(v_{\texttt{[CLS]}}^\top W_Q W_K^\top X^\top\right)$, and u is the uniform mixture $\frac{1}{r+1}\left(\sum_{i\in\mathcal{I}} b_i e_i^{(T)} + e_{\texttt{[CLS]}}^{(T)}\right)$. Repeating the analysis of XW_VW_C, we have that for each $j \in [T]$ that $v_j^\top W_V W_C = b_i \mathbf{1}_{j\in\mathcal{I}} + e_{n(i)}^{(d)\top} + \xi_j$ for $\|\xi_j\|_\infty \leq s\Delta$. Setting $R = \frac{\log^+\left(\frac{2T}{\Delta}\right)}{1-2s\Delta}$, we get

$$\left\|\tilde{\phi}(b) - \phi_{\text{ideal}}(b)\right\|_\infty \leq 2s\Delta.$$

To complete the proof, we can instantiate Lemma B.2 on the first s coordinates of ϕ, with $\delta = 2s\Delta$, $n = 2^s$. □

C EXPERIMENT DETAILS

C.1 EMPIRICAL SCALING LAWS FOR LEARNING SPARSE AND GATES

In this section, we provide details for the empirical sample complexity scaling law experiments, which are the main empirical verification of the $\log T$ dependence of the sample complexity arising from the analysis.

Experimental setup. Synthetic tasks, parameterized by sample size m and context T, were generated by the protocol described in the main paper: in each trial, one of the $\binom{T}{3}$ subsets of indices was selected uniformly at random, under i.i.d. random inputs $X \sim \mathrm{Unif}(\{0,1\}^T)$. A 1-layer Transformer network (with the scalar output convention) was trained with Adam (Kingma & Ba, 2014) and batch gradients of the cross entropy loss for binary classification. For each choice of $T \in \{100, 150, \ldots, 350, 400\} \cup \{500, 600, 700, 800\}$ and $B \in \{50, 60, 70, \ldots, 200\}$, 50 independent trials were performed (re-randomizing the dataset generation, random initialization, and dropout masks). Cross-validation was performed on a holdout sample of size 10^4 every 10 iterations. At the end of 1000 training iterations, the trial was counted as a success if the maximum validation accuracy throughout training was greater than 0.99. (In 100% of runs, the training loss was driven to $< 10^{-4}$, with 100% training accuracy, within 1000 iterations.)

Architecture. Like (Chen et al., 2021a), our experimental setup is based on a popular PyTorch implementation (`https://github.com/karpathy/minGPT`), with some optimizations for faster 1-layer training and inference. This implementation includes widely-used architectural details which deviate slightly from the theoretical presentation; refer to the referenced repository for details. All hyperparameter settings left undiscussed are taken from the defaults in this codebase.

Hyperparameters. A fixed architecture was used ($d = 64$, 16 parallel heads), with trainable positional embeddings initialized with Gaussian entries $\mathcal{N}(0, \sigma^2)$, $\sigma = 0.02$, 3 input token embeddings (corresponding to $0, 1$, `[CLS]`), and 2 output embeddings (corresponding to the possible labels $0, 1$). For regularization mechanisms, typical choices were used: 0.1 for {attention, embedding, output} dropout; 10^{-4} weight decay. The Adam optimizer was instantiated with typical parameters $\eta = 10^{-3}, \beta_1 = 0.9, \beta_2 = 0.999$.

Results and discussion. Our findings are summarized by Figure 4, enlarged from the main paper. On the left, the fraction of successful trials is plotted for $T = \{100, 200, 400, 800\}$ with standard errors derived from the normal approximation. Notice that in this "low-data" regime, Transformer training is quite sensitive to the stochasticity in the experiments, including the training data sample (which needs to be sublinear in the context size), so there is no sharp "phase transition". On the right, critical sample sizes are shown, defined as the smallest m for which the fraction of successes was greater than 0.5, with standard errors derived from 50 bootstrap samples. We observe that this architecture is able to solve this "planted sparse Boolean function" task with a sublinear scaling in the sample size.

Infrastructure and computational costs. Each training run took at most 20 minutes on an NVIDIA RTX A6000 GPU (with most of computation time spent on cross-validation). Although these experiments are not in the same regime as state-of-the-art settings (such as BERT pretraining), they require optimized GPU implementations to run in a reasonable timeframe (days, as opposed to months).

C.2 LEARNING PARITIES

A natural question for further exploration is whether Transformers can learn other Boolean functions. In particular, the statistical probe presented for the s-way AND function is equally valid for XOR. The latter is arguably more intriguing: parities are the basis elements in the monomial (i.e. Fourier) expansion of a Boolean function; (O'Donnell, 2021); furthermore, there are computational hardness barriers; see the works cited in the main paper for further context.

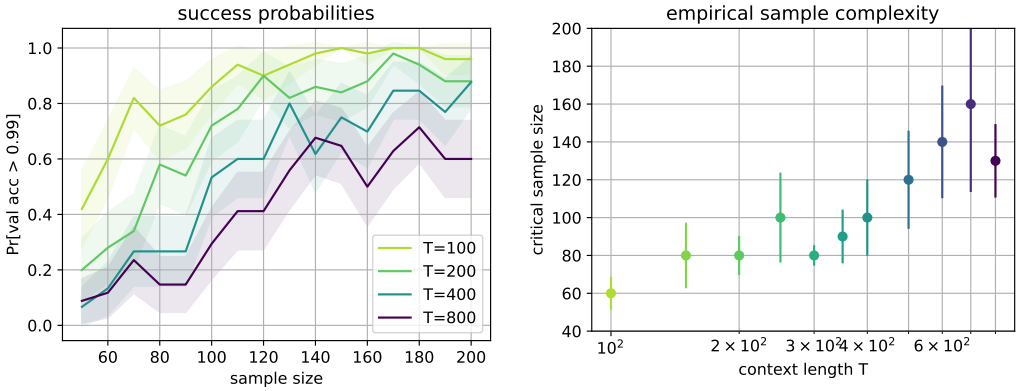

Figure 4: Enlarged plots from Figure 2, the main experimental validation of our theory.

Experimental setup. These experiments match the setting of the main $mathsfAND$ experiments, except for a few differences, which we enumerate here. Gradient-based training is done on streaming online losses (so that there is no training/validation split), with batch size 2048, for 10000 iterations (since parities take significantly longer to fit).

Discussion. The key observation of this exploratory experiment is to point out the phenomenon that Transformer architectures can fit sparse parities at all (rather than to measure any particular trend), pointing to computational mysteries discussed in the main paper. We do not present empirical scaling laws for this set of experiments, as the experiments with small batch sizes are higher in variance, and we are unable to scale this phenomenon (i.e. learn any parities) at context sizes above $T = 50$.