NEW LOWER BOUND TECHNIQUES FOR TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing lower bound techniques for transformers still do not allow us to fully understand even the one-layer case. One example of this is such a fundamental and simple task like PARITY: checking if the number of 1s in a binary input sequence is even. While a 2-layer construction for PARITY due to Chiang and Cholak is known, it is open if 1-layer transformers can solve PARITY. Known techniques like communication complexity and Split-VC dimension are not enough to resolve this problem as in the case of PARITY they do not yield non-trivial lower bounds.

In this paper, we develop a new technique that allows us to resolve this problem. We first show for every language, computable by a 1-layer transformer, can be in some sense expressed in the first-order theory of reals with addition, multiplication and order. More precisely, one can write a fixed formula Φ where, for each input length, free variables can be substituted by linear expressions in input bits in a way that the resulting formula computes the restriction of the language to this input length. Using results from Boolean function complexity, we then conclude that every language 1-layer transformers can do has average sensitivity $O(\sqrt{n} \cdot \mathrm{polylog}(n))$, which means that PARITY is not doable by 1-layer transformers.

Moreover, we notice that for 1-layer transformers with a single attention head, the formula Φ does not require multiplication, and this allows us to separate 1-layer transformers with a single head from 1-layer transformers with two heads. More specifically, we show that the problem of checking, whether a sequence of bits is ordered, is not doable by 1-head 1-layer transformers but is doable by a 2-head 1-layer transformer. A similar separation result was previously obtained for the attention-only model, while our results assume the model with the output MLP.

1 INTRODUCTION

Imagine that you are training a deep learning model (for instance, a transformer), and you see that the accuracy oscillates and never reaches a reasonable threshold. Perhaps the model is too big and overfits to the training dataset. Perhaps the model size is alright, but a rugged loss landscape makes it impossible to find the global minimum using gradient descent. But it may also be the case that the model is not expressive enough, and there is simply no assignment of the model weights that would allow it to compute the underlying function. Such a scenario is the point of interest of the *expressivity* of deep learning architectures, in particular, of *transformers* (Strobl et al., 2024).

Showing that some task is not expressible in a given model with a given number of parameters is similar in spirit to computational complexity lower bounds. Curiously, recent studies of the expressive power of transformers are reminiscent of the development of the lower bounds for constant-depth circuits in the eighties. For transformers, this story started with a paper of Hahn (2020) who considered a theoretical model of transformers called *unique hard attention* (UHA). In contrast to the *soft attention*, used in practice, it strictly focuses on a single position where the attention score is maximal. Hahn used a technique of random restrictions to show that UHA-transformers – with $O(1)$ layers and parameters – are not able to compute PARITY, the language of all binary words with an odd number of 1s. This technique resembles the classical random restriction technique (Furst et al., 1984; Hastad, 1986), which one can use to show that PARITY does not belong to the circuit complexity class AC^0 – consisting of languages computable by $O(1)$-depth circuits with AND, OR, and NOT gates.

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.

Similarity between techniques was not a coincidence – as Hao et al. (2022) have shown, AC^0 subsumes languages, expressible by $O(1)$-layer $O(1)$-parameter UHA-transformers. This means that the fact that this kind of transformer cannot express PARITY follows immediately from the circuit lower bound.

At the same time, soft-attention transformers are already able to compute languages outside AC^0. For instance, a simple one-layer soft-attention transformer computes the language MAJORITY (Merrill et al., 2022), consisting of binary words with more 1s than 0s, well-known to be outside AC^0. As for PARITY, a 2-layer soft-attention transformer for this language was given by Chiang & Cholak (2022). Another language outside AC^0, DYCK – the language of balanced words of parentheses – was also shown to have a 2-layer soft-attention transformer, computing it Yao et al. (2021).

This surge in expressivity makes lower bounds for soft-attention transformers significantly more difficult. While UHA-transformers can be paralleled with AC^0-circuits, soft-attention transformers should rather be compared with TC^0-circuits – $O(1)$-depth circuits that, in addition to AND, OR, and NOT gates, also have MAJORITY gates (and are also able to compute PARITY and DYCK). Lower bounds for TC^0 are notoriously hard and out of reach for the current technique, with the frontier being at the class ACC^0, an extension of AC^0 by modular gates instead of MAJORITY gates (Williams, 2014). While soft-attention transformers, under certain assumptions about the precision of computations, are only upper-bounded by TC^0 (Chiang, 2024), lower bounds for them do not seem to be easier than for TC^0.

To move on, a part of the research focused on lower bounds for 1-layer soft-attention transformers. Sanford et al. (2023) and Peng et al. (2024) have developed a communication complexity technique that works against 1-layer soft-attention transformers with $n^{o(1)}$ bits of precision. Using it, they showed 1-layer lower bounds for problems like function composition (given two functions $f, g\colon \{1, 2 \ldots, n\} \to \{1, 2, \ldots, n\}$, compute $f(g(1))$), and the problem of checking whether there are two numbers in an array that sum up to 0. Then, Kozachinskiy et al. (2026) proposed a revision of the communication complexity technique called Split-VC dimension, with which one can show similar lower bounds, but for infinite-precision transformers.

In a recent breakthrough, Chen et al. (2025) obtained the first multi-layer lower bounds for soft-attention transformers (with $n^{o(1)}$ bits of precision). More specifically, using a new technique called autoregressive communication complexity, they have shown that some form of L-iterated function composition problem cannot be computed by a L-layer transformer. A catch though is that this lower bound applies only to transformers with causal masking – those where every position in a transformer can access only previous positions in the attention. This makes the model very sensitive to the order of input bits – something which is not very relevant for circuit models. Exploiting this technicality, Chen et al. (2025) ingeniously circumvent the barrier to prove multi-layer full-attention transformer lower bounds.

Despite this advancement of Chen et al. (2025), even just the 1-layer case for soft-attention transformers remains not fully understood. For instance, PARITY, while it is doable by a 2-layer soft-attention transformer (Chiang & Cholak, 2022), is apparently not known to be doable by a 1-layer one. Moreover, the technique that we currently have is not enough to resolve this question. For instance, the communication complexity technique requires finding a split of input positions between Alice and Bob so that the resulting function has large communication complexity. But for PARITY, the function will have $O(1)$ communication complexity for any splitting (Alice and Bob can exchange 2 bits encoding the parities of their parts). Split-VC works similarly, but instead of communication complexity, one considers a concept class, arising when considering Alice's part as inputs and Bob's part as parameters. For this technique to work, one has to find a split under which the VC dimension of this concept class is large. However, for PARITY, this VC dimension is always just 1.

Our contribution. To circumvent these limitations, in this paper we present a new *logical* lower bound technique for 1-layer transformers. We show that every language computable by a 1-layer soft-attention transformer can be in some sense "embedded" into the first-order theory of reals with addition, multiplication, and order. More precisely, we show (Section 3), for any such language L, that there is a $(\mathbb{R}, +, \cdot, <)$-formula Φ where, for each input length n, free variables can be substituted

by linear expressions in input bits in a way that the resulting formula computes the restriction of L to the input length n.

Then, in Section 4, using results of Kane (2014) about polynomial threshold functions, we derive a corollary that every language computable by a 1-layer transformer has average sensitivity $O(\sqrt{n} \cdot \mathrm{polylog}(n))$ on inputs of length n. This upper bound is essentially tight, as MAJORITY is computable by 1-layer transformers and has average sensitivity $\Omega(\sqrt{n})$ on inputs of length n.

Moreover, as the average sensitivity of PARITY on inputs of length n is n, we conclude that PARITY is not doable by 1-layer transformers. It resolves an open problem by showing that the minimal number of layers needed to compute PARITY is indeed 2.

Next, we notice that for 1-layer transformers with 1 attention head, the formula Φ from our logical lemma does not actually require multiplication. Exploiting this, in Section 5 we give an example of a language, computable by a 1-layer 2-head transformer, but not computable by a 1-layer 1-head transformer. This language is the language ORDERED, consisting of all ordered binary words (where there is no 1 before a 0).

Similar results were previously obtained for 1-layer *attention-only* transformers – transformers without the residual connection and the output MLP. Namely, Tesfaye et al. (2026) first established a separation between 1 and 2 heads in this model, and then Rajaraman et al. (2026) have shown that the k-bit parity provides a separation between k heads and $k-1$ heads, for every $k \geq 2$. In both cases, separating examples are problems that have a *fixed input length*. Hence, they no longer work for the complete model considered in this paper – even with a single head, the output MLP can memorize all the outputs.

In our construction of a 1-layer transformer with 2 heads for ORDERED, one head essentially looks for the position of the last 0, and the second head looks for the position of the first 1. The output MLP checks that they go one after another. In turn, the main technical component of the lower bound is a geometric lemma, lower-bounding the number of halfspaces needed to separate ordered points of a Boolean cube from the non-ordered ones. Similar geometric problems have been previously considered in the area of *relaxation complexity*. However, the difference there is that ordered points of a Boolean cube have to be separated from all the other integer points, not only points of a Boolean cube. Then the problem for ordered points of a Boolean cube is equivalent to the problem for the vertices of a simplex where the exact asymptotic is known Averkov et al. (2026). However, this is not sufficient for our applications – relaxation complexity eliminates all integer points, whereas we separate the ordered binary points from the (much smaller) set of non-ordered binary points

2 BACKGROUND

Basic Notation. For a vector $x \in \mathbb{R}^d$ we write x_i for its i-th coordinate, and $(\mathbb{R}^d)^*$ for the set of all finite sequences of vectors in $\mathbb{R}^d$. We use PARITY to represent the family $\{\mathrm{PARITY}_n\}_{n\geq 1}$,

$$\mathrm{PARITY}_n \colon \{0,1\}. \to \{0,1\}, \qquad \mathrm{PARITY}_n(x_1,\ldots,x_n) = x_1 \oplus \cdots \oplus x_n,$$

where $\oplus$ is the XOR operator, i.e., $a \oplus b = (a+b) \bmod 2$ for $a, b \in \{0,1\}$. We also study a family of functions $\mathrm{ORDERED} = \{\mathrm{ORDERED}_n\}_{n\geq 1}$, where

$$\mathrm{ORDERED}_n : \{0,1\}^n \to \{0,1\}, \qquad \mathrm{ORDERED}_n(x_1,\ldots,x_n) = \begin{cases} 1 & x_1 \leq x_2 \leq \ldots x_n, \\ 0 & \text{otherwise.} \end{cases}$$

We assume the standard soft-attention transformer model, including the residual output MLP network, with the dimension and the parameters independent of the input length, while the input embedding is arbitrary and can be length-dependent. In fact, in our lower bounds, we do not even assume that the input embedding is additive, that is, decomposable into the sum of a token embedding and a positional encoding. At the same time, the input embedding in our 1-layer 2-head construction for ORDERED is additive, with a simple length-independent positional encoding, consisting of the absolute value of the position and of its square.

Attention Layers. A d-dimensional, H-head *attention layer* is a function $L \colon (\mathbb{R}^d)^* \to (\mathbb{R}^d)^*$ parameterised by query, key, and value matrices $Q^{(k)}, K^{(k)}, V^{(k)} \in \mathbb{R}^{d\times d}$, with $k = 1,\ldots,H$, a

heads-mixing matrix $W_O \in \mathbb{R}^{d\times dH}$, and parameters $W_1, W_2 \in \mathbb{R}^{d\times d}$, $b_1, b_2 \in \mathbb{R}^d$ of the output MLP. On input $(\alpha_1, \ldots, \alpha_n) \in (\mathbb{R}^d)^n$, the layer computes for each head $k = 1, \ldots, H$: the *attention logits* of the jth to ith position $L_{ij}^{(k)} \in \mathbb{R}$ and *head values* in jth position $h_j^{(k)} \in \mathbb{R}^d$ (for $i, j = 1, \ldots, n$) as follows:

$$L_{ij}^{(k)} = \frac{\langle K^{(k)}\alpha_i, Q^{(k)}\alpha_j\rangle}{\sqrt{d}}, \quad (1) \qquad h_j^{(k)} = \frac{\sum_{i=1}^n \exp\{L_{ij}^{(k)}\}\, V^{(k)}\alpha_i}{\sum_{i=1}^m \exp\{L_{ij}^{(k)}\}}. \quad (2)$$

The head values are combined and passed through a position-wise feed-forward network:

$$h_j = W_O \begin{pmatrix} h_j^{(1)} \\ \vdots \\ h_j^{(H)} \end{pmatrix}, \quad (3) \qquad \beta_j = W_2 \cdot \mathrm{ReLU}\big(W_1(h_j + \alpha_j) + b_1\big) + b_2, \quad (4)$$

where $\mathrm{ReLU}(x_1, \ldots, x_d) = (\max\{0, x_1\}, \ldots, \max\{0, x_d\})$. The output sequence is $(\beta_1, \ldots, \beta_n) = L(\alpha_1, \ldots, \alpha_n)$.

Transformers. We consider transformers that map sequences of tokens into the next, most probable token. We restrict our attention to 1-layer transformers. A 1-layer, H-head, d-dimensional *transformer* over a finite vocabulary $\mathcal{V}$ (a set of *tokens* which includes symbol $\perp$) is a function $T\colon \mathcal{V}^* \to \mathcal{V}$ specified by an H-head d-dimensional attention layer L, an input embedding $\mathrm{E}\colon \mathcal{V} \times \mathbb{N}^2 \to \mathbb{R}^d$, and an output distribution matrix $W \in \mathbb{R}^{|\mathcal{V}|\times d}$. On input $x_1 \ldots x_n \in \mathcal{V}^n$, it applies position-wise the input embedding $\alpha_i = \mathrm{E}(x_i, i, n)$, then applies the attention layer computing $(\beta_1, \ldots, \beta_n) = L(\alpha_1, \ldots, \alpha_n)$, and outputs

$$T(x_1, \ldots, x_n) = \arg\max_{x\in\mathcal{V}} \big(\mathrm{softmax}(W\beta_n)\big)_x,$$

with the convention that $T(x_1, \ldots, x_n) = \perp$ if the argmax is not unique. We say that a transformer T *computes* a sequence $\{f_n\}_{n\in\mathbb{N}}$ of Boolean functions $f_n\colon \{0,1\}^n \to \{0,1\}$ if $\{0,1\} \subseteq \mathcal{V}$ and $T(x) = f_n(x)$, for every $n \in \mathbb{N}$ and every $x \in \{0,1\}^n$.

Sensitivity and Average Sensitivity Sensitivity is defined for functions $f\colon \{0,1\}^n \to \{0,1\}$ mapping binary strings of length n to 0 or 1. *Sensitivity* of such a function f at input $x \in \{0,1\}^n$, denoted $s_x(f)$, is the number of input positions $i \in \{1, \ldots, n\}$ such that flipping the ith bit of x changes the value of $f(x)$. The *average sensitivity* of f is defined as $as(f) = \sum_{x\in\{0,1\}^n} \frac{s_x(f)}{2^n}$, that is, the expected number of bit positions at which f is sensitive, where the expectation is taken uniformly over all inputs $x \in \{0,1\}^n$. For example, PARITY on inputs of length n has the maximum possible average sensitivity: since flipping any single bit always changes the parity, we have $s_x(\text{PARITY}) = n$ for every $x \in \{0,1\}^n$, and so, $as(\text{PARITY}) = n$.

In the paper we will also use a technical result establishing a bound on the average sensitivity of polynomial threshold functions. A function $f\colon \{0,1\}^n \to \{0,1\}$ is called a degree-d *polynomial threshold function* (PTF) if there exists a degree-d polynomial $P \in \mathbb{R}[x_1, \ldots, x_n]$ such that $P(x) > 0$ for all $x \in \{0,1\}^n$ with $f(x) = 1$, and $P(x) < 0$ for all $x \in \{0,1\}^n$ with $f(x) = 0$. For example, the majority function $\mathrm{maj}_n(x_1, \ldots, x_n) = 1$ iff $x_1 + \cdots + x_n > \frac{n}{2}$ is a degree-1 PTF, witnessed by the polynomial $P(x_1, \ldots, x_n) = x_1 + \cdots + x_n - \frac{n}{2}$. A result of Kane, which will be important for our paper, establishes an upper bound on the average sensitivity of $O(1)$-degree PTFs

Theorem 1 ((Kane, 2014)). *Let f be a degree-d PTF over $n > 1$ variables. Then*

$$as(f) \le \sqrt{n}(\log n)^{O(d\log d)} 2^{O(d^2 \log d)}.$$

Logic over Reals and Quantifier Elimination A first-order logic over the reals, $(\mathbb{R}, <, +, \cdot)$, is a formalisms allowing to write formulae that are built from real-valued variables and constants, using addition ($+$), multiplication ($\cdot$), and comparison ($<$), as well as standard logical connectives ($\vee$, $\wedge$,

$\neg$, $\rightarrow$) and quantifiers ($\forall$, $\exists$). For example, we can write the formula $\exists x\ (a \cdot x \cdot x + b \cdot x + c < 0)$, stating that the quadratic polynomial $a \cdot x^2 + b \cdot x + c$, with coefficients a, b, c, has a negative value for some $x \in \mathbb{R}$. The seminal Tarski-Seidenberg theorem (Tarski, 1952; Seidenberg, 1954) implies that $(\mathbb{R}, <, +, \cdot)$ admits quantifier elimination, that is, every formula can be rewritten into an equivalent formula that has no quantifiers. For example, the formula above is equivalent to the quantifier-free formula $b \cdot b - 4 \cdot a \cdot c > 0$. In this paper, we will use quantifier elimination in $(\mathbb{R}, <, +, \cdot)$ to show that certain properties of transformer computations can be expressed as polynomial conditions.

We will also use the fact that if we remove the multiplication, the resulting interpretation $(\mathbb{R}, <, +)$ admits quantifier elimination as well (Ferrante & Rackoff, 1975) – for a formula with quantifiers but without multiplication, there is an equivalent quantifier-free and multiplication-free formula.

3 Logical Lemma

In this section, we establish the following structural result about Boolean functions computable by 1-layer transformers.

Lemma 2. *Assume there is a 1-layer transformer T, computing a sequence of Boolean functions $\{f_n\}_{n=1}^{\infty}$. Then there exists a first-order formula $\Phi(z_1, \ldots, z_r)$ in the interpretation $(\mathbb{R}, <, +, \cdot)$ such that for all n there exist r polynomials $l_1, \ldots, l_r \in \mathbb{R}[x_1, \ldots, x_n]$ of degree at most 1 such that for any $x \in \{0,1\}^n$, we have $f_n(x) = 1$ if and only if $\Phi(l_1(x), \ldots, l_r(x)) = 1$.*

Moreover, if T is 1-head, then such a formula Φ can be constructed without multiplications.

Proof of Lemma 2. Due to space constraints, the proof of the second part – the 1-head case – is deferred to Appendix A. We establish the result under the assumption that the last bit of all functions f_n, that is, x_n, is always fixed to 0. This way we get a formula Φ_0, working under this assumption, and also, by the same argument, one can get a formula Φ_1, working under the assumption that $x_n = 1$. The final formula Φ is obtained by

$$\Phi = ((z_0 = 0) \rightarrow \Phi_0) \wedge ((z_0 = 1) \rightarrow \Phi_1)$$

where z_0 is a fresh free variable, which for inputs of length n gets substituted by the last input bit.

Once again, we assume that x_n, the last input bit when the input has length n, is fixed to 0. Thus, the value of f_n becomes the function of the first $n-1$ input bits, $x \in \{0,1\}^{n-1}$.

Assume that the dimension of T is d and the number of attention heads of T is H. Our transformer T is 1-layer. Hence, only the attention in the n-th position is relevant for computing the output. We first observe that for every $k = 1, \ldots, H$, the vector $h_n^{(k)} \in \mathbb{R}^d$ – the value of the k-th attention head in the n-th position – as a function of $x \in \{0,1\}^{n-1}$, can be represented as

$$h_n^{(k)} = \begin{pmatrix} \ell_1^{(k)}/\ell_0^{(k)} \\ \vdots \\ \ell_d^{(k)}/\ell_0^{(k)} \end{pmatrix}, \tag{5}$$

where $\ell_0^{(k)}, \ldots, \ell_d^{(k)} \in \mathbb{R}[x_1, \ldots, x_{n-1}]$ are of degree 1. Indeed, by definition:

$$h_n^{(k)} = \frac{\sum_{i=1}^{n} \exp\{L_{in}^{(k)}\} V^{(k)} \alpha_i}{\sum_{i=1}^{n} \exp\{L_{in}^{(k)}\}} \in \mathbb{R}^d, \tag{6}$$

where $V^{(k)} \in \mathbb{R}^{d \times d}$ is a fixed matrix, $\alpha_1 = E(x_1, 1, n) \in \mathbb{R}^d, \ldots, \alpha_{n-1} = E(x_{n-1}, n-1, n) \in \mathbb{R}^d, n-1, n), \alpha_n = E(0, n, n) \in \mathbb{R}^d$ are embeddings of input bits (recall again that x_n is fixed to 0) and

$$L_{in}^{(k)} = \langle K^{(k)} \alpha_i, Q^{(k)} \alpha_n \rangle / \sqrt{d}$$

for some fixed matrices $K^{(k)}, Q^{(k)} \in \mathbb{R}^{d \times d}$.

The term in the numerator of equation 15, corresponding to $i = n$, is a vector $\gamma_{nk} = \exp\{L^{(k)}_{nn}\}V^{(k)}\alpha_n \in \mathbb{R}^d$, not depending on $x \in \{0,1\}^{n-1}$. Likewise, the term in the denominator of equation 15 for $i = n$ is a number $\rho_{nk} \in \mathbb{R}$, not depending on $x \in \{0,1\}^{n-1}$. In turn, for every $i = 1, \ldots, n-1$, the i-th term in the numerator of equation 15 is a vector:

$$\exp\{L^{(k)}_{in}\}V^{(k)}\alpha_i = \exp\left\{\langle K^{(k)}\alpha_i, Q^{(k)}\alpha_n\rangle/\sqrt{d}\right\}V^{(k)}\alpha_i$$
$$= \exp\left\{\langle K^{(k)} \cdot \mathrm{E}(x_i, i, n), Q^{(k)} \cdot \mathrm{E}(0, n, n)\rangle/\sqrt{d}\right\}V^{(k)}\mathrm{E}(x_i, i, n) \in \mathbb{R}^d,$$

whose value is determined by i, n, k, and x_i. Hence, it can be written as $(1 - x_i)\gamma^0_{ink} + x_i\gamma^1_{ink}$ for some $\gamma^0_{ink}, \gamma^1_{ink} \in \mathbb{R}^d$. Similarly, the i-th term in the denominator of equation 15, for $i = 1, \ldots, n-1$, can be written as $(1 - x_i)\rho^0_{ink} + x_i\rho^1_{ink}$ for some $\rho^0_{ink}, \rho^1_{ink} \in \mathbb{R}$. Overall, we obtain:

$$h^{(k)}_n = \frac{\gamma_{nk} + \sum_{i=1}^{n-1}((1 - x_i)\gamma^0_{ink} + x_i\gamma^1_{ink})}{\rho_{nk} + \sum_{i=1}^{n-1}((1 - x_i)\rho^0_{ink} + x_i\rho^1_{ink})} = \begin{pmatrix} \ell^{(k)}_1/\ell^{(k)}_0 \\ \vdots \\ \ell^{(k)}_d/\ell^{(k)}_0 \end{pmatrix},$$

where $\ell^{(k)}_0, \ldots, \ell^{(k)}_d \in \mathbb{R}[x_1, \ldots, x_{n-1}]$ are of degree 1. With this, equation 5 is established.

The formula Φ_0 will have $r = (d + 1)H + d$ free variables denoted by $z^{(k)}_i$, $i = 0, \ldots, d, k = 1, \ldots, H$ and $a_1, \ldots, a_d$. We want the formula Φ_0 to decide, whether the output of the transformer T is equal to the token 1 on $x \in \{0,1\}^{n-1}$ when one substitutes

$$z^{(k)}_i \leftarrow \ell^{(k)}_i, \qquad i = 0, \ldots, d, k = 1, \ldots, H,$$
$$a_i \leftarrow (\alpha_n)_i, \qquad i = 1, \ldots, d.$$

where $\ell^{(k)}_i \in \mathbb{R}[x_1, \ldots, x_{n-1}]$ are of degree 1 and satisfy equation 5, and $(\alpha_n)_1, \ldots, (\alpha_n)_d$ are coordinates of the vector $\alpha_n = E(0, n, n)$ (in place of variables $a_1, \ldots, a_d$ we thus substitute just constants, that is, 0-degree polynomials).

Let $\beta_n \in \mathbb{R}^d$ be the output of the attention layer in the n-th position. The transformer, to produce its output token, multiplies β_n by the output-distribution matrix $W \in \mathbb{R}^{\mathcal{V}\times d}$ and applies the softmax operation to the resulting vector, obtaining a distribution over the vocabulary $\mathcal{V}$:

$$\mu = \mathrm{softmax}(W\beta_n).$$

The token with the maximal probability in this distribution is set as the output one. According to our conventions, the maximum has to be unique, and if it is not, $\perp$ is set as the output token. Hence, for 1 to be the output token, we must have:

$$(W\beta_n)_1 > (W\beta_n)_\sigma, \qquad \sigma \in \mathcal{V} \setminus \{1\}.$$

The latter, when expanded, turns into

$$W^1_1\beta^1_n + \ldots + W^1_d\beta^d_n > W^\sigma_1\beta^1_n + \ldots + W^\sigma_d\beta^d_n, \qquad \sigma \in \mathcal{V} \setminus \{1\}.$$

Correspondingly, we introduce into our formula Φ variables $B^1, \ldots, B^d$ that are supposed to correspond to d coordinates of β_n. We start by writing:

$$\exists B^1 \ldots \exists B^d \bigwedge_{\sigma\in\mathcal{V}\setminus\{1\}} \left(W^1_1 B^1 + \ldots + W^1_d B^d > W^\sigma_1 B^1 + \ldots + W^\sigma_d B^d\right). \tag{7}$$

This formula is just a conjunction of linear inequalities with some fixed coefficients. These are expressible in $(\mathbb{R}, <, +, \cdot)$ as we are assuming that we can use all real constants in the formulas.

The vector β_n, according to equation 4, is computed by $\beta_n = W_2 \cdot \mathrm{ReLU}\,(W_1(h_n + \alpha_n) + b_1) + b_2 \in \mathbb{R}^d$, where W_1, W_2, b_1, b_2 consist of fixed constants. We already have variables $a_1, \ldots, a_d$,

corresponding to coordinates of α_n. We introduce variables $\chi_1, \ldots, \chi_d$, corresponding to the coordinates of h_n. Our goal is to make sure that the following condition:

$$\exists \chi_1 \ldots \exists \chi_d \begin{pmatrix} B^1 \\ \vdots \\ B^d \end{pmatrix} = W_2 \cdot \mathrm{ReLU}\left(W_1 \cdot \begin{pmatrix} \chi_1 + a_1 \\ \vdots \\ \chi_d + a_d \end{pmatrix} + b_1 \right) + b_2 \tag{8}$$

is expressible in $(\mathbb{R}, <, +, \cdot)$. Indeed, we can introduce a variable for each of the results of intermediate computations in equation 8 (with an existential quantifier). Each of these variables is computed by some linear combinations of other variables, or by ReLU, applied to some other variable. Both are expressible in $(\mathbb{R}, <, +, \cdot)$; the equality $y = \mathrm{ReLU}(x)$ is expressible, for example, via: $(x < 0 \to y = 0) \wedge (x \geq 0 \to y = x)$. Finally, the vector h_n is computed via $h_n = W_O \begin{pmatrix} h_n^{(1)} \\ h_n^{(2)} \\ \vdots \\ h_n^{(H)} \end{pmatrix}$, and the coordinates of vectors $h_n^{(k)}$ are equal to fractions of degree-1 polynomials $\ell_i^{(k)}$ as in equation 5 (and these polynomials are being substituted in place of variables $z_i^{(k)}$). To finish the construction of Φ, for $i = 1, \ldots, d$, $k = 1, \ldots, H$, we introduce a variable $\chi_i^{(k)}$, corresponding to the i-th coordinate of the vector $h_n^{(k)}$. By equation 5, we have $(h_n)_i^{(k)} = \ell_i^{(k)} / \ell_0^{(k)}$. Correspondingly, our formula must include conditions of the form $\chi_i^{(k)} = z_i^{(k)} / z_0^{(k)}$. Writing these conditions using multiplication, we get:

$$\exists \chi_1^{(1)} \ldots \exists \chi_d^{(H)} \begin{pmatrix} \chi_1 \\ \vdots \\ \chi_d \end{pmatrix} = W_O \begin{pmatrix} \chi_1^{(1)} \\ \vdots \\ \chi_d^{(1)} \\ \vdots \\ \chi_1^{(H)} \\ \vdots \\ \chi_d^{(H)} \end{pmatrix} \wedge \left(\bigwedge_{i=1}^{d} \bigwedge_{k=1}^{H} z_i^{(k)} = \chi_i^{(k)} \cdot z_0^{(k)} \right). \tag{9}$$

The formula Φ_0 is obtained as the conjunction of (7–9). □

4 Average Sensitivity Corollary and PARITY

We derive the following upper bound on the average sensitivity of functions computable by 1-layer transformers.

Theorem 3. *Assume that a sequence of Boolean functions $\{f_n\}_{n=1}^{\infty}$ is computable by a 1-layer transformer. Then $as(f_n) = O(\sqrt{n} \cdot (\log n)^{O(1)})$ as $n \to \infty$.*

Proof. Take the formula $\Phi(z_1, \ldots, z_r)$, satisfying the conditions of Lemma 2. Due to the Tarski-Seidenberg theorem, one can eliminate quantifiers in it. Thus, without loss of generality, one can assume that $\Phi(z_1, \ldots, z_t)$ is quantifier-free. In other words, this is just a Boolean combination of a fixed number of fixed polynomial equalities and inequalities in $z_1, \ldots, z_r$. They turn into a fixed number of $O(1)$-degree polynomial equalities and inequalities in $x_1, \ldots, x_n$ when we substitute $\ell_1, \ldots, \ell_r$ in place of $z_1, \ldots, z_r$. After this substitution, the value of the formula coincides with $f_n(x)$.

In other words, there are $k = O(1)$ polynomials $P_1, \ldots, P_k$ of degree $O(1)$ such that the value of $f_n(x)$ for $x \in \{0, 1\}^n$ can be computed by asking the "extended signs" of $P_1, \ldots, P_k$ on x (where by the extended sign of a number we mean 0 if the number is 0, and ± 1 depending if the number is positive/negative, respectively. We require being able to say whether a number is exactly 0 to handle equalities in the formula Φ).

We now argue that f_n must have low average sensitivity, using Theorem 1.

Average sensitivity is equal to the expected number of positions in a random input such that the value of the function changes after flipping the bit in this position. The value of f_n may only change if the extended sign of one of the polynomials $P_1, \ldots, P_k$ changes. It remains to show that a single $O(1)$-degree polynomial P changes its extended sign for at most $\sqrt{n} \cdot (\log n)^{O(1)}$ positions on average for a random input. This follows from Theorem 1 and the following observation. Take $\varepsilon > 0$ which is smaller in absolute value than any non-zero value of P on an input from the Boolean cube. Note that $P + \varepsilon, P - \varepsilon$ never take the value 0 on the Boolean cube and thus compute $O(1)$-degree PTFs. By Theorem 1, each of $P + \varepsilon, P - \varepsilon$ changes its value from positive to negative or vice versa for at most $\sqrt{n} \cdot (\log n)^{O(1)}$ positions on average. It remains to notice that signs of $P(x) + \varepsilon, P(x) - \varepsilon$ determine the extended sign of $P(x)$. Namely, $P(x) > 0$ only if $P(x) + \varepsilon, P(x) - \varepsilon$ are both positive, $P(x) = 0$ if one of these numbers is positive and the other is negative, and $P(x) < 0$ if both numbers are negative. □

Corollary 4. *No 1-layer transformer computes PARITY.*

Proof. $as(\mathrm{PARITY}_n) = n$. □

5 Separation Between 1-Head and 2-Head 1-Layer Transformers

Theorem 5. ORDERED *can be computed by a 2-head 1-layer transformer, but it cannot be computed by a 1-head 1-layer transformer.*

Proof. A construction of a 2-head 1-layer transformer for ORDERED is deferred to Subsection 5.1. For the lower bound, we require the following geometric lemma. We say that an n-dimensional vector $(x_1, \ldots, x_n) \in \mathbb{R}^n$ is binary if $x_i \in \{0, 1\}$ for all $i = 1, \ldots, n$, and is ordered if $x_1 \leq x_2 \leq \ldots \leq x_n$.

Lemma 6. *For $n \in \mathbb{N}$, let θ_n denote the minimal $k \in \mathbb{N}$ such that there exist k halfspaces $H_1, \ldots, H_k \in \mathbb{R}^n$, satisfying the following properties:*

- *all ordered n-dimensional binary vectors belong to $H_1 \cap H_2 \cap \ldots \cap H_k$;*
- *none of the non-ordered n-dimensional binary vectors belongs to $H_1 \cap H_2 \ldots \cap H_k$;*

(each of the halfspaces can be with or without the boundary). It holds that: $\lim_{n \to \infty} \theta_n = +\infty$.

The proof of Appendix 6 is deferred to Section B. Now, assume for contradiction that ORDERED is computable by a 1-head 1-layer transformer T. Then a formula $\Phi(z_1, \ldots, z_r)$ as in Lemma 2 exists for ORDERED. Since T has just 1 head, we may assume that Φ is a $(\mathbb{R}, <, +)$-formula. Further, by the result of Ferrante & Rackoff (1975), we may assume that $\Phi(z_1, \ldots, z_r)$ is quantifier-free. Let C be the number of its atomic subformulas. Without loss of generality, we may assume that all of them are non-strict inequalities. We will show that $\theta_n \leq C$ for infinitely many n, contradicting Lemma 6.

Take any n. There are $\ell_1, \ldots, \ell_r \in \mathbb{R}[x_1, \ldots, x_n]$ of degree at most 1 such that

$$\mathrm{ORDERED}_n(x) = \Phi(\ell_1(x), \ldots, \ell_r(x)) \tag{10}$$

for all $x \in \{0, 1\}^n$. Each atomic subformula in equation 10 is a non-strict linear inequality in $x_1, \ldots, x_n$. Each binary vector $x \in \{0, 1\}^n$ gives rise to a sequence of C bits which we will call the Φ-pattern of x, encoding which atomic subformulas in equation 10 are true on x. The Φ-pattern of a binary vector x determines the value of the formula Φ on it, and hence an ordered binary vector and a non-ordered binary vector cannot have the same Φ-pattern. Out of $n + 1$ ordered binary vectors, there exist at least $m \geq (n + 1)/2^C$ with the same Φ-pattern. Let these m vectors be $x^1, \ldots, x^m$. Their Φ-pattern defines an intersection of C halfspaces $H_1, \ldots, H_C$ (some might be with boundary and the other without) that contains $x^1, \ldots, x^m$ but does not contain any non-ordered binary vector.

Vectors $x^1, \ldots, x^m$ are all equal to 0 in some initial positions and to 1 in some final positions. The rest of positions can be cut into $m - 1$ blocks $B_1, B_2, \ldots, B_{m-1}$ so that:

	B_1	B_2	$\dots$	B_{m-1}
$x^1 =$	$00\dots0$	$00\dots0$	$\dots$	$00\dots0$
$x^2 =$	$00\dots0$	$00\dots0$	$\dots$	$11\dots1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$x^{m-1} =$	$00\dots0$	$11\dots1$	$\dots$	$11\dots1$
$x^m =$	$11\dots1$	$11\dots1$	$\dots$	$11\dots1$

Consider an affine mapping from $\mathbb{R}^{m-1}$ to $\mathbb{R}^n$, given my:

$$(y_1, \dots, y_{m-1}) \mapsto (\underbrace{y_1, y_1, \dots, y_1}_{B_1}, \underbrace{y_2, y_2, \dots, y_2}_{B_2}, \dots, \underbrace{y_{m-1}, \dots, y_{m-1}}_{B_{m-1}})$$

(positions before B_1 are set to 0 and after B_{m-1} to 1). Under this mapping, ordered sequences in $\{0,1\}^{m-1}$ go into $x^1, \dots, x^m$ and thus inside $H_1 \cap \dots \cap H_C$, and non-ordered sequence in $\{0,1\}^{m-1}$ go into non-ordered sequences in $\{0,1\}^n$ – outside $H_1 \cap \dots \cap H_C$. Rewriting inequalities, defining $H_1, \dots, H_C$, in coordinates $y_1, \dots, y_{m-1}$, we obtain that $\theta_{m-1} \le C$. Since n can be chosen arbitrarily and $m \ge n/2^C$, we obtain the inequality $\theta_{m-1} \le C$ for infinitely many m.

□

5.1 Construction with 2 Heads

We use the following input embedding: $E(x_i, i, n) = \begin{pmatrix} i \\ i^2 \\ x_i \end{pmatrix}, \qquad i = 1, \dots, n.$ The two heads compute the following attention logits: $L^{(1)}_{in} = in - x_i n^2, \qquad L^{(2)}_{in} = x_i n^2 - in.$

Let i_0 be the position of the last 0 and i_1 be the position of the first 1. It is easy to see that the attention logits of the first and second head are maximized at positions i_0 and i_1, respectively, with a margin at least n:

$$L^{(1)}_{i_0 n} \ge L^{(1)}_{in} + n \qquad i \ne i_0, \tag{11}$$

$$L^{(2)}_{i_1 n} \ge L^{(2)}_{in} + n \qquad i \ne i_1. \tag{12}$$

The first and the second head compute, respectively,

$$i^*_0 = \frac{\sum_{i=1}^n \exp\{L^{(1)}_{in}\} i}{\sum_{i=1}^n \exp\{L^{(1)}_{in}\}}, \qquad i^*_1 = \frac{\sum_{i=1}^n \exp\{L^{(2)}_{in}\} i}{\sum_{i=1}^n \exp\{L^{(2)}_{in}\}}.$$

Due to (11–12), one can show that $|i_0 - i^*_0| < 0.1$, $|i_1 - i^*_1| < 0.1$. In fact, one can upper bound these differences by $poly(n)e^{-n}$ but the 0.1-upper bound is sufficient. The sequence is ordered if and only if $i_0 + 1 = i_1$. The output MLP checks if $|i^*_1 - 1 - i^*_0| < 0.2$.

One has to separately consider cases when the input has only 0s or only 1s, that is, when either i_0 or i_1 is not defined. It is easy to see that when $x_1 = \dots = x_n$, the attention logits of the first head are maximized at $i = n$, and of the second head at $i = 1$, both with a margin of at least n. That is, i^*_0 will be 0.1-close to n and i^*_1 will be 0.1-close to 1. To distinguish it from the case when the sequence is of the form $1 * \dots * 0$, our attention heads additionally compute:

$$x^*_n = \frac{\sum_{i=1}^n \exp\{L^{(1)}_{in}\} x_i}{\sum_{i=1}^n \exp\{L^{(1)}_{in}\}}, \qquad x^*_1 = \frac{\sum_{i=1}^n \exp\{L^{(2)}_{in}\} x_i}{\sum_{i=1}^n \exp\{L^{(2)}_{in}\}}.$$

When $L^{(1)}_{in}$ is maximized at $i = n$ and $L^{(2)}_{in}$ is maximized at $i = 1$, the values of x^*_1, x^*_n will be 0.1-close to x_1, x_n respectively. Hence, the output MLP can take into account the case $x_1 = \dots = x_n$ as follows. If $|i^*_0 - 1| + |i^*_1 - n| < 0.2$, it compares x^*_1, x^*_n. If they are very close, it outputs that the sequence is ordered; if not, that they are not. Otherwise, if $|i^*_0 - 1| + |i^*_1 - n| \ge 0.2$, it outputs that the sequences are ordered if and only if $|i^*_1 - 1 - i^*_0| < 0.2$.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

We used an LLM in the final step of the proof of Lemma 6, to show the lower bound on the chromatic number of the graph T_n. The rest of the proofs, as well as all the write-up of the paper, were done without AI tools.

ETHICS STATEMENT

(This section is **recommended** and does not count toward the page limit.)

If authors feel that their paper submission raises questions regarding the Code of Ethics, they are encouraged to include a paragraph of Ethics Statement (at the end of the main text before references) to address potential concerns where appropriate. Topics include, but are not limited to, studies that involve human subjects, practices to data set releases, potentially harmful insights, methodologies and applications, potential conflicts of interest and sponsorship, discrimination/bias/fairness concerns, privacy and security issues, legal compliance, and research integrity issues (e.g., IRB, documentation, research ethics). This statement should not be more than 1 page.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

It is important that the work published in ICLR is reproducible. Authors are strongly encouraged to include a paragraph-long Reproducibility Statement at the end of the main text (before references) to discuss the efforts that have been made to ensure reproducibility. This paragraph should not itself describe details needed for reproducing the results, but rather reference the parts of the main paper, appendix, and supplemental materials that will help with reproducibility. For example, for novel models or algorithms, a link to an anonymous downloadable source code can be submitted as supplementary materials; for theoretical results, clear explanations of any assumptions and a complete proof of the claims can be included in the appendix; for any datasets used in the experiments, a complete description of the data processing steps can be provided in the supplementary materials. Each of the above are examples of things that can be referenced in the reproducibility statement.

AUTHOR CONTRIBUTIONS

If you'd like to, you may include a section for author contributions as is done in many journals. This is optional and at the discretion of the authors.

ACKNOWLEDGMENTS

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

REFERENCES

Gennadiy Averkov, Simon Keil, and Stefan Weltge. The relaxation complexity of the standard simplex is logarithmic. *arXiv preprint arXiv:2606.11852*, 2026.

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

David Chiang. Transformers in uniform tc^0. *arXiv preprint arXiv:2409.13629*, 2024.

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 (Volume 1: Long Papers)*, pp. 7654–7664, 2022.

Paul Erdos and Richard Rado. Combinatorial theorems on classifications of subsets of a given set. *Proceedings of the London mathematical Society*, 3(1):417–439, 1952.

Jeanne Ferrante and Charles Rackoff. A decision procedure for the first order theory of real addition with order. *SIAM Journal on Computing*, 4(1):69–76, 1975.

Merrick Furst, James B Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical systems theory*, 17(1):13–27, 1984.

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

Yiding Hao, Dana Angluin, and Robert Frank. Formal language recognition by hard attention transformers: Perspectives from circuit complexity. *Transactions of the Association for Computational Linguistics*, 10:800–810, 2022.

John Hastad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pp. 6–20, 1986.

Daniel M. Kane. The correct exponent for the gotsman-linial conjecture. *Comput. Complex.*, 23(2): 151–175, 2014. doi: 10.1007/S00037-014-0086-Z. URL https://doi.org/10.1007/s00037-014-0086-z.

Alexander Kozachinskiy, Felipe Urrutia, Hector Orellana, Tomasz Steifer, Germán Pizarro, Matías Fuentes, Francisco Meza Vásquez, Cristian Buc Calderon, and Cristóbal Rojas. Strassen attention, split vc dimension and compositionality in transformers. *Advances in Neural Information Processing Systems*, 38:9665–9696, 2026.

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.

Binghui Peng, Srini Narayanan, and Christos Papadimitriou. On limitations of the transformer architecture. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=KidynPuLNW.

Rajmohan Rajaraman, Ravi Sundaram, and Amanuel Tesfaye. The head complexity of boolean functions in single-layer attention. *arXiv preprint arXiv:2609.04046*, 2026.

Clayton Sanford, Daniel J Hsu, and Matus Telgarsky. Representational strengths and limitations of transformers. *Advances in neural information processing systems*, 36:36677–36707, 2023.

Abraham Seidenberg. A new decision method for elementary algebra. *Annals of Mathematics*, 60 (2):365–374, 1954.

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.

Alfred Tarski. A decision method for elementary algebra and geometry. *Journal of Symbolic Logic*, 17(3), 1952.

Amanuel Tesfaye, Zeno Kujawa, Rajmohan Rajaraman, and Ravi Sundaram. Two (narrow) heads are better than (an arbitrarily wide) one. In *International Conference on Learning Representations*, volume 2026, pp. 20112–20128, 2026.

Ryan Williams. Nonuniform acc circuit lower bounds. *Journal of the ACM (JACM)*, 61(1):1–32, 2014.

Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. Self-attention networks can process bounded hierarchical languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3770–3785, 2021.

A Proof of Lemma 2 for 1 Head

We construct two formulas Φ_0, Φ_1 that work for inputs where the last bit is fixed to 0 and 1, respectively. The final formula Φ can be obtained by introducing a fresh variable $\mathbf{z}$ which will be substituted by x_n, and writing

$$\Phi := ((\mathbf{z} = 0) \to \Phi_0) \wedge ((\mathbf{z} = 1) \to \Phi_1).$$

It remains now to construct Φ_0 (the formula Φ_1 can be constructed similarly). The output of the transformer is determined by the maximal coordinate of the vector $W\beta_n$, where W is the output-distribution matrix, and β_n is the output vector of the layer in the n-th position. More specifically, for the output of the transformer to be 1, the value of the 1-coordinate of $W\beta_n$ has to be strictly larger than the corresponding coordinate for all the other tokens of the vocabulary:

$$(W\beta_n)_1 > (W\beta_n)_\sigma, \qquad \sigma \in \mathcal{V} \setminus \{1\}. \tag{13}$$

In turn, β_n is computed as in equation 2 for $j = n$:

$$\beta_n = W_2 \cdot \mathrm{ReLU}\big(W_1(h_n + \alpha_n) + b_1\big) + b_2. \tag{14}$$

Firstly, we introduce d variables $\mathbf{u}_1, \ldots, \mathbf{u}_d$, corresponding to coordinates of the vector $h_n + \alpha_n$. We write a $(\mathbb{R}, <, +)$-formula $\Psi(\mathbf{u}_1, \ldots, \mathbf{u}_d)$, expressing a fact that the vector β_n, computed as in equation 14 with

$$\begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u_2} \\ \vdots \\ \mathbf{u_d} \end{pmatrix}$$

in place of $h_n + \alpha_n$, satisfies inequalities equation 13. This is doable because all the vectors in question have fixed dimension, the matrices and vectors W, W_1, W_2, b_1, b_2 consist of finitely many fixed real numbers, and the ReLU operation is expressible in $(\mathbb{R}, <, +)$:

$$y = \mathrm{ReLU}(x) \iff (x \geq 0 \to y = x) \wedge (x < 0 \implies y = 0).$$

By a result of Ferrante & Rackoff (1975), the interpretation $(\mathbb{R}, <, +)$ admits quantifier elimination, and thus the formula Ψ can be assumed to be quantifier-free. For any input length n, for any transformer input $x \in \{0,1\}^n$, if we compute the values of the coordinates of $h_n + \alpha_n$ on this input and put these values into the formula Ψ, it will output the value $f_n(x)$.

To finish the argument, it is enough to show that for any input length n, for inputs with the last bit x_n fixed to 0, there exist $d+1$ polynomials $\ell_0, \ell_1, \ldots, \ell_d \in \mathbb{R}[x_1, \ldots, x_n]$ of degree at most 1 such that:

$$h_n + \alpha_n = \begin{pmatrix} \ell_1(x)/\ell_0(x) \\ \ell_2(x)/\ell_0(x) \\ \vdots \\ \ell_d(x)/\ell_0(x) \end{pmatrix}, \tag{15}$$

where $\ell_0(x) > 0$ for all $x \in \{0,1\}^n$. Indeed, then we can introduce $d+1$ variables $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_d$ and consider a formula:

$$\Phi_0(\mathbf{v}_0, \ldots, \mathbf{v}_d) = \Psi(\frac{\mathbf{v}_1}{\mathbf{v}_0}, \ldots, \frac{\mathbf{v}_d}{\mathbf{v}_0}).$$

As written, the right-hand side formula will be a Boolean combination of linear inequalities with fractions $\frac{\mathbf{v}_1}{\mathbf{v}_0}, \ldots, \frac{\mathbf{v}_d}{\mathbf{v}_0}$. We can simply multiply them by their common[1] denominator $\mathbf{v}_0$ (taking into account that the function $\ell_0(x)$ that will be substituted in place of $\mathbf{v}_0$ takes only positive values on the transformer inputs) to obtain equivalent linear inequalities in $\mathbf{v}_0, \ldots, \mathbf{v}_d$. It remains to substitute $\ell_0, \ell_1, \ldots, \ell_d$ in place of $\mathbf{v}_0, \mathbf{v}_1 \ldots, \mathbf{v}_d$.

It remains to show that the representation as in equation 15 takes place. It is enough to obtain such a representation for the vector $h_n^{(1)}$ from equation 2 because $h_n + \alpha_n = W_O h_n^{(1)} + \alpha_n$, the matrix W_O is a constant $d \times d$ matrix, and $\alpha_n = E(0, n, n)$ does not depend on $x \in \{0,1\}^n$.

[1] It is critical here that there is just 1 attention head; if there were already 2, we would have had two different denominators, and after multiplication, we would no longer have linear inequalities.

Now, the vector $h_n^{(1)}$ is defined as:

$$h_n^{(1)} = \frac{\sum_{i=1}^{n} \exp\{L_{in}^{(1)}\} V^{(1)} \alpha_i}{\sum_{i=1}^{n} \exp\{L_{in}^{(1)}\}}. \tag{16}$$

By definition, expressions

$$\exp\{L_{in}^{(1)}\} V^{(1)} \alpha_i, \qquad \exp\{L_{in}^{(1)}\}$$

are functions of i, n, x_i, x_n. Under the fixation $x_n = 0$, they become functions of just i, n, x_i, and thus can be written as:

$$\exp\{L_{in}^{(1)}\} V^{(1)} \alpha_i = x_i \rho_{in}^1 + (1 - x_i)\rho_{in}^0, \qquad \exp\{L_{in}^{(1)}\} = x_i \theta_{in}^1 + (1 - x_i)\theta_{in}^0$$

for some $\rho_{in}^1, \rho_{in}^0 \in \mathbb{R}^d, \theta_{in}^1, \theta_{in}^0 \in \mathbb{R}$. We see that both the numerator and the denominator in equation 16 depend linearly on $x_1, \ldots, x_n$, and the denominator is a sum of positive numbers for every $x \in \{0, 1\}^n$, as required.

B Proof of Lemma 6

Let G_n be a graph whose nodes are non-ordered n-dimensional binary vectors, and two such vectors x, y if the segment between them intersects the convex hull of the set of ordered binary vectors.

Claim 7. *For any $n \in \mathbb{N}$, the graph G_n is θ_n-colorable.*

Proof. Denote $k = \theta_n$, and let $H_1, \ldots, H_k \subseteq \mathbb{R}^n$ be k halfspaces such that all ordered n-dimensional binary vectors belong to $H_1 \cap H_2 \cap \ldots \cap H_k$, but none of the non-ordered n-dimensional non-ordered binary vectors (i.e., none of the vertices of G_n) does not belong to this intersection. Hence, for every vertex x of G_n there is $i \in \{1, 2, \ldots, k\}$ such that $x \in \mathbb{R}^n \setminus H_i$. Let this index i be the color of x. If there is more than one such index, choose the smallest one.

We have to show that if two vertices x, y of G_n are connected, then they have different colors. That is, we have to show that they cannot both belong to $\mathbb{R}^n \setminus H_i$ for the same $i \in \{1, \ldots, k\}$. Indeed, if x, y both belong to $\mathbb{R}^n \setminus H_i$, then so does the segment between them. Hence, this segment cannot intersect the convex hull of the set of ordered binary vectors, as this convex hull is a subset of $H_1 \cap \ldots \cap H_k$. □

We now give the lower bound on the chromatic number of G_n. We start by observing that G_n has the following subgraph G'_n. The vertices of G'_n are the same as in G_n, where two vertices x, y are connected if and only if $x + y$ is an ordered vector. *(For example, binary vectors:*

$$x = (0, 0, 1, 0, 1), \qquad y = (0, 1, 0, 1, 1),$$

are non-ordered, but their sum:

$$x + y = (0, 1, 1, 1, 2)$$

is an ordered vector). We have to show that G'_n is a subgraph of G_n; that is, if $x, y \in V(G_n)$ and $x + y$ is ordered, then the segment between x and y intersects the convex hull of the set of ordered binary vectors. Indeed, if $x + y$ is ordered, it first has a block of 0s, then a block of 1s, and then a block of 2s (with some of these blocks being potentially empty). Hence, we can write:

$$\frac{x+y}{2} = \left(\underbrace{0, 0, \ldots, 0}_{a}, \underbrace{\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}}_{b}, \underbrace{1, 1, \ldots, 1}_{c} \right)$$

for some $a, b, c \geq 0$. Therefore,

$$\begin{aligned} \frac{x+y}{2} = \frac{1}{2} \cdot \left(\underbrace{0, 0, \ldots, 0}_{a}, \underbrace{0, 0, \ldots, 0}_{b}, \underbrace{1, 1, \ldots, 1}_{c} \right) \\ + \frac{1}{2} \cdot \left(\underbrace{0, 0, \ldots, 0}_{a}, \underbrace{1, 1, \ldots, 1}_{b}, \underbrace{1, 1, \ldots, 1}_{c} \right), \end{aligned}$$

meaning that the midpoint of the segment between x and y intersects the convex hull of the set of binary ordered vectors.

Further, consider a restriction of G'_n to vertices x that consist of precisely 4 blocks of equal bits, starting from the block of 0s, that is, those that have a form:

$$x = \left(\underbrace{0, 0, \ldots, 0}_{a}, \underbrace{1, 1, \ldots, 1}_{b}, \underbrace{0, 0, \ldots, 0}_{c}, \underbrace{1, 1, \ldots, 1}_{d} \right)$$

for some $a, b, c, d > 0$. Each such vertex is uniquely defined by a 3-element subset $\{a, a+b, a+b+c\} \subseteq \{1, 2, \ldots, n-1\}$.

Now, take any $i, j, k, \ell \in \{1, 2, \ldots, n-1\}$ such that:

$$i < j < k < \ell.$$

We claim that vertices x, y that correspond to subsets $\{i, j, k\}$ and $\{j, k, \ell\}$, respectively, are connected in G'_n. Indeed, dividing positions into blocks of lengths

$$i, \qquad j-i, \qquad k-j, \qquad \ell-k, \qquad n-\ell,$$

we can write:

	i	$j-i$	$k-j$	$\ell-k$	$n-\ell$
$x =$	$0 \ldots 0$	$1 \ldots 1$	$0 \ldots 0$	$1 \ldots 1$	$1 \ldots 1$
$y =$	$0 \ldots 0$	$0 \ldots 0$	$1 \ldots 1$	$0 \ldots 0$	$1 \ldots 1$

One can see that the sum of x, y is ordered.

We conclude that the graph G'_n has a subgraph isomorphic to the graph T_n, defined as follows. Vertices of T_n are 3-element subsets of $\{1, 2, \ldots, n\}$. Two subsets are connected if the two largest elements in one coincide with the two smallest elements in the other. It is now enough to show that the chromatic number of T_n goes to $+\infty$ as $n \to +\infty$. That is, for every $q \in \mathbb{N}$, we have to show that there exists $n_0 \in \mathbb{N}$ such that T_n is not q-colorable for every $n \geq n_0$.

This is an easy consequence of the Ramsey theorem for hypergraphs:

Theorem 8 (Erdos & Rado (1952))**.** *Let $s > t \geq 2$ and q be natural numbers. Then there exists n_0 such that for every $n \geq n_0$ the following holds. For every coloring of the t-element subsets of $\{1, 2, \ldots, n\}$ in q colors there exists an s-element subset of $\{1, 2, \ldots, n\}$ such that all its t-element subsets have the same color.*

Fix any q and take the bound for n_0 from the last theorem for $s = 4, t = 3$. Consider any $n \geq n_0$ and assume for contradiction that T_n is q-colorable. Apply the result to the corresponding coloring of the vertices of T_n. There will be a 4-element subset $\{i, j, k, \ell\}$ such that all its 3-element subsets are colored in the same way. However, if $i < i < k < \ell$, then $\{i, j, k\}, \{j, k, \ell\}$ are connected in T_n and hence cannot be colored with the same color, a contradiction.