VC Dimension and Expressivity of Real-Valued Transformers

Anonymous authors
Paper under double-blind review

Abstract

Whereas previous results on abilities and limitations of transformers have restricted the definition of transformers in various ways, here we study softmax-attention, multi-layer transformers operating on real values, with very few additional assumptions. Applying results from real geometry, we obtain upper bounds on the VC dimension and split VC dimension of such transformers ($O(n^4)$ and $O(n^6)$, respectively, where n is the input length). Conversely, we also construct specific transformers witnessing lower bounds on these quantities ($\Omega(n)$ in each case). These results have some notable consequences. For example, within the class of symmetric (permutation-invariant) functions, we show that transformers can uniformly express all functions over an alphabet of one symbol and non-uniformly express all functions over an alphabet of two symbols, but cannot (even non-uniformly) express some functions over an alphabet of six symbols. We also prove limitations on how many bits of a real number a transformer can access.

1 Introduction

Previous results on abilities and limitations of transformers have imposed restrictions of various kinds: on attention (e.g., hard attention instead of soft attention (Hahn, 2020; Pérez et al., 2021; Yang et al., 2026b)), on numerical precision (e.g., $O(\log n)$ bits where n is the input sequence length (Merrill & Sabharwal, 2023; Chen et al., 2025)), or on the number of layers (Kozachinskiy et al., 2025; 2026). In addition, previous results on limitations of transformers have relied on complexity-theoretic conjectures, most notably $\mathsf{TC}^0 \neq \mathsf{NC}^1$. Indeed, Kozachinskiy et al. (2026) state that it is still an open problem to find any particular task that is not expressible by multi-layer infinite-precision transformers.

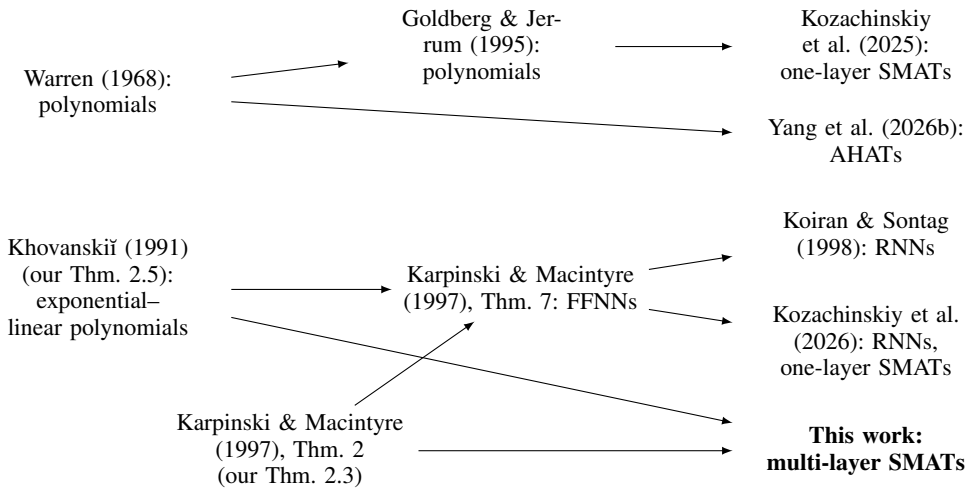


Figure 1: Relationship to previous work. FFNN = feedforward neural network, RNN = recurrent neural network, AHAT = average-hard attention transformer, SMAT = softmax-attention transformer.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In this paper, we study multi-layer softmax-attention transformers (SMATs) operating on real values. Although it has sometimes been argued that some real-valued neural networks can compute any function (Siegelmann & Sontag, 1994; Merrill et al., 2022), and indeed we prove that real-valued transformers can solve some undecidable problems (Props. D.2 and D.3), we also prove, to our knowledge for the first time, that there are limitations on their expressivity.

As a real-valued neural network can be regarded as a composition of real functions (possibly preceded or followed by discrete functions like a word or position embedding), results from real geometry have proved useful for studying the limitations of these networks. Goldberg & Jerrum (1995) applied a quantitative bound of Warren (1968) on the number of connected components in a set defined using polynomials to bound the VC dimension of functions defined using polynomials, including many neural networks. Although transformers are not polynomials, these results can be applied to transformers in limited ways: Kozachinskiy et al. (2025) used Goldberg & Jerrum's bound to obtain an inexpressibility result for one-layer transformers, and Yang et al. (2026b) used Warren's bound to prove VC dimension bounds for hard-attention transformers.

Karpinski & Macintyre (1997) extended Goldberg & Jerrum's bound to a much broader class of functions. Among other things, they used their theorem, together with a quantitative bound of Khovanskiĭ (1991) on the connected components of a set defined by exponential–linear polynomials, to bound the VC dimension of feedforward neural networks (FFNNs). Later work applied this result to other kinds of neural networks by (at least indirectly) simulating them by FFNNs: Koiran & Sontag (1998), to bound the VC dimension of recurrent neural networks (RNNs), and Kozachinskiy et al. (2026), to prove inexpressibility results for RNNs, state-space models, linear-attention transformers, and one-layer SMATs.

No such simulation by FFNNs is possible in the case of general transformers, which means that analogous bounds for general transformers must directly appeal to the main theorem of Karpinski & Macintyre (1997) in its full generality instead of going through their bound for FFNNs. In this paper, we do just that to obtain a VC dimension bound of $O(n^4)$ and a split VC dimension bound of $O(n^6)$ for multi-layer SMATs, where n is the input length. Applying this general result entails, as observed by Kozachinskiy et al. (2026), "significant additional technical work." (See Fig. 1 for a summary of how this paper relates to previous work.)

Some particular consequences of these results are:

- There exist symmetric functions over an alphabet of six symbols that cannot be expressed by any real-valued transformer.
- Real-valued transformers cannot extract the k-th bit of an inputed number if $k \in \omega(n^4)$. Intuitively, although real-valued transformers are allowed infinite precision, they cannot do much with that precision beyond $O(n^4)$ bits.

We also provide examples of tasks that transformers can express, complementing the impossibility results summarized above.

- There exists a transformer that can compute any symmetric function on an alphabet of two symbols (with a different positional encoding for each symmetric function) (Prop. D.2).
- There exists a transformer (with one layer) that can look up the value of the bit at a queried position in an input string (Prop. 6.3).

As we will see below, the two examples above will provide lower bounds on the maximum possible VC dimension of an untrained transformer and split VC dimension of a trained transformer, respectively. We summarize our upper and lower bounds on these quantities in Tab. 1.

Table 1: VC dimension bounds for transformers.

	VC dimension (untrained transformers)		split-VC dimension (trained transformers)	
upper bound	$O(n^4)$	(Thm. 3.1)	$O(n^6)$	(Cor. 3.5)
lower bound	$\Omega(n)$	(Prop. D.2)	$\Omega(n)$	(Prop. 6.3)

2 BACKGROUND

2.1 NOTATION

We write $[n]$ for the set $\{1, \ldots, n\}$ and $\log$ for the base-2 logarithm.

Definition 2.1 (Bachmann–Landau notation)**.** Given functions $f, g \colon \mathbb{N} \to \mathbb{N}$, we write:

- $g \in O(f)$ if there exist c and N such that, for all $n \geq N$, $g(n) \leq cf(n)$,
- $g \in \Omega(f)$ if there exist $c > 0$ and N such that, for all $n \geq N$, $cf(n) \leq g(n)$,
- $g \in \omega(f)$ if, for all $c > 0$, there exists an N such that, for all $n \geq N$, $cf(n) \leq g(n)$, and
- $g \in \tilde{O}(f)$ if there exists a k such that $g \in O(f \log^k f)$.

When the functions f and g are related to a transformer T, we write $g \in O_T(f)$ to emphasize that the constant c is allowed to depend on the transformer T.

2.2 PREVIOUS RESULTS ON VC DIMENSION

Definition 2.2. Let X be a set, $\mathcal{S}$ a family of subsets of X, and A a finite subset of X. We say that $\mathcal{S}$ *shatters* A if, for all $A' \subseteq A$, there is a set $S \in \mathcal{S}$ such that $S \cap A = A'$. Define the *VC dimension of* $\mathcal{S}$ (denoted $\mathrm{VC}(\mathcal{S})$) to be the greatest n such that $\mathcal{S}$ shatters a set of size n (or ∞ if none).

Throughout this paper, $X = \mathbb{R}^m$ for some m. We will be adapting the methods of Karpinski & Macintyre (1997) from FFNNs to transformers. We represent an untrained neural network as a finite Boolean combination $\phi(\bar{x}, \bar{y})$ of expressions of the form $\tau(\bar{x}, \bar{y}) = 0$ or $\tau(\bar{x}, \bar{y}) > 0$, where $\tau(\bar{x}, \bar{y})$ is a C^∞ function from $\mathbb{R}^{m+k}$ to $\mathbb{R}$. Here, $\bar{x}$ is an m-tuple of *input* variables, and $\bar{y}$ a k-tuple of *parameter* variables. This ϕ gives rise to a family of subsets of $\mathbb{R}^m$, indexed by parameters in $\mathbb{R}^k$:

$$\mathcal{S}_\phi = \{\{\bar{a} \in \mathbb{R}^m \mid \phi(\bar{a}, \bar{\theta})\} \mid \bar{\theta} \in \mathbb{R}^k\}.$$

In passing, when the τ are given by polynomials over $\mathbb{R}$, then such $\mathcal{S}_\phi$ are precisely the *semi-algebraic families of semialgebraic subsets of* $\mathbb{R}^m$. It is well-known that such families have finite VC dimension (Goldberg & Jerrum, 1995).

Let us first describe Theorem 2 from Karpinski & Macintyre (1997). For the necessary background regarding submanifolds of $\mathbb{R}^n$, see App. B. Let $\phi(\bar{x}, \bar{y})$ be as above, and let $\tau_1(\bar{x}, \bar{y}), \ldots, \tau_s(\bar{x}, \bar{y})$ be the functions appearing in ϕ. Define a *p-function* of ϕ to be a function $\bar{y} \mapsto \tau_i(\bar{a}, \bar{y})$, where $\bar{a} \in \mathbb{R}^m$ and τ_i is a function appearing in ϕ.

Theorem 2.3. *Given $\phi(\bar{x}, \bar{y})$ as above, suppose that B is an integer such that for any choice of $r \leq k$ many p-functions $F = (F_1, \ldots, F_r)$ of ϕ, and values $(\epsilon_1, \ldots, \epsilon_r) \in \mathbb{R}^r$, if $F^{-1}((\epsilon_1, \ldots, \epsilon_r))$ is a $(k - r)$-dimensional submanifold of $\mathbb{R}^k$, then it has at most B connected components. Then the VC dimension of $\mathcal{S}_\phi$ is at most $2 \log B + (16 + 2 \log s)k$.*

The application by Karpinski & Macintyre (1997) of Thm. 2.3 to FFNNs and our application to transformers (which are both nontrivial) will depend on a choice of B for suitable ϕ and τ_i and this will be given by a theorem of Khovanskiĭ (1991).

Definition 2.4. A *family of r exponential–linear polynomials of degree at most D in k variables and q exponentials* consists of r functions of the form $P_i(\bar{x}, e^{\Lambda_1(\bar{x})}, \ldots, e^{\Lambda_q(\bar{x})})$ where $\bar{x} = (x_1, \ldots, x_k)$, P_i is a polynomial of degree at most D, and the $\Lambda_j(\bar{x})$ are linear polynomials.

Theorem 2.5 (Khovanskiĭ, 1991)**.** *Let $P_1, \ldots, P_r$ and $\Lambda_1, \ldots, \Lambda_q$ be a family of exponential–linear polynomials of degree at most D in k variables and q exponentials, where $k \geq r$. Assume that*

$$\{\bar{a} \in \mathbb{R}^k \mid P_1(\bar{a}) = 0 \wedge \cdots \wedge P_r(\bar{a}) = 0\}$$

is a submanifold of $\mathbb{R}^k$ of dimension $k - r$. Then the number of connected components of this set is at most $B = 2^{\frac{q(q-1)}{2}} \cdot D^r(rD + k - r + 1)^{k-r}((k - r + 1)(rD + k - r + 1) + r - k)^q$.

If M is the largest of r, k, and D in the theorem statement above, then the bound can be written as $\log B \in O(q^2 + M \log M)$.

2.3 TRANSFORMERS

We follow the standard definition of transformer (App. A; Vaswani et al., 2017). Our main results (Thm. 3.1 and Cor. 3.5) are independent of many details, like attention masking, layer normalization, residual connections, position encodings, and so on.

An *untrained* transformer T has a fixed number of layers and heads and fixed input/hidden dimension d, but depends on unspecified learnable parameters $\bar{y}$. Let k be the number of such parameters; we may think of $\bar{y}$ as belonging to $\mathbb{R}^k$. A *trained* transformer $T_{\bar{\theta}}$ is an untrained transformer T together with some fixed parameter values $\bar{\theta}$.

A transformer operates on sequences. We write $T_{\bar{\theta}}^{(n)}$ for the restriction of $T_{\bar{\theta}}$ to sequences of length n. In Def. A.1, the input is a sequence of real vectors. But in many applications, transformers take discrete inputs and map them to real vectors via word and/or position embeddings (Def. A.2). Our main results (Thm. 3.1 and Cor. 3.5) hold with or without word/position embeddings, and whether they are considered to be fixed or to vary with the learned parameters.

Def. A.1 defines a final output layer, which computes a single real value from the last position. We say that $T_{\bar{\theta}}$ accepts a sequence X of length n iff $T_{\bar{\theta}}^{(n)}(X) > 0$. Then for any input length n, an untrained transformer T gives rise to a family of sets, indexed by parameters $\bar{\theta}$:

$$\mathcal{S}_{T^{(n)}} := \{\{X \in \mathbb{R}^{n\times d} \mid T_{\bar{\theta}}^{(n)}(X) > 0\} \mid \bar{\theta} \in \mathbb{R}^k\}.$$

We are interested in the VC dimension of this family, which we call $\mathrm{VC}(T^{(n)})$ for short. On general grounds (namely, the o-minimality of the field of real numbers equipped with the exponential function), it is finite. We will use the methods of Karpinski & Macintyre (1997) to give the bound $O(n^4)$.

3 MAIN RESULTS

3.1 VC DIMENSION OF TRANSFORMERS

Theorem 3.1. *Let T be an untrained transformer. Then* $\mathrm{VC}(T^{(n)}) \in O(n^4)$.

Proof sketch. See App. C for the full proof.

When Karpinski & Macintyre give their VC dimension bound for sigmoid FFNNs, they apply Thm. 2.3 to a formula $\tau(\bar{x}, \bar{y}) > 0$ where τ is the whole network, which is a C^∞ function. However, to obtain the bound B by appeal to Thm. 2.5, a rather subtle argument is used, adding new variables to obtain a system of exponential–linear polynomial equations.

Unlike an FFNN, a transformer does not compute a C^∞-function, due to the appearance of $\max(-,-)$ in the position-wise feed-forward network $\mathcal{F}$ (in each layer). This is actually the only obstruction, as the various rational expressions always have nonzero denominator and the square root in layer normalization is applied to a positive real number bounded away from 0. So, in the first step, we write a formula $\phi(\bar{x}, \bar{y})$, which is equivalent to a transformer's acceptance condition and therefore does not change its VC dimension. It is a Boolean combination of inequalities involving only C^∞ functions $\tau_1(\bar{x}, \bar{y}), \ldots, \tau_s(\bar{x}, \bar{y})$, where $s \in 2^{O(n)}$, so we can apply Thm. 2.3 to it.

It remains to find the bound B on the number of connected components of the submanifold defined by the equations $F_i(\bar{y}) = \epsilon_i$ for $i \in [r]$, where the F_i are p-functions of ϕ. Now Thm. 2.5 requires a system of exponential–linear polynomials, but the F_i are not. So, we add some new variables and equations to eliminate square roots (in layer normalization) and division (in attention and layer normalization) and to ensure that the argument of each $\exp$ (in attention) is linear. The number of new variables and equations is $\kappa \in O(n^2)$. Then Thm. 2.5 gives that the number of connected components of the set defined by the new equations (which is homeomorphic to the set defined by the old equations) is at most B, where $\log B \in O(q^2 + M \log M)$.

Here, q is the total number of exponentials appearing in the system of equations, which is in $O(n^2)$ (one for each pair of positions, for each layer, head, and for each equation). And M is the greatest among of the number of equations in the system ($r + \kappa$), the number of variables appearing in the system ($k + \kappa$), and the maximum degree of the exponential–linear polynomials used (4). Therefore, $\log B \in O(n^4)$. By Thm. 2.3, the VC dimension of T^n is at most $2 \log B + (16 + 2 \log s)k$ where

s is the number of C^∞ functions appearing in ϕ above. As $\log s \in O(n)$, the VC dimension is in $O(n^4)$, concluding the proof of Thm. 3.1. □

3.2 Inputs as parameters

Kozachinskiy et al. (2025) devised a way of using VC dimension to study the complexity of a single function defined on sequences (as opposed to a family of such functions). Given a set S, let Σ^S denote the set of functions from S to Σ. Identify the set of strings Σ^n and the set of functions $\Sigma^{[n]}$ in the natural way. Given $\bar{x} \in \Sigma^{[n]\setminus K}$ and $\bar{y} \in \Sigma^K$, let $\bar{x} \oplus \bar{y}$ denote the unique function (i.e., string) $\bar{z} \in \Sigma^n$ such that $\bar{z}|_{[n]\setminus K} = \bar{x}$ and $\bar{z}|_K = \bar{y}$.

Definition 3.2. Given a subset $L^{(n)} \subseteq \Sigma^n$ (thought of as a language in most applications) and a subset $K^{(n)} \subseteq [n]$, define

$$\mathcal{S}_{L^{(n)},K^{(n)}} := \{\{\bar{x} \in \Sigma^{[n]\setminus K^{(n)}} \mid \bar{x} \oplus \bar{y} \in L^{(n)}\} \mid \bar{y} \in \Sigma^{K^{(n)}}\}.$$

Given a trained transformer $T_{\bar{\theta}}$ and family of subsets of positions $K_n \subseteq [n]$, define

$$\mathcal{S}_{T^{(n)}_{\bar{\theta}},K^{(n)}} := \{\{X \in (\mathbb{R}^d)^{[n]\setminus K^{(n)}} \mid T^{(n)}_{\bar{\theta}}(X \oplus Y) > 0\} \mid Y \in (\mathbb{R}^d)^{K^{(n)}}\}.$$

By considering the quantity $\mathrm{VC}(\mathcal{S}_{L^{(n)},K^{(n)}})$, we can measure the complexity of a single function (as opposed to a family of functions). While this quantity depends on a specified partition of the input positions, the following does not.

Definition 3.3. Given a subset $L^{(n)} \subseteq \Sigma^n$, define the *split VC dimension of* L to be

$$\text{Split-VC}(L^{(n)}) := \max_{K^{(n)}\subseteq[n]} (\mathrm{VC}(\mathcal{S}_{L^{(n)},K^{(n)}})).$$

Theorem 3.4. *Let $T_{\bar{\theta}}$ be a trained transformer and let $K^{(n)} \subseteq [n]$. Then*

$$\mathrm{VC}\big(\mathcal{S}_{T^{(n)}_{\bar{\theta}},K^{(n)}}\big) \in O(n^4 \cdot |K^{(n)}|^2).$$

Proof. The proof proceeds exactly as in Thm. 3.1, but now, $|K^{(n)}|$ of the input vectors become parameters. Each vector has d components, so there are now $k \in O(|K^{(n)}|)$ parameters. In the application of Thm. 2.3, we must consider up to $r \leq k$ many p-functions, so in the application of Thm. 2.5, the number of exponentials must be multiplied by k: Previously we had $q \in O(n^2)$, but now we have $q \in O(n^2 \cdot |K^{(n)}|)$. Therefore, the VC dimension is in $O(n^4 \cdot |K^{(n)}|^2)$. □

Corollary 3.5. *Let $T_{\bar{\theta}}$ be a trained transformer. Then* $\text{Split-VC}(T^{(n)}_{\bar{\theta}}) \in O(n^6)$.

Proof. For every $K^{(n)} \subseteq [n]$, we have $|K^{(n)}| \in O(n)$. □

If we try to apply this split VC dimension bound (Cor. 3.5), we at once encounter a difficulty.

Proposition 3.6. *Let Σ be a finite alphabet, and let $L \subseteq \Sigma^*$. For all n,* $\text{Split-VC}(L^{(n)}) \leq n \log |\Sigma|$.

Proof. Let $V = \text{Split-VC}(L^{(n)})$, and take a $K^{(n)} \subseteq [n]$ such that $\mathrm{VC}(\mathcal{S}_{L^{(n)},K^{(n)}}) = V$. By the definition of $\mathcal{S}_{L^{(n)},K^{(n)}}$, there must exist at least 2^V many distinct tuples $\bar{y} \in (L^{(n)})^{K^{(n)}}$. Since there are no more than $|\Sigma|^n$ tuples $\bar{y} \in (L^{(n)})^{K^{(n)}}$ in total, it follows that $2^V \leq |\Sigma|^n$, which implies that $V \leq n \log |\Sigma|$. □

Since our split VC dimension bound (Cor. 3.5) is asymptotically higher than this, if we want to find a task that is not expressible by transformers, we must look beyond languages over finite alphabets. We encounter natural examples of such languages below when studying the extent to which transformers can access the bits of a real number (Sec. 5 below).

4 Symmetric functions

Given a language $L \subseteq \Sigma^*$, we let $L^{(n)} \subseteq \Sigma^n$ denote the elements of L with length n.

Definition 4.1. A language $L \subseteq \Sigma^*$ is *symmetric* if, for all $w_0, w_1 \in \Sigma^n$ such that each symbol appears in w_0 the same number of times that it appears in w_1, we have $w_0 \in L^{(n)}$ if and only if $w_1 \in L^{(n)}$. A function $f\colon \Sigma^n \to \{0,1\}$ is *symmetric* if it is the characteristic function of a symmetric language. Let $\mathcal{S}_n^p := \{L^{(n)} \subseteq [p]^n \mid L^{(n)} \text{ is symmetric}\}$. (In the machine learning literature, symmetric functions are sometimes called *permutation equivariant* or *permutation invariant*.)

4.1 Non-expressible functions

Proposition 4.2. *For any $p \geq 1$, $\mathrm{VC}(\mathcal{S}_n^p) = \binom{n+p-1}{n} \in \Omega(n^{p-1})$.*

Proof. Let A be a set of strings of such that for any natural numbers $a_0, \dots, a_{p-1}$ such that $a_0 + \cdots + a_{p-1} = n$, A includes exactly one word w such that, for each i, w has exactly a_i many occurrences of the symbol i. By definition, the family $\mathcal{S}_n^p$ shatters this set A. By the well-known formula for the multiset coefficients (e.g., Stanley, 2012, Sec. 1.2), it follows that

$$\mathrm{VC}(\mathcal{S}_n^p) = \left(\binom{n}{p}\right) = \binom{n+p-1}{n} = \frac{(n+p-1)!}{n!\,(p-1)!} \in \Omega(n^{p-1}). \qquad \square$$

It follows from Thm. 3.1 that there exist symmetric functions on six symbols that cannot be expressed by transformers. A seeming shortcoming of this result is that the functions that it guarantees to be inexpressible are not necessarily computable. We address this critique by pointing out a flipside to this non-uniformity—this inexpressibility result applies to non-uniform transformers (of fixed size) just as well as it does to uniform ones (i.e., it applies to transformers whose parameter values and encoding functions vary with the input length, even uncomputably). Such non-uniform transformers can be used to express uncomputable functions, as we see below in Sec. 4.3.

4.2 Alphabet of size 1

Proposition 4.3. *For any family of (symmetric) functions $f^{(n)}\colon \{\mathtt{1}\}^n \to \{0,1\}$, there exists a trained transformer $T_{\bar{\theta}}$ such that, for any n, $T_{\bar{\theta}}^{(n)}(\mathtt{1}^n) = f^{(n)}(\mathtt{1}^n)$.*

See App. D for the proof. For each (symmetric) function $f^{(n)}\colon \{\mathtt{1}\}^n \to \{0,1\}$, the transformer constructed in this proof is uniform in the sense that the same trained transformer, including word and position embeddings, is used for all input lengths. In Sec. 4.3 below, we extend this result from unary to binary languages, but the positional encoding used there depends on the input length.

4.3 Alphabet of size 2

Proposition 4.4. *There exists a trained transformer $T_{\bar{\theta}}$ such that, for any $n > 0$ and symmetric function $f\colon \{\mathtt{0},\mathtt{1}\}^n \to \{0,1\}$, there exists a positional encoding* $\mathrm{PE}\colon [n] \to \mathbb{R}^d$ *such that $T_{\bar{\theta}}^{(n)}$ with positional encoding* PE *computes f.*

See App. D for the proof. In both this construction and the one from Sec. 4.2, if the symmetric function to be expressed is uncomputable, then the positional encoding used is also uncomputable. In the subsection below, we illustrate the power of real weights themselves by constructing a (real-valued) transformer with a computable positional encoding that expresses an uncomputable function.

4.4 Expressing an uncomputable function using real weights

Let U be a universal prefix-free Turing machine (that is, U can simulate any other prefix-free Turing machine, and if U halts on input x, then it does not halt on input xy for any nonempty y). Chaitin's Ω is defined as the probability that U halts, that is, $\Omega = \sum_{x:U \text{ halts on } x} 2^{-|x|}$. We only need the fact that Chaitin's Ω is an uncomputable left-c.e. real number strictly between 0 and 1. This means that the language $L = \{w \in \{\mathtt{0},\mathtt{1}\}^* \mid \#_{\mathtt{1}}(w)/|w| < \Omega\}$ is computably enumerable but not computable.

Proposition 4.5. *L is recognized by a transformer.*

See App. D for the proof.

5 BIT INDEXING

Although we have seen an example where infinite precision can be used to express a language that would not be expressible otherwise (Prop. D.3), we show here that there are limits to how much a transformer can use a real number as a "data structure."

5.1 DYNAMIC BIT INDEXING

In this section, we describe a general family of tasks centered around computing a specific bit of a real number, where both the bit and the real number are given as input. Given functions $p\colon \mathbb{N} \to [n]$ and $\ell\colon \mathbb{N} \to \mathbb{N}$, the input for this task consists of a real number $x \in [0, 1)$ (coded into vectors in the first $p(n)$ input positions) and a bit $b \in [\ell(n)]$ (coded into vectors in the remaining $n - p(n)$ input positions). The intended output of this task is the b-th bit of x. Our formulation of the task is deliberately agnostic regarding the way that the index and the real are coded as sequences of vectors, since our results hold regardless of the codings used. We will show that, if $\ell(n) \in \omega(n^4(p(n))^2)$, then this task cannot be solved by a transformer.

Definition 5.1. Let $p : \mathbb{N} \to [n]$ and $\ell : \mathbb{N} \to \mathbb{N}$. In the *$p$-partitioned ℓ-bit index task*, the input is a sequence of real vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$. Let $\langle - \rangle_{\text{real}} : (\mathbb{R}^d)^{p(n)} \to [0, 1)$ and $\langle - \rangle_{\text{index}} : (\mathbb{R}^d)^{n-p(n)} \to [\ell(n)]$ be arbitrary surjections (i.e., arbitrary functions encoding real numbers and indices as sequences of $p(n)$ and $n - p(n)$ real vectors, respectively). Let $x := \langle \mathbf{x}_1, \ldots, \mathbf{x}_{p(n)} \rangle_{\text{real}}$ and $b := \langle \mathbf{x}_{p(n)+1}, \ldots, \mathbf{x}_n \rangle_{\text{index}}$. The intended output of the task is the b-th bit of x. We denote this task, regarded as a subset of $(\mathbb{R}^d)^n$ in the natural way, by $\mathsf{bit}^{(n)}_{p,\ell}$.

Proposition 5.2. *Let $p\colon \mathbb{N} \to [n]$, $\ell : \mathbb{N} \to \mathbb{N}$, and $K^{(n)} = [p(n)]$. Then* $\text{VC}(\mathcal{S}_{\mathsf{bit}^{(n)}_{p,\ell}, K^{(n)}}) \geq \ell(n)$.

Proof. For each $x \in [0, 1)$, let $(\mathbf{x}_{1,x}, \ldots, \mathbf{x}_{p(n),x}) \in (\mathbb{R}^d)^{p(n)}$ be a sequence of real vectors such that $\langle \mathbf{x}_{1,x}, \ldots, \mathbf{x}_{p(n),x} \rangle_{\text{index}} = x$. For each $b \in [\ell(n)]$, let $(\mathbf{x}_{p(n)+1,b}, \ldots, \mathbf{x}_{n,b}) \in (\mathbb{R}^d)^{n-p(n)}$ be a sequence of real vectors such that $\langle \mathbf{x}_{p(n)+1,b}, \ldots, \mathbf{x}_{n,b} \rangle_{\text{index}} = b$. Let $A = \{(\mathbf{x}_{p(n)+1,b}, \ldots, \mathbf{x}_{n,b}) \mid b \in [\ell(n)]\}$. We claim that $\mathcal{S}_{\mathsf{bit}^{(n)}_{p,\ell}, K}$ shatters this set A. For each $A' \subseteq A$, define $\langle A' \rangle_{\text{index}} \subseteq [\ell(n)]$ by $\langle A' \rangle_{\text{index}} := \{\langle \mathbf{x}_{p(n)+1,b}, \ldots, \mathbf{x}_{n,b} \rangle_{\text{index}} \mid (\mathbf{x}_{p(n)+1,b}, \ldots, \mathbf{x}_{p(n),b}) \in A'\}$. For each A', let $x_{A'}$ be a real number that has a 1 at every bit in $\langle A' \rangle_{\text{index}}$ and a 0 at every bit in $[\ell(n)] \setminus \langle A' \rangle_{\text{index}}$. We have $(\mathbf{x}_{1,x_{A'}}, \ldots, \mathbf{x}_{p(n),x_{A'}}, \mathbf{x}_{p(n)+1,b}, \ldots, \mathbf{x}_{n,b}) \in \mathsf{bit}^{(n)}_{p,\ell}$ if and only if $b \in A'$, as needed. □

It follows from Thm. 3.4 that no transformer can solve the p-partitioned ℓ-bit index task if $\ell(n) \in \omega(n^4(p(n))^2)$. That is, for any trained transformer $T_{\bar{\theta}}$ and function $\ell(n) \in \omega(n^4(p(n))^2)$, there is an N such that, for all $n \geq N$, $T_{\bar{\theta}}$ does not solve the $\ell(n)$-bit index task for strings of length n.

5.2 STATIC BIT INDEXING

Indeed, we show that, if the real number x is input directly in the first position, then no transformer can index the $b(n)$-th bit of x when $b(n) \in \omega(n^4)$ even if b is "hard-coded" into the transformer instead of being taken as part of the input.

Definition 5.3. Let $b\colon \mathbb{N} \to \mathbb{N}$ be any function mapping sequence lengths to indices. In the *hard-coded $b(n)$-th-bit index task*, the input is a sequence of real numbers $x_1, \ldots, x_n$, where $0 \leq x_1 < 1$. The intended output is the $b(n)$-th bit of x_1.

Proposition 5.4. *Let $b(n) \in \omega(n^4)$. For any trained transformer $T_{\bar{\theta}}$, there exists a length N such that for all $n \geq N$, $T^{(n)}_{\bar{\theta}}$ does not solve the hard-coded $b(n)$-th-bit index task for length n.*

Proof sketch. See App. E for the full proof.

Our proof uses similar ideas to our proof of Thm. 3.1 but has a fundamentally different structure. In this proof, we do not use the concept of VC dimension and instead directly use Thm. 2.5 to prove that, for any fixed inputs v, the following set has $2^{O(n^4)}$ connected components:

$$S^{(n)}_{\bar{\theta},v} := \{x \in [0, 1) \mid T^{(n)}_{\bar{\theta}}(x, v) > 0\}.$$

Importantly, we cannot simply assume that $S^{(n)}_{\bar{\theta},v}$ is a $(k-r)$-dimensional submanifold of $\mathbb{R}^k$. In fact, it is not. In order to apply Thm. 2.5, we construct a set that is a $(k - r)$-dimensional submanifold of

$\mathbb{R}^k$ and has no fewer connected components than $S^{(n)}_{\bar{\theta},v}$. The essential tools used in this construction are the regular level set theorem and Sard's theorem (Thms. B.4 and B.6), and our proof depends crucially on the fact that the real number x is entered directly as a single coordinate of the input. □

5.3 Discussion

Even though real-valued transformers have an infinite number of bits at their disposal, and can make use of this infinite precision in some cases (Prop. D.3), the results in this section show that they can make only limited use of this precision.

These results rule out the use of real numbers as "data structures" that can store an unbounded or infinite amount of information. For example, Merrill et al. (2022, Theorem 1) argue that an average-hard attention transformer with rational values can decide any language, by encoding an entire string of n bits into a single rational number and using a position-wise FFNN to decide whether the string belongs to the language. But by Thm. 3.1, transformers cannot compute all such functions, as this family of functions has VC dimension 2^n.

Although our results apply to a wide range of transformer variants, they only apply to transformers defined using a certain set of operations. Not included among these operations is rounding. So a possible criticism is that real-world transformers (or the hardware they run on) do in fact use rounding, raising the possibility that they could exploit rounding to solve problems that idealized, real-valued transformers cannot. However, the fact that language models are routinely trained at higher precision and then quantized to lower precision without catastrophic loss of accuracy (e.g., Frantar et al., 2023) shows that real-world transformers do not exploit rounding in this way.

Another possible criticism of the inexpressibility results proven in this section is that each of our bit indexing tasks, when regarded as a language, requires an alphabet whose size grows unboundedly with $\ell(n)$. Expanding on a footnote of Hahn (2020), Chen et al. (2025) prove that any symmetric circuit of depth L can be simulated by a transformer of depth $6L$. Consequently, if we could obtain a sufficiently strong transformer lower bound (an inexpressibility result, in our terminology) for a task over a fixed finite alphabet, their simulation would yield a corresponding lower bound for symmetric circuits. Obtaining such a lower bound would constitute a breakthrough in circuit complexity: in particular, no superlinear wire lower bound is known for general symmetric circuits of depth 3 or greater (Chen et al., 2025).

6 Positional indexing

Next, we consider the task of selecting the k-th *position* of the sequence (as opposed to the k-th bit, as above). Using this task, we get an $\Omega(n)$ lower bound on the split VC dimension of transformers.

Definition 6.1. In the *index task*, which we write as ind_n, the input is $x_1 \cdots x_{n-1} k$, where $x_1, \ldots, x_{n-1} \in \{0, 1\}$ and $k \in [n-1]$. The intended output is the value of the bit x_k.

Proposition 6.2. split-VC$(\mathsf{ind}_n) \geq n - 1$.

Proof. Let $K = \{n\}$ be the set of positions to treat as parameters, so we want to show $\mathrm{VC}(\mathcal{S}_{\mathsf{ind}_n, K}) \geq n - 1$. Let $A = [n-1]$ be the set to be shattered. For any subset $A' \subseteq A$, let $w_{i,A'} = 1$ if and only if $i \in A'$. We have $\mathsf{ind}_n(w_{1,A'}, \ldots, w_{n-1,A'}, k) = 1$ if and only if $k \in A'$. It follows that $\mathrm{VC}(\mathcal{S}_{\mathsf{ind}_n, K}) \geq n - 1$, as needed. □

Proposition 6.3. *The index task is recognizable by a one-layer transformer.*

Proof. We set up the word and position embeddings so that

$$X_{i,*} = \begin{bmatrix} x_i & 0 & i & i^2 \end{bmatrix}^\top \quad (i \in [n-1]) \qquad X_{n,*} = \begin{bmatrix} 0 & k & n & n^2 \end{bmatrix}^\top$$

The first self-attention layer computes queries, keys, and values

$$W^{\mathrm{Q}} X_{n,*} = \begin{bmatrix} 4k\gamma & -2\gamma \end{bmatrix}^\top \qquad W^{\mathrm{K}} X_{j,*} = \begin{bmatrix} j & j^2 \end{bmatrix}^\top \qquad W^{\mathrm{V}} X_{j,*} = \begin{bmatrix} x_j \end{bmatrix}.$$

The attention scores at the last position are $s_{nj} = 4\gamma k j - 2\gamma j^2$. So position k has the highest score and decreases by at least γ for each position in either direction. Then, as in the proof of Prop. D.1, we can make the attention output positive iff $x_k = 1$. □

This result may appear to contradict Theorem 3.4 of Kozachinskiy et al. (2025), which states that no one-layer transformer can solve the index task. This apparent contradiction is resolved by observing out that the inexpressibility result of Kozachinskiy et al. (2025) applies to a specific version of the index task where the output is computed at a blank token appended to the input.

7 Generalization bounds

Our VC dimension bounds imply generalization bounds for multi-layer, real-valued SMATs under very weak restrictions. Using Thm. 3.1, we can bound the generalization error of such a transformer by $\tilde{O}(n^2/\sqrt{N})$, where N is the training data size (Shalev-Shwartz & Ben-David, 2014). Yang et al. (2026b) give a logarithmic upper-bound on the VC dimension of AHATs (which crucially do not use $\exp$), implying a generalization bound of $\tilde{O}(\sqrt{\log n/N})$.

Edelman et al. (2022) used a covering number bound to derive the first generalization bound for transformers, of $\tilde{O}(\sqrt{\mathsf{poly}(C)\log(n)/N})$, where C is a function of the norms of the attention and FFNN matrices. Trauger & Tewari (2024) also derived a covering number generalization bound, independent of n. For Edelman et al. (2022), the bound for a single attention head would scale as $\tilde{O}(\sqrt{C_X^2\mathsf{poly}(C)\log(n)/N})$, where $\|X\|_{2,\infty} < C_X$ bounds the norms of the inputs. The bounds of Trauger & Tewari (2024) also depend on C_X. Thus, when C_X grows sufficiently fast with n, our bound is asymptotically stronger than these bounds. Li et al. (2026) proved generalization bounds for cases where the input norms are unbounded, but they still make assumptions about the distributions the inputs are drawn from. Our bound is asymptotically stronger than theirs when the parameters of the input distributions grow sufficiently fast with n or when no such assumptions can be made.

8 Future research

Thm. 3.1 and Cor. 3.5 give upper bounds on the VC dimension of an (untrained) transformer and the split VC dimension of a (trained) transformer, respectively, and Props. D.2 and 6.3 give lower bounds on both. These upper and lower bounds are listed in Tab. 1. There are clearly gaps between our upper and lower bounds. While it would certainly be desirable to close these gaps, it is worth noting that Koiran & Sontag (1998, Sec. 5) leave it as an open problem to close similar gaps for RNNs, and their problem is still open to the best of our knowledge.

Another direction for further research is to explore the extent to which our theoretical results manifest in practice. Experiments regarding the abilities of transformers to learn symmetric or indexing functions or to access specific bits of real numbers in other ways would shed light on the practical implications of our results.

Use of artificial intelligence

We used artificial intelligence to review literature and polish writing. We did not use artificial intelligence to generate proofs.

References

Lijie Chen, Binghui Peng, and Hongxun Wu. Theoretical limitations of multi-layer transformer. In *IEEE 66th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 2631–2653, 2025. doi:10.1109/FOCS63196.2025.00136.

David Chiang and Peter Cholak. Overcoming a theoretical limitation of self-attention. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 7654–7664, May 2022. doi:10.18653/v1/2022.acl-long.527.

Benjamin L. Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5793–5831, 2022. URL https://proceedings.mlr.press/v162/edelman22a.html.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=tcbBPnfwxS.

Paul W. Goldberg and Mark R. Jerrum. Bounding the Vapnik–Chervonenkis dimension of concept classes parameterized by real numbers. *Machine Learning*, 18(2):131–148, 1995. doi:10.1007/BF00993408.

Michael Hahn. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171, 2020. doi:10.1162/tacl_a_00306.

Marek Karpinski and Angus Macintyre. Polynomial bounds for VC dimension of sigmoidal and general Pfaffian neural networks. *Journal of Computer and System Sciences*, 54(1):169–176, 1997. doi:https://doi.org/10.1006/jcss.1997.1477.

A. G. Khovanskiĭ. *Fewnomials*, volume 88 of *Translations of Mathematical Monographs*. American Mathematical Society, 1991. doi:10.1090/mmono/088. Translated from the Russian by Smilka Zdravkovska.

Pascal Koiran and Eduardo D. Sontag. Vapnik–Chervonenkis dimension of recurrent neural networks. *Discrete Applied Mathematics*, 86(1):63–79, 1998. doi:https://doi.org/10.1016/S0166-218X(98)00014-6.

Alexander Kozachinskiy, Felipe Urrutia, Hector Orellana, Tomasz Steifer, Germán Pizarro, Matías Fuentes, Francisco Meza Vaśquez, Cristian Buc Calderon, and Cristobal Rojas. Strassen attention, split VC dimension and compositionality in transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 9665–9696, 2025. doi:10.52202/085713-0327.

Alexander Kozachinskiy, Vicente Opazo, and Felipe Urrutia. Indexing: the beginning and the end, 2026. URL https://arxiv.org/abs/2607.22361.

Yawen Li, Tao Hu, Zhouhui Lian, Wan Tian, Yijie Peng, Huiming Zhang, and Zhongyi Li. Sharper generalization bounds for transformer, 2026. URL https://arxiv.org/abs/2603.21541.

William Merrill and Ashish Sabharwal. A logic for expressing log-precision transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 52453–52463, 2023. doi:10.52202/075280-2284.

William Merrill, Ashish Sabharwal, and Noah A. Smith. Saturated transformers are constant-depth threshold circuits. *Transactions of the Association for Computational Linguistics*, 10:843–856, 2022. doi:10.1162/tacl_a_00493.

Jorge Pérez, Pablo Barceló, and Javier Marinkovic. Attention is Turing complete. *Journal of Machine Learning Research*, 22(75):1–35, 2021. URL https://jmlr.org/papers/v22/20-302.html.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. doi:10.1017/CBO9781107298019.

Hava T. Siegelmann and Eduardo D. Sontag. Analog computation via neural networks. *Theoretical Computer Science*, 131(2):331–360, 1994. doi:10.1016/0304-3975(94)90178-3.

Richard P. Stanley. *Enumerative combinatorics. Volume 1*, volume 49 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, second edition, 2012. ISBN 978-1-107-60262-5.

Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. What formal languages can transformers express? A survey. *Transactions of the Association for Computational Linguistics*, 12:543–561, 2024. doi:10.1162/tacl_a_00663.

Jacob Trauger and Ambuj Tewari. Sequence length independent norm-based generalization bounds for transformers. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 238 of *Proceedings of Machine Learning Research*, pp. 1405–1413, 2024. URL https://proceedings.mlr.press/v238/trauger24a.html.

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 (NIPS)*, volume 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Hugh E. Warren. Lower bounds for approximation by nonlinear manifolds. *Transactions of the American Mathematical Society*, 133:167–178, 1968. doi:10.2307/1994937.

Andy Yang, Lena Strobl, David Chiang, and Dana Angluin. Simulating hard attention using soft attention. *Transactions of the Association for Computational Linguistics*, 14:147–166, 2026a. doi:10.1162/TACL.a.597.

Chenxiao Yang, Nathan Srebro, and Zhiyuan Li. Tight sample complexity of transformers. In *Proceedings of Thirty Ninth Conference on Learning Theory (COLT)*, volume 336 of *Proceedings of Machine Learning Research*, pp. 6887–6923, 2026b. URL https://proceedings.mlr.press/v336/yang26a.html.

A Definition of Transformers

We follow, for the most part, notational conventions used by Strobl et al. (2024).

Definition A.1 (Vaswani et al., 2017)**.** Fix an input–output dimension $d \in \mathbb{N}$.

An *attention head* is a family of functions, indexed by input sequence length n, of the form

$$\begin{aligned} \mathcal{H}^{(n)} \colon \mathbb{R}^{n\times d} &\to \mathbb{R}^{n\times d} \\ s_{i,j} &= \frac{1}{\sqrt{d}}(W^{\mathrm{Q}} X_{i,*}) \cdot (W^{\mathrm{K}} X_{j,*}) \\ \alpha_{i,j} &= [\mathrm{softmax} s_{i,*}]_j \\ &= \frac{\exp s_{i,j}}{\sum_{j' \in [n]} \exp s_{i,j'}} \qquad (1) \\ [\mathcal{H}^{(n)}(X)]_i &= \sum_{j \in [n]} \alpha_{i,j} (W^{\mathrm{V}} X_{j,*}) \qquad (2) \end{aligned}$$

with parameters $W^{\mathrm{Q}} \in \mathbb{R}^{d\times d}$, $W^{\mathrm{K}} \in \mathbb{R}^{d\times d}$, and $W^{\mathrm{V}} \in \mathbb{R}^{d\times d}$. In many cases, $\mathcal{H}$ is *masked* so that $[\mathcal{H}(X)]_i$ only depends on X_j for $j \le i$. The presence or absence of masking makes no difference to our results.)

An *attention sublayer* is a family of functions of the form

$$\begin{aligned} \mathcal{A}^{(n)} \colon \mathbb{R}^{n\times d} &\to \mathbb{R}^{n\times d} \\ X &\mapsto \sum_{h=1}^{H} \mathcal{H}_h^{(n)}(X) \qquad (3) \end{aligned}$$

with some number of heads H and an attention head $\mathcal{H}_h$ for each $h \in [H]$.

A *layer normalization* is a function

$$\begin{aligned} \mathcal{N} \colon \mathbb{R}^{d} &\to \mathbb{R}^{d} \\ \mathbf{x} &\mapsto \gamma \odot \frac{\mathbf{x} - \overline{\mathbf{x}}}{\sqrt{\mathrm{var}(\mathbf{x}) + \varepsilon}} + \beta, \qquad (4) \end{aligned}$$

with some numerical stabilizer $\varepsilon > 0$ and parameters $\gamma \in \mathbb{R}^d$ and $\beta \in \mathbb{R}^d$. Here, $\odot$ denotes element-wise multiplication, $\overline{\mathbf{x}}$ denotes the mean of $\mathbf{x}$ ($\overline{\mathbf{x}} := \frac{1}{d}\sum_{i\in[d]} \mathbf{x}_i$) and $\mathrm{var}(\mathbf{x})$ denotes the variance of $\mathbf{x}$ ($\mathrm{var}(\mathbf{x}) := \frac{1}{d}\sum_{i\in[d]} (\mathbf{x}_i - \overline{\mathbf{x}})^2$).

A *position-wise feed-forward network* is a function of the form

$$\begin{aligned} \mathcal{F} \colon \mathbb{R}^{d} &\to \mathbb{R}^{d} \\ \mathbf{x} &\mapsto W_2(\max(0, W_1\mathbf{x} + \mathbf{b}_1)) + \mathbf{b}_2, \qquad (5) \end{aligned}$$

with parameters $W_1 \in \mathbb{R}^{d\times d}$, $\mathbf{b}_1 \in \mathbb{R}^d$, $W_2 \in \mathbb{R}^{d\times d}$, and $\mathbf{b}_2 \in \mathbb{R}^d$. The $\max$ is taken element-wise.

Both $\mathcal{N}$ and $\mathcal{F}$ can be extended position-wise to functions from $\mathbb{R}^{n\times d}$ to $\mathbb{R}^{n\times d}$.

A *transformer layer* is a family of functions of the form

$$\begin{aligned} \mathcal{L}^{(n)} \colon \mathbb{R}^{n\times d} &\to \mathbb{R}^{n\times d} \\ X &\mapsto \mathcal{N}_2\Big(\mathcal{N}_1\big(X + \mathcal{A}^{(n)}(X)\big) + \mathcal{F}\Big(\mathcal{N}_1\big(X + \mathcal{A}^{(n)}(X)\big)\Big)\Big), \end{aligned}$$

where $\mathcal{A}^{(n)}$ is an attention sub-layer, $\mathcal{F}$ is a position-wise feed-forward network, and $\mathcal{N}_1$ and $\mathcal{N}_2$ are layer normalizations. We have shown the original definition, which places layer normalizations after their respective sub-layers. Other orderings are possible, and would make no difference to our results.

In this paper, we treat transformers as binary classifiers. So an *output layer* is a family of functions of the form

$$\begin{aligned} \mathcal{O}^{(n)} \colon \mathbb{R}^{n\times d} &\to \mathbb{R} \\ X &\mapsto W X_{n,*} + b \end{aligned}$$

with parameters $W \in \mathbb{R}^{1\times d}$ and $b \in \mathbb{R}$.

A *transformer* is a family of functions of the form

$$T^{(n)} \colon \mathbb{R}^{n\times d} \to \mathbb{R}$$
$$X \mapsto \mathcal{O}^{(n)}(\mathcal{L}_L^{(n)}(\dots(\mathcal{L}_1^{(n)}(X))\dots)),$$

with some depth L, where $\mathcal{L}_1^{(n)}, \dots, \mathcal{L}_L^{(n)}$ are transformer layers, and $\mathcal{O}^{(n)}$ is an output layer.

Definition A.2. Given an alphabet Σ, a *word embedding* is a function $\mathrm{WE} \colon \Sigma \to \mathbb{R}^d$. A *position embedding* is a family of functions $\mathrm{PE}^{(n)} \colon [n] \to \mathbb{R}^d$. An *embedding* is a family of functions of the form

$$e^{(n)} \colon \Sigma^n \to (\mathbb{R}^d)^n$$
$$e^{(n)}(w)_i := \mathrm{WE}(w_i) + \mathrm{PE}^{(n)}(i),$$

where $\mathrm{WE} \colon \Sigma \to \mathbb{R}^d$ is a word embedding and $\mathrm{PE}^{(n)} \colon [n] \to \mathbb{R}^d$ is a position embedding. A *discrete-input transformer* is a family of functions of the form

$$\mathcal{T}^{(n)} \colon \Sigma^n \to \mathbb{R}$$
$$w \mapsto T^{(n)}(e^{(n)}(w)),$$

where T is a (real-input) transformer and e is an embedding.

B Background on submanifolds of $\mathbb{R}^n$

Definition B.1 (Some basic topology of $\mathbb{R}^n$)**.** For $r > 0$ and $\mathbf{p} \in \mathbb{R}^n$, define the *open ball of radius r and center* $\mathbf{p}$ to be the set $B_r(\mathbf{p}) := \{\mathbf{x} \in \mathbb{R}^n : ||\mathbf{x} - \mathbf{p}|| < r\}$. A set $U \subseteq \mathbb{R}^n$ is called *open (in $\mathbb{R}^n$)* if, for any $\mathbf{p} \in U$, there exists $r > 0$ such that $B_r(\mathbf{p}) \subseteq U$. Given a subset $X \subseteq \mathbb{R}^n$, a set $U \subseteq X$ is called *open in X* if, for any $\mathbf{p} \in U$, there exists $r > 0$ such that $B_r(\mathbf{p}) \cap X \subseteq U$. When there is no risk of ambiguity, we refer to sets that are open in U as *open subsets of U*. A set $X \subseteq \mathbb{R}^n$ is called *connected* if it cannot be expressed as the disjoint union of two non-empty open subsets of itself. A *connected component of X* is a maximal connected subset of X (i.e., a connected subset $C \subseteq X$ such that, for any connected subset $C \subseteq C' \subseteq X$, we have $C = C'$).

Intuitively, a k-dimensional submanifold of $\mathbb{R}^n$ is a subset of $\mathbb{R}^n$ that locally resembles $\mathbb{R}^k$.

Definition B.2 (Diffeomorphisms and submanifolds of $\mathbb{R}^n$)**.** Given an open set $U \subseteq \mathbb{R}^n$, a function $f : U \to \mathbb{R}^n$ is called C^m if all of its m-th derivatives exist and are continuous, and is called C^∞ (or *smooth*) if it is C^m for every m. Given an open subset $U \subseteq \mathbb{R}^n$, a function $f : U \to \mathbb{R}^n$ is called a *diffeomorphism* if f and f^{-1} are both C^∞, f is injective, and $f(U)$ is open in $\mathbb{R}^k$. If $k \leq n$, define $\mathbb{R}^k \times \mathbf{0} := \{(x_1, \dots, x_n) \in \mathbb{R}^n \mid x_i = 0 \text{ for all } k < i \leq n\}$. A set X is called a *k-dimensional submanifold of $\mathbb{R}^n$* if, for every $\mathbf{p} \in X$, there exists an open $U \subseteq \mathbb{R}^n$ with $\mathbf{p} \in U$ and a diffeomorphism $f : U \to \mathbb{R}^n$ such that $f(X \cap U) = \mathbb{R}^k \times \mathbf{0} \cap f(U)$.

Definition B.3 (Regular and critical values)**.** Let $U \subseteq \mathbb{R}^n$ be an open set, and let $f : U \to \mathbb{R}^m$ be a smooth function. Let df denote the differential of f, that is, the matrix $\left[\frac{\partial f}{\partial x^1} \quad \cdots \quad \frac{\partial f}{\partial x^n}\right]$, where x^i is the ith coordinate of $\mathbb{R}^n$ and each partial derivative is regarded as a column of length m. Given a point $\mathbf{x} \in U$, the evaluated differential $df(\mathbf{x})$ is identified with the linear function from $\mathbb{R}^n$ to $\mathbb{R}^m$ that it induces via matrix–vector multiplication. A point $\mathbf{x} \in U$ is called a *regular point of f* if the map $df(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$ is surjective, and called a *critical point of f* otherwise. A point $\mathbf{y} \in \mathbb{R}^m$ is called a *regular value of f* if every point in $f^{-1}(\mathbf{y})$ is regular, and called a *critical value of f* otherwise. For any $\mathbf{y} \in \mathbb{R}^m$, the set $f^{-1}(\mathbf{y})$ is called a *level set of f*. A level set $f^{-1}(\mathbf{y})$ is called a *regular level set of f* if $\mathbf{y}$ is a regular value of f.

We state the following two theorems only for our special case of submanifolds of $\mathbb{R}^n$, but their natural generalizations hold for arbitrary smooth manifolds.

Theorem B.4 (Regular level set theorem)**.** *Let X be k-dimensional submanifold of $\mathbb{R}^n$, let Y be a j-dimensional submanifold of $\mathbb{R}^m$, and $f : X \to Y$ be a smooth function. Every regular level set of f is a $(k - j)$-dimensional submanifold of $\mathbb{R}^m$.*

Definition B.5 (Measure in a submanifold)**.** A set $A \subseteq \mathbb{R}^k$ has *(k-dimensional) measure zero* if, for any $\delta > 0$, A can be covered by a countable collection of k-dimensional hyperrectangles the sum of whose volumes is less than δ. Let X be a k-dimensional submanifold of $\mathbb{R}^n$, and let $A \subseteq X$. We say that A has *(k-dimensional) measure zero in X* if, for any open $U \subseteq \mathbb{R}^n$ and any diffeomorphism $f : U \to \mathbb{R}^n$ such that $f(X \cap U) = \mathbb{R}^k \times \mathbf{0} \cap f(U)$, the image $f(A \cap U)$ has (k-dimensional) measure zero when regarded as a subset of $\mathbb{R}^k$.

Theorem B.6 (Sard's theorem)**.** *Let X be k-dimensional submanifold of $\mathbb{R}^n$, let Y be a j-dimensional submanifold of $\mathbb{R}^m$, and let $f : X \to Y$ be a smooth function. The set of critical values of f has measure zero in Y.*

C Proof of Thm. 3.1

When Karpinski & Macintyre give their VC dimension bound for FFNNs, they apply Thm. 2.3 directly to a sigmoid FFNN as a single function f (so $s = 1$). This is possible because a sigmoid FFNN is a C^∞ function. The value B is computed by appeal to Thm. 2.5, but to use this theorem a rather subtle argument is used, adding new variables to obtain a suitable system of exponential–linear polynomial equalities and inequalities.

Let T be our given (untrained) transformer, with d the dimension of the vectors, and L the number of layers, and H the number of heads. We let $\bar{y}$ be the tuple of parameters. We fix an input length n. So the input is in $\mathbb{R}^{nd}$. Let us write $f(\bar{x}, \bar{y})$ for the function computed by $T^{(n)}$, on input $\bar{x}$ and with parameters $\bar{y}$. We will write f explicitly as a composition of the various functions in Def. A.1 over the L layers. So we could think of $f(\bar{x}, \bar{y})$ as a "term" and the VC-dimension of $T^{(n)}$ is the VC dimension of the expression $f(\bar{x}, \bar{y}) > 0$.

But f is *not* a C^∞-function, due to the appearance of $\max(-,-)$ in the position-wise feed-forward network $\mathcal{F}$ (in each layer). This is actually the only obstruction, as the various rational expressions always have nonzero denominator and the square root in layer normalization is applied to a positive real number bounded away from 0.

So what we will do in the first step is replace the "formula" $f(\bar{x}, \bar{y}) > 0$ by an equivalent formula $\phi(\bar{x}, \bar{y})$ involving a certain number of inequalities $f_\eta(\bar{x}, \bar{y}) > 0$ built by making "substitutions" in $f > 0$, and an additional finite number of inequalities where all the functions involved, including the f_η, are C^∞. So we can apply Thm. 2.3 to $\phi(\bar{x}, \bar{y})$.

We define the f_η and the additional inequalities inductively. As mentioned above, f is a composition of terms representing the various stages. Consider the first time $\max$ appears in the first layer. Remember $\max$ is the pointwise maximum of 0 and a certain d-dimensional column vector, and n of these appear. So inside f, we have $\max(0, \tau_{i,j})$ for $i = 1, \ldots, n$, $j = 1, \ldots, d$, and $\tau_{i,j}$ are terms built so far and stand for C^∞ functions. Let $\eta \in \{0,1\}^{[n]\times[d]}$. Let f_η be the result of replacing each occurrence of $\max(0, \tau_{i,j})$ in f by $\tau_{i,j}$ if $\eta(i,j) = 1$, and by 0 if $\eta(i,j) = 0$. Let g_η be the conjunction of $f_\eta > 0$ with the set of inequalities $\tau_{i,j} \geq 0$ whenever $\eta(i,j) = 1$ and $\tau_{i,j} < 0$ whenever $\eta(i,j) = 0$ (for $i \in [n]$ and $j \in [d]$). And let $\phi_1(\bar{x}, \bar{y})$ be the disjunction of the g_η for $\eta \in \{0,1\}^{[n]\times[d]}$, which is equivalent to $\phi(\bar{x}, \bar{y})$ but has fewer occurrences of $\max(0, -)$. Now iterate this process through the layers. Let us briefly describe what to do at layer 2. Fix η, and consider again where the $\max$ from layer 2 appears in f_η. And again we will have inside f_η, $\max(0, \pi_{i,j})$ for $(i,j) \in nd$. Choose $\nu \in \{0,1\}^{[n]\times[d]}$, and let $f_{(\eta,\nu)}$ result from f_η by replacing $\max(0, \pi_{i,j})$ by $\pi(i.j)$ if $\nu(i,j) = 1$ and by 0 if $\nu(i,j) = 0$. And let $g_{(\eta,\nu)}$ consist of the conjunction of $f_{(\eta,\nu)} > 0$ with the rest of the inequalities in g_η and the inequalities $\pi_{i,j} \geq 0$ whenever $\nu(i.j) = 1$ and $\pi_{i,j} < 0$ whenever $\nu(i,j) < 0$. Let $\phi_2(\bar{x}, \bar{y})$ be the disjunction of the $g_{(\eta,\nu)}$ as η, ν vary. Continue in the same way through the layers and let $\phi(\bar{x}, \bar{y}) = \phi_L(\bar{x}, \bar{y})$. Then notice that ϕ is the disjunction of $(2^{nd})^L$ expressions g_α say, each consisting of the conjunction of some $f_\alpha > 0$ (arising from f by making the appropriate substitutions) and a further Lnd inequalities, where all the functions (or terms) appearing are C^∞ (in $(\bar{x}, \bar{y})$). So $\phi(\bar{x}, \bar{y})$ is a Boolean combination of (in fact a disjunction of conjunctions of) $s = (2^{nd})^L(Lnd + 1)$ inequalities $\tau(\bar{x}, \bar{y}) > 0$, $\tau(\bar{x}, \bar{y}) \geq 0$, for τ a C^∞ function. Let us write $\{\tau_i \mid i \in [s]\}$ for these τ appearing in ϕ.

So we can apply Thm. 2.3 and it remains to find the bound B from there. Let k be the dimension of the parameter space $\bar{y}$. Let $\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_V$ be nd-tuples. Fix $r \leq k$ and let $F_1, \ldots, F_r \colon \mathbb{R}^k \to \mathbb{R}$ be

r many p-functions from among the $\tau_i(\bar{a}_j, \bar{y})$. Let $\epsilon_1, \ldots, \epsilon_r \in \mathbb{R}$ and suppose that the preimage of $(\epsilon_1, \ldots, \epsilon_r)$ under $(F_1, \ldots, F_r)$ is a $(k-r)$-dimension C^∞-submanifold of $\mathbb{R}^k$. We want to bound the number of connected components of this submanifold (independent of the choices of the data $\bar{a}_j$).

We now add some new variables and defining equations so as to be able to apply Thm. 2.5. Now for each $\tau(\bar{x}, \bar{y})$ such that for some $\bar{a}_j$, $\tau(\bar{a}_j, \bar{y})$ is among the $F_i(\bar{y})$, τ represents all or some initial part of a run of the transformer (with input $\bar{x}$, parameters $\bar{y}$). Consider the first time that softmax appears in some such τ (so in the first layer).

We consider the softmax expression (Eq. (1)) as a function of the input variables $\bar{x} \in \mathbb{R}^{nd}$ and parameters $\bar{y}$. For $1 \leq i, j \leq n$, add new variables $z^{\mathrm{s}}_{i.j}$ and $z^{\mathrm{a}}_{i,j}$, replace $\alpha_{i,j}$ in Eq. (2) with new variables $z^{\mathrm{a}}_{i,j}$, and add conditions $z^{\mathrm{s}}_{i,j} = s_{i,j}$ (where note the latter is a linear polynomial in $\bar{x}$ and $\bar{y}$) as well as:

$$z^{\mathrm{a}}_{i,j} \left(\sum_{\ell} \exp z^{\mathrm{s}}_{i,\ell} \right) = \exp z^{\mathrm{s}}_{i,j}.$$

Now consider the first time that layer normalization appears in some τ. This involves taking square roots of terms $v_i(\bar{x}, \bar{y})$ which already appear in τ and whose value is positive and bounded away from 0. Add new variables z^{s}_i, and equations $(z^{\mathrm{s}}_i)^4 = v_i$, and replace the square root term in τ by $(z^{\mathrm{s}}_i)^2$. Now we have to deal with the division in layer normalization. For a given i the expression in the numerator is a column vector with coordinates polynomials, say $w_1, \ldots, w_d$ in $\bar{x}$, $\bar{y}$ and new variables added in the softmax stage. Add new variables $z^{\mathrm{n}}_{i,j}$ for $j = 1, \ldots, d$ and equations $z^{\mathrm{n}}_{i,j}(z^{\mathrm{s}}_i)^2 = w_j$. And substitute the column vector with coordinates $z^{\mathrm{n}}_{i,j}$ for the rational expression in layer normalization.

Continue the process inductively and cumulatively through the layers and for the relevant τ.

For each $i \in [r]$, let $F'_i(\bar{y}, \bar{z}_i)$ be the system of equations resulting from the above rewriting process on $F_i(\bar{y})$, with all the variables in $\bar{z}_i$ renamed to fresh names. Let $S(\bar{y}, \bar{z}_1, \ldots, \bar{z}_r)$ be the union of the $F'_i(\bar{y}, \bar{z}_i)$ for $i \in [r]$. Let κ be the total number of variables in $\bar{z}_i$ for $i \in [r]$. So the number of variables in the system S is $k + \kappa$ and the number of equations is $r + \kappa$.

As in the proof of Karpinski & Macintyre (1997, p. 173–174) we have the following claims:

(i) Suppose $S(\bar{y}, \bar{z}_{F_1}, \ldots, \bar{z}_{F_r})$, then $F_i(\bar{y}) = \epsilon_i$ for $i = 1, \ldots, r$.
(ii) Suppose $F_i(\bar{y}) = \epsilon_i$ for $i = 1, \ldots, r$ then there are unique $\bar{z}_{F_i}$ for $i = 1, \ldots, r$ such that $S(\bar{y}, \bar{z}_{F_1}, \ldots, \bar{z}_{F_r})$.
(iii) The set in $\mathbb{R}^k$ defined by $F_i(\bar{y}) = \epsilon_i$ for $i = 1, \ldots, r$ is homeomorphic to the set in $\mathbb{R}^{k+\kappa}$ defined by $S(\bar{y}, \bar{z}_{F_1}, \ldots, \bar{z}_{F_r})$.

By the claims above and our assumptions on the F_i, the subset of $\mathbb{R}^{k+\kappa}$ defined by S is a manifold of dimension $k - r$. On the other hand S consists of $r + \kappa$ equations in $k + \kappa$ variables, and $(k + \kappa) - (r + \kappa) = k - r$.

Since the $\tau_i(\bar{a}_j, \bar{y})$ are exponential–linear polynomials, Thm. 2.5 applies, to give that the number of connected components of the set defined by S (so also by part (iii) of the Claim, defined by $F_1(\bar{y}) = \epsilon_1, \ldots, F_r(\bar{y}) = \epsilon_r$) is at most B where $\log B \in O(q^2 + M \log M)$, where q is the total number of exponentials appearing in S, and M is the maximum of $r + \kappa$, $k + \kappa$, and D (the maximum degree of polynomials in S). The only quantities in the expression for B that depend on the input are $\kappa \in O(n^2)$ and q. The transformer has $O(n^2)$ exponentials (Eq. (1)), but because the exponents $\Lambda_1, \ldots, \Lambda_q$ may depend on $\bar{x}$, and $\bar{x}$ may be substituted with up to r different points $\bar{a}_i$, the number of exponentials must be multiplied by $r \leq k$. But k does not depend on n, so we still have $q \in O(n^2)$. Thus $\log B \in O(n^4)$. Hence, by Thm. 2.3, the VC dimension of T^n is at most $2 \log B + (16 + 2 \log s)k$ where s is the number of C^∞ functions appearing in ϕ above. As $\log s \in O(n)$ and k depends just on the untrained transformer T, the VC dimension is at most $O(n^4)$, concluding the proof of Thm. 3.1.

D TRANSFORMERS COMPUTING SYMMETRIC FUNCTIONS

Proposition D.1. *For any family of (symmetric) functions $f^{(n)}\colon \{\mathtt{1}\}^n \to \{0,1\}$, there exists a trained transformer $T_{\bar\theta}$ such that, for any n, $T_{\bar\theta}^{(n)}(\mathtt{1}^n) = f^{(n)}(\mathtt{1}^n)$.*

Proof. We don't need a word embedding, since there is only one symbol. We use a positional encoding $\mathrm{PE}(i) = (i, f^{(i)}(\mathtt{1}^i))$. The self-attention layer just has to retrieve the last coordinate of the positional encoding at the last position n. For each vector $X_{i,*}$, we set the weights such that:

$$W^{\mathrm{Q}}X_{i,*} = [\gamma] \qquad W^{\mathrm{K}}X_{j,*} = [j] \qquad W^{\mathrm{V}}X_{j,*} = \left[f^{(j)}(\mathtt{1}^j)\right].$$

At position $i = n$, the score s_{nj} is maximized for $j = n$, and the scores decrease by γ for each position to the left. By Lemma 6 of Yang et al. (2026a), there is a choice of γ that ensures that the attention on position n is greater than $3/4$, so we can make the attention output positive iff $f^{(n)}(\mathtt{1}^n) = 1$. If there is layer normalization, then we can use the technique of Chiang & Cholak (2022, Sec. 4.1) so that layer normalization does not change the sign of the attention output. □

Proposition D.2. *There exists a trained transformer $T_{\bar\theta}$ such that, for any $n > 0$ and symmetric function $f\colon \{\mathtt{0},\mathtt{1}\}^n \to \{0,1\}$, there exists a positional encoding $\mathrm{PE}\colon [n] \to \mathbb{R}^d$ such that $T_{\bar\theta}^{(n)}$ with positional encoding PE computes f.*

Proof. For each symmetric function $f\colon \{\mathtt{0},\mathtt{1}\}^n \to \{0,1\}$, there exists by definition a function $\bar f\colon [n] \to \{-1,1\}$ such that, for all (or, equivalently, any) string w that has exactly i many $\mathtt{1}$s, we have $\bar f(i) = 1$ if $f(w) = 1$ and $\bar f(i) = -1$ if $f(w) = 0$. (Since we are working over a binary alphabet with strings of a fixed length, the number of $\mathtt{1}$s determines the number of $\mathtt{0}$s.) To construct a transformer computing f, we follow verbatim the construction of Chiang & Cholak (2022) of a transformer computing the PARITY function (Section 3.2 of their paper), but with one change. Where they include $\cos(i\pi)$ (equivalently, $(-1)^i$) in the final coordinate of their positional encoding, we instead include $\bar f(i)$. For the same reason that their construction computes the PARITY function, ours computes f. All of the transformers constructed in this way use the exact same learned parameters—their only difference is that they use different positional encodings. □

Proposition D.3. *L is recognized by a transformer, where $\Omega = \sum_{x:U \text{ halts on } x} 2^{-|x|}$ and $L = \{w \in \{\mathtt{0},\mathtt{1}\}^* \mid \#_{\mathtt{1}}(w)/|w| < \Omega\}$.*

Proof. We construct a transformer with one layer. The self-attention attends uniformly ($W^{\mathrm{Q}} = W^{\mathrm{K}} = \mathbf{0}$). The values are $\mathbb{I}[w_i = \mathtt{1}]$. The FFNN computes $\mathit{ffn}(x) = \max(0, \Omega - x)$. The transformer accepts the string if $\mathit{ffn}(x) > 0$, which occurs if and only if $\#_{\mathtt{1}}(w)/|w| < \Omega$. □

E PROOF OF PROP. 5.4

The key argument is that extracting a deeper bit requires the output to alternate increasingly many times as x varies. We first bound how many connected components a fixed transformer can produce.

Lemma E.1. *Let T be an untrained transformer. For any parameter values $\bar\theta$, length n, and sequence $v \in \mathbb{R}^{(n-1)}$, the set*

$$S_{\bar\theta,v}^{(n)} = \left\{x \in [0,1) : T_{\bar\theta}^{(n)}(x, v) > 0\right\}.$$

has $2^{O(n^4)}$ connected components.

Proof. Fix n, $\bar\theta$, and v. We want to apply Thm. 2.5 directly to bound the number of connected components of $S_{\bar\theta,v}^{(n)}$, so we need to rewrite the transformer in a suitable form.

First, we need to eliminate square roots and divisions, and to ensure that the argument of every $\exp$ is linear, just as in the proof of Thm. 3.1. Second, we need to eliminate $\max$, and third, we need to replace inequalities with equalities.

For each ReLU $z = \max(0, h)$, use the equivalent conjunction

$$z \geq 0, \qquad z - h \geq 0, \qquad z(z - h) = 0.$$

Then for each inequality, introduce a new real variable u and use

$$p \geq 0 \iff (\exists u)\; p - u^2 = 0.$$

The resulting system is logically equivalent to the original transformer computation.

Let $\bar{z}$ denote the tuple consisting of all variables. We obtain exponential–linear polynomials $P_1(x, \bar{z}), \ldots, P_s(x, \bar{z})$ such that

$$T_{\bar{\theta}}^{(n)}(x, v) > 0 \quad \Longleftrightarrow \quad (\exists \bar{z}) \bigwedge_{i=1}^{s} P_i(x, \bar{z}) = 0. \tag{6}$$

For a fixed transformer, the above representation has $O(n^2)$ computation variables and constraints, $O(n^2)$ exponential terms, and polynomial degrees bounded by $O(1)$. The replacement of inequalities with equations adds $O(n^2)$ variables and equations, without introducing new exponential terms. Thus, let $\kappa = |\bar{z}|$, let r, q, and D denote respectively the number of equations, the number of exponential terms, and the maximum degree of the exponential–linear polynomials P_i. Then we have

$$\kappa, r, q \in O(n^2), \qquad D \in O(1).$$

To apply Thm. 2.5, the common solution set in Eq. (6) must be a smooth manifold, but this may not be the case. For example, the equations used above to encode a ReLU relation $z = \max(0, h)$ become

$$z - u^2 = 0, \qquad z - h - w^2 = 0, \qquad z(z - h) = 0.$$

Their solution set is the union of the two curves

$$\{(-w^2, 0, 0, w) : w \in \mathbb{R}\} \quad \text{and} \quad \{(u^2, u^2, u, 0) : u \in \mathbb{R}\},$$

where the coordinates are ordered as (h, z, u, w). These curves meet at the origin with distinct tangent directions, so there is no single tangent line to the solution set there. Thus the solution set is not a smooth manifold at the origin. To get around this issue, we construct a nearby regular level set, which, by the regular level set theorem, will allow us to apply Thm. 2.5. Then ϵ is a regular value of F, and the regular level set theorem implies that $F^{-1}(\epsilon)$ is a smooth sub-manifold of $\mathbb{R}^{\kappa+2}$ of co-dimension one, and hence of dimension $\kappa + 1$.

We begin by combining the equations into a single nonnegative exponential–linear polynomial

$$F_0(x, \bar{z}) = \sum_{j=1}^{s} P_j(x, \bar{z})^2.$$

Since every term is nonnegative,

$$F_0(x, \bar{z}) = 0 \quad \Longleftrightarrow \quad P_1(x, \bar{z}) = \cdots = P_s(x, \bar{z}) = 0.$$

Choose any finite number $N \geq 2$ of points $a_1 < \cdots < a_N$ in distinct connected components of $S_{\bar{\theta},v}^{(n)}$. For each $i = 1, \ldots, N-1$, choose $c_i \in (a_i, a_{i+1})$ not in $S_{\bar{\theta},v}^{(n)}$. By Eq. (6), each a_i has a tuple $\bar{z}_i$ satisfying

$$P_1(a_i, \bar{z}_i) = \cdots = P_r(a_i, \bar{z}_i) = 0,$$

whereas no such tuple exists at any c_i.

The next step is to write a modified function F and choose ϵ so that the set $F^{-1}(\epsilon)$ does not contain any point whose x coordinate equals to any of the intermediate values c_i. So we need a strictly positive lower bound for F when $x = c_i$. We therefore modify F_0 as follows: Choose $R > 0$ such that $a_i^2 + \|\bar{z}_i\|^2 < R^2$ for all i. Introduce one more real variable t, and define

$$F(x, \bar{z}, t) = \sum_{i=1}^{s} P_i(x, \bar{z})^2 + \left(x^2 + \|\bar{z}\|^2 + t^2 - R^2\right)^2. \tag{7}$$

The variable t is a slack variable: the equation $x^2 + \|\bar{z}\|^2 + t^2 = R^2$ is solvable in t exactly when $x^2 + \|\bar{z}\|^2 \leq R^2$. Thus, at level zero, the final squared term restricts $(x, \bar{z})$ to the closed ball of radius R about the origin while expressing this restriction using an equation instead of an inequality.

This resulting function has three properties we need. First, for each a_i there are values $\bar{z}_i$ and t_i such that $F(a_i, \bar{z}_i, t_i) = 0$. Second, for each c_i, we have that $\delta_i := \min_{\bar{z},t} F(c_i, \bar{z}, t) > 0$, meaning the function $F(c_i, \bar{z}, t)$ is bounded away from zero (while $F_0(c_i, \bar{z})$ was not, as $\bar{z}$ does not necessarily range over a compact domain). Finally, F is still an exponential–linear polynomial, with the same exponential terms as the P_i and degree $O(1)$.

Now, chose a value $0 < \epsilon < \min_{1 \le i \le N-1} \delta_i$ and inputs $x \in \mathbb{R}$, $\bar{z} \in \mathbb{R}^\kappa$, and $t \in \mathbb{R}$ such that $F(x, \bar{z}, t) = \epsilon$ and $\nabla F(x, \bar{z}, t) \neq \bar{0}$ (in other words, ϵ is a regular value of F)." Such a choice is possible by Sard's theorem. Then $F^{-1}(\epsilon)$ is a smooth manifold. Since $F(a_i, \bar{z}_i, t_i) = 0$ and $F(a_i, \bar{z}_i, t) \to \infty$ as $|t| \to \infty$, continuity shows that $F^{-1}(\epsilon)$ contains a point with $x = a_i$ for every i. On the other hand, by the choice of ϵ, it contains no point with $x = c_i$. Hence these N points lie in distinct connected components of $F^{-1}(\epsilon)$.

The equation $F - \epsilon = 0$ is in $k = \kappa + 2$ variables: the κ variables in $\bar{z}$, together with x and the additional variable t. The equation $F - \epsilon = 0$ has $k = \kappa + 2$ variables and $r = 1$ equation, so

$$\dim F^{-1}(\epsilon) = \kappa + 1 = k - r.$$

We can now apply Thm. 2.5 to the equation $F - \epsilon = 0$. It gives at most

$$2^{O(q^2 + k \log k)} \in 2^{O(n^4)}$$

connected components. The implied constant depends only on the fixed transformer T. Hence,

$$N \in 2^{O(n^4)}$$

and the number of connected components must be at least N. The cases of zero or one component are trivial. □

Proof of Prop. 5.4. Suppose that $T_{\bar{\theta}}$ solves the $b(n)$-th-bit index task. Fix any v. On the intervals

$$\big(j2^{-b(n)}, (j+1)2^{-b(n)}\big), \qquad j = 0, \ldots, 2^{b(n)} - 1,$$

the $2^{-b(n)}$ bit alternates between 0 and 1. Hence $S^{(n)}_{\bar{\theta}, v}$ has at least $2^{b(n)-1}$ connected components. By the preceding lemma,

$$2^{b(n)-1} \in 2^{O_T(n^4)}.$$

Thus $b(n) \in O_T(n^4)$. Thus, the proposition holds. □