Appendix

Table of Contents

Notation. The notation $e_i^{d'} \in \mathbb{R}^{d'}$ refers to the one-hot encoding of i in d' dimensions. In other words it is the i^{th} standard basis vector in d' dimensions. The notation $\texttt{Blkdiag}(\{A_1, A_1, \cdots, A_m\})$ refers to the block diagonal matrix with i^{th} block as A_i.

A Proof of Theorem 1

We will first prove Theorem 1. In the first layer, choose the embeddings as,

$$\boldsymbol{x}_n^{(1)} = \texttt{Emb}(x_n) = \kappa \begin{bmatrix} \mathbf{1}_{1\times 2} & e_{x_n}^S & \mathbf{0}_{1\times 2S} \end{bmatrix}^T \in \mathbb{R}^d. \tag{4}$$

for a constant $\kappa > 0$ to be chosen later and $d = 2S + 2$. The relative position encodings will essentially be supported on the first two coordinates, the middle S coordinates are a one-hot encoding of the symbol x_n and the last $2S$ coordinates are 0. The relative position encodings in the first layer are chosen to be $\boldsymbol{p}_{n-i}^{(1),K} = \kappa\left(-1 + \mathbb{I}(n-i=1\})\right) e_1^d \in \mathbb{R}^d$ and $\boldsymbol{p}_{n-i}^{(1),V} = \mathbf{0} \in \mathbb{R}^d$. Choose $\boldsymbol{W}_K^{(1)}$ and $\boldsymbol{W}_Q^{(1)}$ to be $e_1^d (e_1^d)^T \in \mathbb{R}^{d\times d}$. With this choice,

$$\left\langle \boldsymbol{W}_K^{(1)} \left(\boldsymbol{x}_i^{(1)} + \boldsymbol{p}_{n-i}^{(1),K}\right), \boldsymbol{W}_Q^{(1)} \boldsymbol{x}_n^{(1)} \right\rangle = \kappa \mathbb{I}(n-i=1) \tag{5}$$

As $\kappa \to \infty$, the attention pattern (which takes the softmax over of these inner products over $i \in [n]$) computes,

$$\text{att}_{n,i}^{(1)} = \mathbb{I}(i = n-1) \tag{6}$$

for any $n > 1$. Choose the value matrix as,

$$\boldsymbol{W}_V^{(1)} = \begin{bmatrix} \mathbf{0}_{(2+S)\times 2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{S\times S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{d\times d} \tag{7}$$

And with this choice and the residual connection, we get,

$$\boldsymbol{x}_n^{(2)} = \kappa \begin{bmatrix} \mathbf{1}_{1\times 2} & e_{x_n}^S & e_{x_{n-1}}^S & \mathbf{0} \end{bmatrix} \in \mathbb{R}^d \tag{8}$$

which serves as the input to the 2^{nd} transformer layer.

Layer 2. In layer 2, the relative position encodings $\boldsymbol{p}_{n-i}^{K,(2)}$ and $\boldsymbol{p}_{n-i}^{V,(2)}$ are all set as 0. The key matrix picks out the $e^S_{x_n}$ block out of $\boldsymbol{x}_n^{(2)}$ and the query vector picks out the $e^S_{x_{i-1}}$ block out of $\boldsymbol{x}_{i-1}^{(2)}$. In particular, these matrices are chosen so that,

$$\begin{aligned} \boldsymbol{W}_K^{(2)}\boldsymbol{x}_i^{(2)} &= \kappa \begin{bmatrix} \mathbf{1}_{1\times 2} & e^S_{x_{i-1}} & \mathbf{0} \end{bmatrix}^T \in \mathbb{R}^d, \\ \boldsymbol{W}_Q^{(2)}\boldsymbol{x}_n^{(2)} &= \kappa \begin{bmatrix} \mathbf{1}_{1\times 2} & e^S_{x_n} & \mathbf{0} \end{bmatrix}^T \in \mathbb{R}^d \end{aligned} \tag{9}$$

Taking the inner product of these vectors, and taking $\kappa \to \infty$, observe that the attention pattern concentrates on the uniform distribution over all coordinates i such that $x_{i-1} = x_n$. More formally, the attention pattern for any $n > 1$ is,

$$\mathrm{att}_{n,i}^{(2)} = \frac{\mathbb{I}(x_{i-1} = x_n)}{\sum_{i=2}^n \mathbb{I}(x_{i-1} = x_n)}, \tag{10}$$

assuming $\sum_{i=2}^n \mathbb{I}(x_{i-1} = x_n) > 0$. Having realized this attention pattern, may choose the value and subsequent linear layer appropriately. The value matrix simply picks out the $e^S_{x_i}$ block from $\boldsymbol{x}_i^{(2)}$ and places it into the last S coordinates of $\boldsymbol{x}_i^{(3)}$, and the linear layer simply extracts this block and outputs it (after scaling down by a factor of κ), realizing the logits,

$$\mathrm{logit}_n = \frac{1}{\sum_{i=2}^n \mathbb{I}(x_{i-1} = x_n)} \sum_{i=2}^n \mathbb{I}(x_{i-1} = x_n) \cdot e^S_{x_i}. \tag{11}$$

if $\sum_{i=2}^n \mathbb{I}(x_{i-1} = x_n) > 0$. In particular, under the same condition,

$$\mathrm{logit}_T(x_{T+1}) = \frac{\sum_{n=2}^T \mathbb{I}(x_n = x_{T+1}, x_{n-1} = x_T)}{\sum_{i=2}^n \mathbb{I}(x_{n-1} = x_T)} \tag{12}$$

assuming $\sum_{i=2}^n \mathbb{I}(x_{n-1} = x_T)$, which is the conditional 1-gram model.

A.1 Extension to k-heads: Proof of Theorem 2

In the first layer, the embeddings are chosen to be,

$$\boldsymbol{x}_n^{(1)} = \mathtt{Emb}(x_n) = \kappa \left[\ \mathbf{0}_{1\times k} \ \middle|\ 1 \ \middle|\ e^S_{x_n} \ \middle|\ \mathbf{0}_{1\times(k+1)S}\ \right]^T \in \mathbb{R}^d \tag{13}$$

With $d = (k+1)(S+1) + S$. The relative position encodings are chosen as $\boldsymbol{p}_i^{K,(1)} = \begin{bmatrix} e^k_i & \mathbf{0} \end{bmatrix}^T$ for $1 \leqslant i \leqslant k$ and $\boldsymbol{p}_i^{K,(1)} = \mathbf{0}$ otherwise. Similarly, $\boldsymbol{p}_i^{V,(1)} = \mathbf{0}$ for every i. The h^{th} head has key and query matrices,

$$\begin{aligned} \boldsymbol{W}_Q^{(1,h)} &= \begin{bmatrix} \mathbf{0}_{1\times k} & 1 & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \end{bmatrix} \\ \boldsymbol{W}_K^{(1,h)} &= \begin{bmatrix} \mathbf{0}_{1\times(h-1)} & 1 & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \end{bmatrix} \end{aligned} \tag{14}$$

With these choices, and letting $\kappa \to \infty$, the h^{th} layer computes the attention pattern,

$$\mathrm{att}_{n,i}^{(1,h)} = \mathbb{I}(i = n - h). \tag{15}$$

Choose the corresponding value matrix as,

$$\boldsymbol{W}_V^{(1,h)} = \begin{bmatrix} \mathbf{0}_{(2+hS)\times 2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{S\times S} & \mathbf{0} \end{bmatrix} \tag{16}$$

choosing the projection matrix appropriately, the output of the transformer after the first residual connection is,

$$\boldsymbol{x}_n^{(2)} = \kappa \left[\ \mathbf{0}_{1\times k} \ \middle|\ 1 \ \middle|\ e^S_{x_n} \ \middle|\ \cdots \ \middle|\ e^S_{x_{n-k}}\ \right]^T. \tag{17}$$

Layer 2. In this layer, the relative position encodings $\boldsymbol{p}_{n-i}^{K,(2)}$ and $\boldsymbol{p}_{n-i}^{V,(2)}$ are all set as 0. The key and query matrices are chosen as,

$$\begin{aligned}
\boldsymbol{W}_Q^{(2)} &= \begin{bmatrix} \mathbf{0}_{Sk \times k} & I_{(Sk+1)\times(Sk+1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \\
\boldsymbol{W}_K^{(2)} &= \begin{bmatrix} \mathbf{0}_{Sk \times (k+S)} & I_{(Sk+1)\times(Sk+1)} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.
\end{aligned} \tag{18}$$

With this choices, we have that,

$$\left\langle \boldsymbol{W}_K^{(2)} \boldsymbol{x}_i^{(2)}, \boldsymbol{W}_Q^{(2)} \boldsymbol{x}_n^{(2)} \right\rangle = \kappa \sum_{j=1}^{k} \mathbb{I}(x_{i-j} = x_{n-j+1}). \tag{19}$$

Taking $\kappa \to \infty$, observe that the attention pattern concentrates on the uniform distribution over all coordinates i such that $x_{i-j} = x_{n-j+1}$ for all $j \in [k]$. More formally, if $\sum_{i=2}^{n} \mathbb{I}(x_{i-1} = x_n) > 0$, the attention pattern for any $n > 1$ is,

$$\mathrm{att}_{n,i}^{(2)} = \frac{\mathbb{I}(\forall j \in [k],\ x_{i-j} = x_{n-j+1})}{\sum_{i=k+1}^{n} \mathbb{I}(\forall j \in [k],\ x_{i-j} = x_{n-j+1})}. \tag{20}$$

The value matrix picks out $e_{x_i}^S$ from the embedding $\boldsymbol{x}_i^{(2)}$ (Equation (17)) and places it in the last S coordinates. The subsequent linear layer picks out the last S coordinates, resulting in the logits,

$$\mathrm{logit}_n = \sum_{i=k+1}^{n} \frac{\mathbb{I}(\forall j \in [k],\ x_{i-j} = x_{n-j+1})}{\sum_{i=k+1}^{n} \mathbb{I}(\forall j \in [k],\ x_{i-j} = x_{n-j+1})} e_{x_i}^S, \tag{21}$$

assuming that $\sum_{i=k+1}^{n} \mathbb{I}(\forall j \in [k],\ x_{i-j} = x_{n-j+1}) > 0$. In particular,

$$\mathrm{logit}_T(x_{T+1}) = \frac{\sum_{n=k+1}^{T} \mathbb{I}(\forall 0 \leqslant j \leqslant k,\ x_{n-j} = x_{T-j+1})}{\sum_{n=k+1}^{T} \mathbb{I}(\forall 1 \leqslant j \leqslant k,\ x_{n-j} = x_{T-j+1})}, \tag{22}$$

assuming $\sum_{n=k+1}^{T} \mathbb{I}(\forall 1 \leqslant j \leqslant k,\ x_{n-j} = x_{T-j+1}) > 0$, i.e., the conditional k-gram model.

B Proof of Theorem 3

Define $k^\star = 2^{\lceil \log_2(k+1) \rceil}$ by rounding $k+1$ up to the nearest power of 2 and $\ell^\star = \log_2(k^\star)$. In the setting of relative position encodings, given the sequence $x_1, \cdots, x_n$, while generating the output of the attention + feedforward layer for the symbol x_n, the embeddings $\boldsymbol{x}_i = \mathtt{Emb}(x_n) + \boldsymbol{p}_{n-i}$ are used for $i \in [n]$. In other words, the position encoding vector is taken relative to the end of the sequence, rather than the start of the sequence. Consider the embedding of x as,

$$\boldsymbol{x}_n^{(1)} = \mathtt{Emb}(x_n) = \left[\ \mathbf{0}_{1\times \ell^\star}\ \middle|\ 1\ \middle|\ e_{x_n}^S\ \middle|\ \mathbf{0}_{1\times(k^\star-1)S}\ \middle|\ \mathbf{0}_{1\times S}\ \right]^T \in \mathbb{R}^{(k^\star+1)S+\ell^\star+1} \tag{23}$$

where $e_i^{d'} \in \mathbb{R}^S$ is the standard basis vector in d' dimensions. And the relative position encoding for the keys as,

$$\boldsymbol{p}_i^{(1),K} = \begin{cases} \begin{bmatrix} \mathbf{1}_{1\times\ell^\star} & \mathbf{0} \end{bmatrix}^T, & \text{if } i = 0, \\ \begin{bmatrix} e_{1+\log_2(i)}^{\ell^\star} & \mathbf{0} \end{bmatrix}^T & \text{if } i \in \{1, 2, 4, \cdots, k^\star/2\} \\ \mathbf{0}_{d\times 1} & \text{otherwise.} \end{cases} \tag{24}$$

And for the value vectors, $\boldsymbol{p}_i^V = \mathbf{0}$ for all i.

For the first layer and first head, we will describe the value, key and query matrices. Choose,

$$\begin{aligned}
\boldsymbol{W}_K^{(1)} &= \sqrt{\kappa} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \text{ and,} \\
\boldsymbol{W}_Q^{(1)} &= \sqrt{\kappa} \begin{bmatrix} \mathbf{0}_{1\times l^\star} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}.
\end{aligned} \tag{25}$$

Then, observe that for $i \geqslant 1$,

$$\left\langle \boldsymbol{W}_K^{(1)}\big(\boldsymbol{x}_{n-i} + \boldsymbol{p}_i^{(1),K}\big), \boldsymbol{W}_Q^{(1)}\boldsymbol{x}_n \right\rangle = \kappa\mathbb{I}(i = 1)$$

and for $i = 0$,

$$\left\langle \boldsymbol{W}_K^{(1)}\big(\boldsymbol{x}_n + \boldsymbol{p}_0^{(1),K}\big), \boldsymbol{W}_Q^{(1)}\boldsymbol{x}_n \right\rangle = \kappa$$

In particular, letting $\kappa \to \infty$, the attention pattern is,

$$\text{att}_{n,n-i}^{(1)} = \frac{1}{2}\mathbb{I}(i = 0) + \frac{1}{2}\mathbb{I}(i = 1). \tag{26}$$

Choose the value matrix as,

$$\boldsymbol{W}_V^{(1)} = \begin{bmatrix} \mathbf{0}_{(\ell^\star+S)\times\ell^\star} & \mathbf{0} \\ \mathbf{0} & 2I \end{bmatrix}$$

together with the residual connection, we get,

$$\boldsymbol{x}_n^{(2)} = \left[\; \mathbf{0}_{1\times\ell^\star} \;\middle|\; 1 \;\middle|\; e_{x_n}^S \;\middle|\; e_{x_n}^S + e_{x_{n-1}}^S \;\middle|\; \mathbf{0}_{1\times(k^\star-2)S} \;\middle|\; \mathbf{0}_{1\times S} \;\right]^T \tag{27}$$

Layer $\ell + 1$. By induction, assume that the output of the ℓ^{th} transformer layer is of the form,

$$\boldsymbol{x}_n^{(\ell+1)} = \left[\; \mathbf{0}_{1\times\ell^\star} \;\middle|\; 1 \;\middle|\; \boldsymbol{v}_n \;\middle|\; \mathbf{0}_{1\times(k^\star-2^\ell)S} \;\middle|\; \mathbf{0}_{1\times S} \;\right]^T \tag{28}$$

for some vector $\boldsymbol{v}_n \in \mathbb{R}^{2^\ell S}$. We will show that with appropriately chosen key, query and value vectors in the $(\ell+1)^{\text{th}}$ layer, the output of this layer is,

$$\boldsymbol{x}_n^{(\ell+2)} = \left[\; \mathbf{0}_{1\times\ell^\star} \;\middle|\; 1 \;\middle|\; \boldsymbol{v}_n \;\middle|\; \boldsymbol{v}_n + \boldsymbol{v}_{n-2^\ell} \;\middle|\; \mathbf{0}_{1\times(k^\star-2^{\ell+1})S} \;\middle|\; \mathbf{0}_{1\times S} \;\right]^T \tag{29}$$

We will consider the same relative position encodings and query matrix in this layer as in the first layer (Equations (24) and (25)). Consider a key matrix of the form,

$$\boldsymbol{W}_K^{(\ell+1)} = \begin{bmatrix} \mathbf{0}_{1\times\ell} & \sqrt{\kappa} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

With this choice, observe that for $i \geqslant 1$,

$$\left\langle \boldsymbol{W}_K^{(\ell+1)}\big(\boldsymbol{x}_{n-i}^{(\ell+1)} + \boldsymbol{p}_i^{(\ell+1),K}\big), \boldsymbol{W}_Q^{(\ell+1)}\boldsymbol{x}_n^{(\ell+1)} \right\rangle = \kappa \cdot \mathbb{I}(i = 2^\ell)$$

and for $i = 0$,

$$\left\langle \boldsymbol{W}_K^{(\ell+1)}\big(\boldsymbol{x}_n^{(\ell+1)} + \boldsymbol{p}_0^{(\ell+1),K}\big), \boldsymbol{W}_Q^{(\ell+1)}\boldsymbol{x}_n^{(\ell+1)} \right\rangle = \kappa$$

In particular, letting $\kappa \to \infty$, the attention pattern is,

$$\text{att}_{n,n-i}^{(\ell+1)} = \frac{1}{2}\mathbb{I}(i = 0) + \frac{1}{2}\mathbb{I}(i = 2^\ell). \tag{30}$$

Choosing the value matrix as,

$$\boldsymbol{W}_V^{(\ell+1)} = \begin{bmatrix} \mathbf{0}_{(\ell^\star+2^\ell S)\times\ell^\star} & \mathbf{0} \\ \mathbf{0} & 2I \end{bmatrix},$$

we get,

$$\boldsymbol{x}_n^{(\ell+2)} = \left[\; \mathbf{0}_{1\times\ell^\star} \;\middle|\; 1 \;\middle|\; \boldsymbol{v}_n \;\middle|\; \boldsymbol{v}_n + \boldsymbol{v}_{n-2^\ell} \;\middle|\; \mathbf{0}_{1\times(k^\star-2^{\ell+1})S} \;\middle|\; \mathbf{0}_{1\times S} \;\right]^T \tag{31}$$

Final last transformer layer ($\ell = \ell^\star$). The output of the second last transformer layer, indexed $\ell^\star - 1$ is,

$$
\begin{aligned}
\boldsymbol{z}_n^{(\ell^\star)} \triangleq \boldsymbol{x}_n^{(\ell^\star)} &= \left[\begin{array}{c|c|c|c|c} \mathbf{0}_{1\times\ell^\star} & 1 & \boldsymbol{v}_n^{(\ell^\star-1)} & \boldsymbol{v}_n^{(\ell^\star-1)} + \boldsymbol{v}_{n-2^{\ell^\star-1}}^{(\ell^\star-1)} & \mathbf{0}_{1\times S} \end{array}\right]^T \\
&= \left[\begin{array}{c|c|c|c|c} \mathbf{0}_{1\times\ell^\star} & 1 & \boldsymbol{v}_n^{(\ell^\star-1)} & \boldsymbol{v}_n^{(\ell^\star-1)} + \boldsymbol{v}_{n-\frac{k^\star}{2}}^{(\ell^\star-1)} & \mathbf{0}_{1\times S} \end{array}\right]^T,
\end{aligned}
$$

which follows by plugging in the definition of $k^\star$. Note that there exists a linear transformation $\boldsymbol{L}^{(\ell^\star)}$ such that,

$$
\boldsymbol{z}_n^{(\ell^\star-1)} \triangleq \boldsymbol{L}^{(\ell^\star)}\boldsymbol{x}_n^{(\ell^\star)} = \left[\begin{array}{c|c|c|c|c} \mathbf{0}_{1\times\ell^\star} & 1 & \boldsymbol{v}_n^{(\ell^\star-1)} & \boldsymbol{v}_{n-\frac{k^\star}{2}}^{(\ell^\star-1)} & \mathbf{0}_{1\times S} \end{array}\right]^T
$$

This can be further decomposed as,

$$
\begin{aligned}
&\boldsymbol{z}_n^{(\ell^\star-1)} \\
&= \left[\begin{array}{c|c|c|c|c|c|c} \mathbf{0}_{1\times\ell^\star} & 1 & \boldsymbol{v}_n^{(\ell^\star-2)} & \boldsymbol{v}_n^{(\ell^\star-2)} + \boldsymbol{v}_{n-2^{\ell^\star-2}}^{(\ell^\star-2)} & \boldsymbol{v}_{n-\frac{k^\star}{2}}^{(\ell^\star-2)} & \boldsymbol{v}_{n-\frac{k^\star}{2}}^{(\ell^\star-2)} + \boldsymbol{v}_{n-\frac{k^\star}{2}-2^{\ell^\star-2}}^{(\ell^\star-2)} & \mathbf{0}_{1\times S} \end{array}\right]^T
\end{aligned}
$$

And yet again there exists a linear transformation $\boldsymbol{L}^{(\ell^\star-1)}$ which transforms this as,

$$
\begin{aligned}
&\boldsymbol{z}_n^{(\ell^\star-2)} \triangleq \boldsymbol{L}^{(\ell^\star-1)}\boldsymbol{z}_n^{(\ell^\star-1)} \\
&\quad= \left[\begin{array}{c|c|c|c|c|c|c} \mathbf{0}_{1\times\ell^\star} & 1 & \boldsymbol{v}_n^{(\ell^\star-2)} & \boldsymbol{v}_{n-2^{\ell^\star-2}}^{(\ell^\star-2)} & \boldsymbol{v}_{n-\frac{k^\star}{2}-2^{\ell^\star-2}}^{(\ell^\star-2)} & \boldsymbol{v}_{n-\frac{k^\star}{2}-2^{\ell^\star-2}}^{(\ell^\star-2)} & \mathbf{0}_{1\times S} \end{array}\right]^T \\
&\quad= \left[\begin{array}{c|c|c|c|c|c|c} \mathbf{0}_{1\times\ell^\star} & 1 & \boldsymbol{v}_n^{(\ell^\star-2)} & \boldsymbol{v}_{n-\frac{k^\star}{4}}^{(\ell^\star-2)} & \boldsymbol{v}_{n-\frac{k^\star}{2}}^{(\ell^\star-2)} & \boldsymbol{v}_{n-\frac{3k^\star}{4}}^{(\ell^\star-2)} & \mathbf{0}_{1\times S} \end{array}\right]^T
\end{aligned}
\tag{32}
$$

By recursing this argument and composing all the linear transformations, up to a global permutation, we get that,

$$
\begin{aligned}
\prod_{\ell=1}^{\ell^\star} \boldsymbol{L}^{(\ell)}\boldsymbol{x}_n^{(\ell^\star)} &= \left[\begin{array}{c|c|c|c|c|c|c} \mathbf{0}_{1\times\ell^\star} & 1 & \boldsymbol{v}_n^{(1)} & \boldsymbol{v}_{n-1}^{(1)} & \cdots & \boldsymbol{v}_{n-(k^\star-1)}^{(1)} & \mathbf{0}_{1\times S} \end{array}\right]^T \\
&= \left[\begin{array}{c|c|c|c|c|c} \mathbf{0}_{1\times\ell^\star} & 1 & e_{x_n}^S & \cdots & e_{x_{n-(k^\star-1)}}^S & \mathbf{0}_{1\times S} \end{array}\right]^T
\end{aligned}
\tag{33}
$$

In the final layer, we will right multiply the key, query and value matrices by $\boldsymbol{L}^\star = \prod_{\ell=1}^{\ell^\star} \boldsymbol{L}^{(\ell)}$. The effect can be interpreted as operating the original key, query and value matrices on the embedding vectors in Equation (33). In the final layer, we will set all the position encodings to be $\mathbf{0}$ and consider the key and query matrices,

$$
\begin{aligned}
\boldsymbol{W}_K^{(\ell^\star)} &= \sqrt{\kappa}\begin{bmatrix} \mathbf{0}_{Sk\times(\ell^\star+1+S)} & I_{Sk\times Sk} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \\
\boldsymbol{W}_Q^{(\ell^\star)} &= \sqrt{\kappa}\begin{bmatrix} \mathbf{0}_{Sk\times(\ell^\star+1)} & I_{Sk\times Sk} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}
\end{aligned}
\tag{34}
$$

Then,

$$
\left\langle \boldsymbol{W}_K^{(\ell^\star)}\boldsymbol{L}^\star\boldsymbol{x}_{n-i}^{(\ell^\star)}, \boldsymbol{W}_Q^{(\ell^\star)}\boldsymbol{L}^\star\boldsymbol{x}_n^{(\ell^\star)} \right\rangle = \kappa \sum_{j=0}^{k-1} \mathbb{I}(x_{n-j} = x_{i-1-j})
\tag{35}
$$

Where we must be careful to note that the input $\boldsymbol{x}_n^{(\ell^\star)}$ contains copies of $e_{x_n}, e_{x_{n-1}}, \cdots, e_{x_{n-k}}$ since $k^\star \geqslant k+1$ by definition.

Letting $\kappa \to \infty$, if there exists i such that $\sum_{j=0}^{k-1} \mathbb{I}(x_{n-j} = x_{i-j-1}) > 0$, for $n \geqslant k$, the attention pattern is,

$$
\mathrm{att}_{n,i}^{(\ell^\star)} = \frac{\mathbb{I}(x_{i-1} = x_n, x_{i-2} = x_{n-1}, \cdots, x_{i-k} = x_{n-k+1})}{\sum_{i=k}^{n} \mathbb{I}(x_{i-1} = x_n, x_{i-2} = x_{n-1}, \cdots, x_{i-k} = x_{n-k+1})}
\tag{36}
$$

Finally, choose,

$$\boldsymbol{W}_V^{(\ell^\star+2)} = \begin{bmatrix} \mathbf{0}_{(d-S)\times(\ell^\star+1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{S\times(\ell^\star+1)} & I_{S\times S} & \mathbf{0} \end{bmatrix}, \tag{37}$$

we get,

$$\boldsymbol{x}_n^{(\ell^\star+1)} + \sum_{i=k}^{n} \frac{\mathbb{I}(x_{i-1} = x_n, x_{i-2} = x_{n-1}, \cdots, x_{i-k} = x_{n-k+1})}{\sum_{i=k}^{n} \mathbb{I}(x_{i-1} = x_n, x_{i-2} = x_{n-1}, \cdots, x_{i-k} = x_{n-k+1})} \begin{bmatrix} \mathbf{0}_{(d-S)\times 1} \\ e_{x_i} \end{bmatrix} \tag{38}$$

Choosing the subsequent linear layer as,

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{0}_{S\times(d-S)} & I_{S\times S} \end{bmatrix} \tag{39}$$

$$\boldsymbol{b} = \mathbf{0}_{S\times 1} \tag{40}$$

Results in the output,

$$\mathrm{logit}_T(x_{T+1}) = \sum_{n=k}^{T} \frac{\mathbb{I}(x_n = x_{T+1}, x_{n-1} = x_T, x_{n-2} = x_{T-1}, \cdots, x_{n-k} = x_{T-k+1})}{\sum_{n=k}^{T} \mathbb{I}(x_{n-1} = x_T, x_{n-2} = x_{T-1}, \cdots, x_{n-k} = x_{T-k+1})} \tag{41}$$

which is precisely the in-context conditional k-gram.

C Proof of Theorem 4

C.1 Modifying the definition of layer normalization

In every layer, we will perform a simple transformation which is to double the hidden dimension d and add a copy of $-\boldsymbol{x}_n^{(\ell)}$ into the last d coordinates. This is possible by modifying the weights of the transformer appropriately as discussed below. A consequence of this transformation is that the feature mean of the $\boldsymbol{x}_n$'s is $\mu_n = 0$, and therefore the standard deviation σ_n simply normalizes by the L_2-norm of the features. In order to avoid having to explicitly state this transformation at each layer, we will simply redefine the layer norm LN to output $\boldsymbol{v}/\|\boldsymbol{v}\|_2$ for the input vector $\boldsymbol{v}$, which is realized on the first d coordinates of the transformed embeddings.

This transformation can be realized automatically by redefining the initial embeddings $\texttt{Emb}(x_n)$, and modifying the weights of the attention and feedforward subnetworks as follows: The input embeddings are changed to $\begin{bmatrix}\texttt{Emb}(x_n) & -\texttt{Emb}(x_n)\end{bmatrix}^T \in \mathbb{R}^{2d}$. The key and query matrices are chosen to be 0 on the last d coordinates in every layer; the value matrix for $i \geqslant 1$ is transformed to $\texttt{Blkdiag}\big(\{\boldsymbol{W}_V^{(\ell)}, \boldsymbol{W}_V^{(\ell)}\}\big)$, and likewise changing the feedforward layer to the block diagonal matrices $\texttt{Blkdiag}\big(\{\boldsymbol{W}_1^{(\ell)}, \boldsymbol{W}_1^{(\ell)}\}\big)$ and $\texttt{Blkdiag}\big(\{\boldsymbol{W}_2^{(\ell)}, \boldsymbol{W}_2^{(\ell)}\}\big)$. This transformation adds a copy of $-\boldsymbol{x}_n^{(\ell)}$ into the last d coordinates of the corresponding embeddings.

C.2 Notation and supplementary lemmas

For each $i \in [T]$, define,

$$\boldsymbol{v}_i = e_{x_{i-1}} + 3 \cdot e_{x_{i-2}} + \cdots + 3^{k-1} \cdot e_{x_{i-k}} \tag{42}$$

$$\boldsymbol{u}_i = e_{x_i} + 3 \cdot e_{x_{i-1}} + \cdots + 3^{k-1} \cdot e_{x_{i-k+1}} \tag{43}$$

Note that although $\boldsymbol{v}_i = \boldsymbol{u}_{i-1}$, we make the distinction between the two to avoid any confusion in what is stored in the embedding vector at time i and at time $i-1$. Furthermore, define,

$$\mathcal{I}_n = \{k+1 \leqslant i \leqslant n : \forall j \in [k], x_{i-j} = x_{n-j+1}\}. \tag{44}$$

Lemma 1. *If $i \in \mathcal{I}_n$, $\boldsymbol{z}_i = \boldsymbol{z}_{n-1}$. However, if $i \geqslant k+1$ but $i \notin \mathcal{I}_n$, then, $\left\| \frac{\boldsymbol{v}_i}{\|\boldsymbol{v}_i\|_2} - \frac{\boldsymbol{u}_n}{\|\boldsymbol{u}_n\|_2} \right\|_2 \geqslant 3^{-k}$.*

Let $j^\star \in \{0, 1, \cdots, k-1\}$ denote the largest index j such that $x_{n-j} \neq x_{i-j-1}$. Consider the coordinates $a = x_{n-j^\star} \in [S]$ and $b = x_{i-j^\star-1} \in [S]$. Then,

$$\langle \boldsymbol{v}_n, e_a \rangle - \langle \boldsymbol{u}_i, e_a \rangle \geqslant 3^j - \sum_{j=0}^{j^\star-1} 3^j = \frac{3^{j^\star}}{2}, \tag{45}$$

$$\langle \boldsymbol{u}_i, e_b \rangle - \langle \boldsymbol{v}_n, e_b \rangle \geqslant \frac{3^{j^\star}}{2} \tag{46}$$

If $\|\boldsymbol{v}_n\|_2 \geqslant \|\boldsymbol{u}_i\|_2$, then,

$$\left\langle \frac{\boldsymbol{u}_i}{\|\boldsymbol{u}_i\|_2}, e_b \right\rangle - \left\langle \frac{\boldsymbol{v}_n}{\|\boldsymbol{v}_n\|_2}, e_b \right\rangle \geqslant \frac{\langle \boldsymbol{u}_i, e_b \rangle - \langle \boldsymbol{v}_n, e_b \rangle}{\max\{\|\boldsymbol{u}_i\|_2, \|\boldsymbol{v}_n\|_2\}} \geqslant \frac{3^{j^\star}}{2 \cdot \frac{3^k}{2}} = 3^{j^\star - k} \tag{47}$$

This uses the fact that $\boldsymbol{u}_i$ and $\boldsymbol{v}_n$ are coordinate-wise non-negative. On the other hand, if $\|\boldsymbol{v}_n\|_2 \leqslant \|\boldsymbol{u}_i\|_2$, using a similar analysis,

$$\left\langle \frac{\boldsymbol{u}_i}{\|\boldsymbol{u}_i\|_2}, e_a \right\rangle - \left\langle \frac{\boldsymbol{v}_n}{\|\boldsymbol{v}_n\|_2}, e_a \right\rangle \geqslant 3^{j^\star - k}. \tag{48}$$

In either case, there is a coordinate (a or b) such that, $\boldsymbol{u}_i/\|\boldsymbol{u}_i\|_2$ and $\boldsymbol{v}_n/\|\boldsymbol{v}_n\|_2$ differ by at least $3^{j^\star - k}$. This implies the lower bound on the L_2 norm of the difference of the vectors.

C.3 Proof of Theorem 4

Choose the input embeddings as,

$$\boldsymbol{x}_n^{(1)} = \texttt{Emb}(x_n) = \begin{bmatrix} \mathbf{0}_{1\times 3} & e_x^S & \mathbf{0}_{1\times 5S} \end{bmatrix}^T \in \mathbb{R}^{6S+3} \tag{49}$$

In the first two layers we will use the same relative position embeddings, in particular,

$$\boldsymbol{p}_i^{(1),K} = \boldsymbol{p}_i^{(2),K} = \begin{cases} \sqrt{\log(3)} \cdot \begin{bmatrix} 1 & \mathbf{0} \end{bmatrix}^T, & \text{if } i = 0, \\ (i+1)\sqrt{\log(3)} \cdot \begin{bmatrix} 0 & 1 & \mathbf{0} \end{bmatrix}^T, & \text{if } i \in \{1, 2, \cdots, k-1\}, , \\ (k+1)\sqrt{\log(3)} \cdot \begin{bmatrix} 0 & 0 & 1 & \mathbf{0} \end{bmatrix}^T, & \text{if } i = k. \end{cases} \tag{50}$$

and the value embeddings,

$$\boldsymbol{p}_i^{(1),V} = \boldsymbol{p}_i^{(2),V} = \begin{cases} 3^i \begin{bmatrix} 1 & \mathbf{0} \end{bmatrix}^T & \text{for } i \leqslant k \\ \mathbf{0} & i > k. \end{cases} \tag{51}$$

In the final layer, we will drop all position-related information and choose $\boldsymbol{p}_i^{(3),K} = \boldsymbol{p}_i^{(3),V} = \mathbf{0}$ for all i.

Layer 1. Consider the key and query matrices,

$$\begin{aligned} \boldsymbol{W}_K^{(1)} &= \sqrt{\kappa} \cdot \begin{bmatrix} \mathbf{1}_{1\times 2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ \boldsymbol{W}_Q^{(1)} &= \sqrt{\kappa} \cdot \begin{bmatrix} \mathbf{0}_{1\times 3} & \mathbf{1}_{1\times S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \end{aligned} \tag{52}$$

Then, observe that,

$$\left\langle \boldsymbol{W}_K^{(1)} \left(\texttt{Emb}(x_{n-i}) + \boldsymbol{p}_i^{(1),K} \right), \boldsymbol{W}_Q^{(1)} \texttt{Emb}(x_n) \right\rangle = \kappa(i+1)\log(3) \cdot \mathbb{I}(0 \leqslant i \leqslant \min\{n,k\} - 1)$$

Letting $\kappa \to \infty$, this results in the attention pattern,

$$\text{att}_{n,n-i}^{(1)} = \frac{3^i \mathbb{I}(0 \leqslant i \leqslant \min\{n,k\} - 1)}{\sum_{i'=0}^{\min\{n,k\}-1} 3^{i'}} \tag{53}$$

Choose the value matrix as,

$$\boldsymbol{W}_V^{(1)} = \begin{bmatrix} \mathbf{0}_{(S+3)\times 3} & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix}$$

The output of the attention layer (with the residual connection) is,

$$\widetilde{\boldsymbol{x}}_n^{(1)} = \left[\ \mathbf{0}_{1\times 3} \ \middle|\ e_{x_n}^S \ \middle|\ \boldsymbol{u}_n \ \middle|\ \mathbf{0}_{1\times 3S}\ \right]^T, \text{ where, } \boldsymbol{u}_n = \sum_{i=0}^{\min\{n,k\}-1} \text{att}_{n,n-i}\, e_{x_{n-i}}^S. \tag{54}$$

In the feedforward layer to follow, we will choose,

$$\begin{aligned}
\boldsymbol{W}_1^{(1)} &= I \\
\boldsymbol{W}_2^{(1)} &= \begin{bmatrix} \mathbf{0}_{(3+2S)\times(3+S)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{S\times S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}
\end{aligned} \tag{55}$$

Which simply extracts $\boldsymbol{u}_n$ from $\widetilde{\boldsymbol{x}}_n^{(1)}$. With the subsequent layer norm and residual connection, the output of the first layer is,

$$\boldsymbol{x}_n^{(2)} = \left[\begin{array}{c|c|c|c|c} \mathbf{0}_{1\times 3} & e_{x_n}^S & \boldsymbol{u}_n & \frac{\boldsymbol{u}_n}{\|\boldsymbol{u}_n\|_2} & \mathbf{0}_{1\times 3S} \end{array} \right]^T \tag{56}$$

Layer 2. In this layer, the relative position encodings and query matrix are the same as in layer 1 but the key matrix is chosen as,

$$\boldsymbol{W}_K^{(2)} = \sqrt{\kappa} \begin{bmatrix} 0 & \mathbf{1}_{1\times 2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{57}$$

With this choice, observe that,

$$\left\langle \boldsymbol{W}_K^{(2)}(\boldsymbol{x}_{n-i}^{(2)} + \boldsymbol{p}_i^{(1),K}), \boldsymbol{W}_Q^{(2)} \boldsymbol{x}_n^{(2)} \right\rangle = \kappa(i+1)\log(3) \cdot \mathbb{I}(1 \leqslant i \leqslant k) \tag{58}$$

As before, since $\kappa \to \infty$, this results in the attention pattern,

$$\text{att}_{n,n-i}^{(2)} = \frac{3^i \mathbb{I}(1 \leqslant i \leqslant \min\{k, n-1\})}{\sum_{i'=1}^{\min\{k,n-1\}} 3^{i'}} \tag{59}$$

which is similar, but subtly different from the attention pattern in the first layer (Equation (53)). The first layer focuses on indices $n-i$ such that $0 \leqslant i \leqslant k-1$, while this layer focuses on $1 \leqslant i \leqslant k$. Choosing the value and projection matrices as,

$$\boldsymbol{W}_V^{(2)} = \begin{bmatrix} I_{3\times 3} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{3S\times 3} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{S\times S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{60}$$

The output of the attention layer (with the first residual connection) is,

$$\begin{aligned}
\widetilde{\boldsymbol{x}}_n^{(2)} &= \left[\begin{array}{c|c|c|c|c|c|c} Z_n & \mathbf{0}_{1\times 2} & e_{x_n}^S & \boldsymbol{u}_n & \frac{\boldsymbol{u}_n}{\|\boldsymbol{u}_n\|_2} & \boldsymbol{v}_n & \mathbf{0}_{1\times 2S} \end{array} \right]^T, \\
\text{where, } \boldsymbol{v}_n &= \sum_{i=1}^{\min\{k,n-1\}} \text{att}_{n,n-i}\, e_{x_{n-i}}^S, \\
\text{and, } Z_n &= \sum_{i=1}^{\min\{k,n-1\}} \text{att}_{n,n-i}\, 3^i,
\end{aligned} \tag{61}$$

It is a short calculation to see that $Z_n = 3^{k+1}/5$ if $n \geqslant k+1$ and otherwise, $Z_n \leqslant 3^k/5$. This will be useful later, since the value of Z_n can be used to determine whether $n \geqslant k+1$ or $n \leqslant k$ which will allow the the next layer to avoid calculating the attention at $i \leqslant k$, where the evaluation $x_n = x_{i-1}, \cdots, x_{n-k+1} = x_{i-k}$ is not well defined. In the subsequent FFN layer, we will choose,

$$\begin{aligned}
\boldsymbol{W}_1^{(2)} &= I \\
\boldsymbol{W}_2^{(2)} &= \begin{bmatrix} \mathbf{0}_{(3+4S)\times(3+3S)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{S\times S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{S\times 2S} \end{bmatrix}
\end{aligned} \tag{62}$$

Which extracts $\boldsymbol{v}_n$ from the embedding $\widetilde{\boldsymbol{x}}_n^{(2)}$. With the layer norm and adding the final residual connection, the output of this layer is,

$$\boldsymbol{x}_n^{(3)} = \left[\begin{array}{c|c|c|c|c|c|c|c} Z_n & \mathbf{0}_{2\times 1} & e_{x_n}^S & \boldsymbol{u}_n & \frac{\boldsymbol{u}_n}{\|\boldsymbol{u}_n\|_2} & \boldsymbol{v}_n & \frac{\boldsymbol{v}_n}{\|\boldsymbol{v}_n\|_2} & \mathbf{0}_{S\times 1} \end{array} \right]^T \tag{63}$$

Layer 3. In this layer, all the relative position encodings are set as $\mathbf{0}$ and instead,

$$\begin{aligned} \boldsymbol{W}_Q^{(3)} &= \sqrt{2\kappa}\begin{bmatrix} 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{S\times(2+3S)} & I_{S\times S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \\ \boldsymbol{W}_K^{(3)} &= \sqrt{2\kappa}\cdot\begin{bmatrix} 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{S\times(2+4S)} & I_{S\times S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \end{aligned} \tag{64}$$

With these choices,

$$\begin{aligned} \left\langle \boldsymbol{W}_K^{(3)}\boldsymbol{x}_i^{(3)}, \boldsymbol{W}_Q^{(3)}\boldsymbol{x}_n^{(3)} \right\rangle &= 2\kappa Z_i Z_n + \frac{2\kappa\langle \boldsymbol{v}_i, \boldsymbol{u}_n\rangle}{\|\boldsymbol{v}_i\|_2\cdot\|\boldsymbol{u}_n\|_2} \\ &= 2\kappa Z_i Z_n + 2\kappa - \kappa\left\| \frac{\boldsymbol{v}_i}{\|\boldsymbol{v}_i\|_2} - \frac{\boldsymbol{u}_n}{\|\boldsymbol{u}_n\|_2} \right\|^2 \end{aligned} \tag{65}$$

The resulting attention scores are,

$$\mathrm{att}_{n,i}^{(3)} \propto \exp\left(-\kappa\left\| \frac{\boldsymbol{v}_i}{\|\boldsymbol{v}_i\|_2} - \frac{\boldsymbol{u}_n}{\|\boldsymbol{u}_n\|_2} \right\|^2 + 2\kappa Z_i Z_n \right) \tag{66}$$

Recall that $\mathcal{I}_n = \{k+1 \leqslant i \leqslant n : \forall j \in [k],\ x_{n-j+1} = x_{i-j}\}$. Then for any $i \in \mathcal{I}_n$, $\boldsymbol{v}_i = \boldsymbol{u}_n$, and by Lemma 1, for any $i \geqslant k+1$ but not in $\mathcal{I}_n$,

$$\left\| \frac{\boldsymbol{v}_i}{\|\boldsymbol{v}_i\|_2} - \frac{\boldsymbol{u}_n}{\|\boldsymbol{u}_n\|_2} \right\|_2 \geqslant \frac{1}{3^k}.$$

Note that this gap is small but non-zero. Furthermore, recall that $Z_i = 3^{k+1}/5$ if $i \geqslant k$ and otherwise $Z_i \leqslant 3^k/5$. Thus the attention prefers values of i such that $\boldsymbol{v}_i = \boldsymbol{u}_n$ and such that $i \geqslant k+1$. In particular, as $\kappa \to \infty$, the resulting attention pattern is,

$$\mathrm{att}_{n,\cdot}^{(3)} = \mathrm{Unif}(\mathcal{I}_n). \tag{67}$$

Choosing,

$$\boldsymbol{W}_V^{(3)} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{S\times 3} & I_{S\times S} & \mathbf{0} \end{bmatrix}.$$

We get that,

$$\widetilde{\boldsymbol{x}}_n^{(3)} = \boldsymbol{x}_n^{(3)} + \sum_{i=1}^{n} \mathrm{att}_{n,i}^{(3)} \begin{bmatrix} \mathbf{0} \\ e_{x_i}^S \end{bmatrix} = \boldsymbol{x}_n^{(3)} + \frac{1}{|\mathcal{I}_n|}\sum_{i\in\mathcal{I}_n} \begin{bmatrix} \mathbf{0} \\ e_{x_i}^S \end{bmatrix}.$$

The feedforward layer is chosen to have $\boldsymbol{W}_1^{(3)} = \boldsymbol{W}_2^{(3)} = \mathbf{0}$, and the overall output of the final transformer layer is therefore just $\widetilde{\boldsymbol{x}}_n^{(3)}$. In the output linear layer, choose,

$$\begin{aligned} \boldsymbol{A} &= \begin{bmatrix} \mathbf{0}_{S\times(d-S)} & I_{S\times S} \end{bmatrix} \\ \boldsymbol{b} &= \mathbf{0} \end{aligned} \tag{68}$$

which results in,

$$\mathrm{logit}_n = \frac{1}{|\mathcal{I}_n|}\sum_{i\in\mathcal{I}_n} e_{x_i}^S = \sum_{i=k+1}^{n} \frac{\mathbb{I}(\forall 1 \leqslant j \leqslant k,\ x_{i-j} = x_{n-j+1})}{\sum_{i'=k+1}^{n}\mathbb{I}(\forall 1 \leqslant j \leqslant k,\ x_{i'-j} = x_{n-j+1})} \cdot e_{x_i}$$

In particular,

$$\mathrm{logit}_T(x_{T+1}) = \frac{\sum_{n=k+1}^{T}\mathbb{I}(\forall 0 \leqslant i \leqslant k,\ x_{n-i} = x_{T-i+1})}{\sum_{n=k+1}^{T}\mathbb{I}(\forall 1 \leqslant i \leqslant k,\ x_{n-i} = x_{T-i+1})} \tag{69}$$

which is the conditional k-gram.

D Representation lower bounds for 1-layer transformers: Proof of Theorem 5

We prove this lower bound by a reduction to communication complexity, and specifically to the set disjointness problem.

Suppose Alice and Bob are given strings $\boldsymbol{a}, \boldsymbol{b} \in \{0,1\}^n$ which are indicator vectors of sets A and B. Their goal is to jointly compute $\mathrm{DIS}(\boldsymbol{a}, \boldsymbol{b}) = \mathbb{I}(\exists i : \boldsymbol{a}_i = \boldsymbol{b}_i = 1)$, which indicates whether A and B intersect or not. Alice and Bob may send a single bit message to the other party over a sequence of communication rounds. The following seminal result by [29] asserts a lower bound on amount of communication required between Alice and Bob to carry out this task.

Theorem 7 ([29])**.** *Any deterministic protocol for computing* $\mathrm{DIS}(\boldsymbol{a}, \boldsymbol{b})$ *requires at least n rounds of communication.*

We show that a 1-layer transformer with sufficiently small embedding dimension / number of heads can be used to simulate a two-way communication protocol between Alice and Bob to solve $\mathrm{DIS}(\boldsymbol{a}, \boldsymbol{b})$ in a way which contradicts Yao's lower bound in Theorem 7.

With $m = T/3 - 1$, suppose Alice and Bob have length m bit strings $\boldsymbol{a}, \boldsymbol{b} \in \{0,1\}^m$. The transformer's input will be a sequence of the form,

$$2, \boldsymbol{a}_1, \boldsymbol{b}_1, 2, \boldsymbol{a}_2, \boldsymbol{b}_2, \cdots, 2, \boldsymbol{a}_m, \boldsymbol{b}_m, 2, 1, \tag{70}$$

of length $3m + 2 = T - 1$. The input basically contains a repeating motif, composed of the symbol 2 followed by one of Alice's bits, and then one of Bob's bits. The last 2 symbols are 2 and 1. We will consider the empirical conditional 3-gram probability the transformer associates with the symbol $x_T = 2$. Noting that $x_{T-1} = 1$ and $x_{T-2} = 1$, the conditional 3-gram is computed to be,

$$\frac{\sum_{i=3}^{T-1} \mathbb{I}(x_i = 1, x_{i-1} = 1, x_{i-2} = 2)}{\sum_{i=3}^{T-1} \mathbb{I}(x_{i-1} = 1, x_{i-2} = 2)} \tag{71}$$

Note that if $x_{i-2} = 2$, then i must be of the form $3j$ for $j = 1, \cdots, n$, and we may rewrite the sum as,

$$\frac{\sum_{j=1}^{m} \mathbb{I}(x_{3j} = 1, x_{3j-1} = 1)}{\sum_{j=1}^{m} \mathbb{I}(x_{3j-1} = 1)} = \frac{|A \cap B|}{|B|} \tag{72}$$

Now, let us use the transformer to construct a deterministic communication protocol between Alice and Bob. Alice is given $(x_2, x_5, \cdots, x_{3m-1}) = (\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_m)$ and Bob is given $(x_3, x_6, \cdots, x_{3m}) = (\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_m)$.

In the first round, Alice computes the normalization in the softmax of the attention which comes from the set of inputs she holds. For simplifying notation define,

$$\mathrm{score}^{(h)}(i) = \exp\left(\left\langle \boldsymbol{W}_K^{(h)}(\mathtt{Emb}(x_i) + \boldsymbol{p}_i), \boldsymbol{W}_Q^{(h)}\mathtt{Emb}(x_{T-1})\right\rangle\right) \tag{73}$$

In particular, for each head $h \in [H]$, she computes,

$$Z_{\text{Alice}}^{(h)} = \log\left(\sum_{j=1}^{m} \mathrm{score}^{(h)}(3j-1)\right) \tag{74}$$

Assuming that the transformer uses p bits of precision, Alice communicates $Z_{\text{Alice}}^{(h)}$ for each h, which corresponds to pH bits of communication. With this information, Bob completes the rest of the normalization term (again up to p bits of precision) and computes,

$$Z^{(h)} = \log\left(Z_{\text{Alice}}^{(h)} + Z_{\text{Bob}}^{(h)} + Z_{\text{common}}^{(h)}\right), \tag{75}$$

$$\text{where } Z_{\text{Bob}}^{(h)} = \log\left(\sum_{j=1}^{m} \mathrm{score}^{(h)}(3j)\right) \tag{76}$$

$$\text{and } Z_{\text{common}}^{(h)} = \log\left(\sum_{j=1}^{m} \mathrm{score}^{(h)}(3j-2) + \mathrm{score}^{(h)}(T-2) + \mathrm{score}^{(h)}(T-1)\right) \tag{77}$$

which is the overall normalization term in the softmax. This is communicated back to Alice, using another pH bits of communication. Next using this information, Alice computes the output of the attention layer, taking the convex combination corresponding to the inputs she knows. In particular, for each $h \in [H]$ she computes,

$$\sum_{j=1}^{n} \frac{\text{score}^{(h)}(3j-1)}{\exp(Z^{(h)})} \texttt{Emb}(x_{3j-1}) \in \mathbb{R}^d. \tag{78}$$

across all the heads. Rather than transmitting everything, she concatenates the outputs of the heads, and multiplies them by the value and projection matrices to result in the output $\boldsymbol{y}_{\text{Alice}}$ which is d-dimensional. This is sent to Bob using dp bits of communication. Subsequently, Bob computes the terms in the attention corresponding to the inputs he knows as well as the public inputs (all the 2's at positions $3j-2$ as well as the last two symbols). In particular,

$$\sum_{j=1}^{m} \frac{\text{score}^{(h)}(3j)}{\exp(Z^{(h)})} \texttt{Emb}(x_{3j}) + \sum_{j=1}^{m+1} \frac{\text{score}^{(h)}(3j-2)}{\exp(Z^{(h)})} \texttt{Emb}(2) + \frac{\text{score}^{(h)}(T-1)}{\exp(Z^{(h)})} \texttt{Emb}(1) \tag{79}$$

These are yet again concatenated across all the heads and multiplied by the value and projection matrices to result in the output $\boldsymbol{y}_{\text{Bob}}$ which is added to $\boldsymbol{y}_{\text{Alice}}$ to result in $\boldsymbol{y}$. Bob passes $\boldsymbol{y}$ through the residual connection, layer norm, and feedforward layers, and subsequently through the linear layer and softmax of the model to result in the output of the model. By assumption, the output of the model approximately captures the conditional 3-gram, which by Equation (72) equals $|A \cap B|/|B|$. Note that if $|A \cap B|/|B|$ is non-zero, it must be at least $1/T$. This means, if the transformer is able to compute the conditional 3-gram to within an additive error of $1/3T$, then Bob can simply threshold the output of the transformer to decide whether $A \cap B = \varnothing$ or not, thereby solving $\text{DIS}(\boldsymbol{a}, \boldsymbol{b})$.

Since this communication protocol is deterministic, by Yao's lower bound in Theorem 7, the number of bits communicated between Alice and Bob must be at least $m = T/3 - 1$. The total number of bits of communication in the protocol is $2pH + dp + 1$ (the last 1 comes from Bob having to communicate the answer to Alice), completing the proof.

E Lower bounds on representing k^{th}-order induction heads: Proof of Theorem 6

In this section we prove the size-lower bound on attention-only transformers representing k^{th}-order induction heads in Theorem 6. To enable this result to be better interpreted, we will break it down into two corollaries.

Corollary 1. *Consider an L-layer attention-only transformer with 1 head per layer and relative position encodings, which satisfies Assumption 1. If $L \leqslant 1 + \log_2(k-2)$, the attention pattern in layer L of the transformer cannot represent a k^{th}-order induction head.*

Corollary 2. *Consider an 2-layer attention-only transformer with H heads in the first layer and relative position encodings, and assume that Assumption 1 is satisfied. If $H \leqslant k-3$, the attention pattern in the 2^{nd} layer cannot represent a k^{th}-order induction head.*

We will first prove the result for the case $L = 2$ and $H = 1$, which falls in the intersection of both of these corollaries. We will show that these models cannot represent k^{th}-order induction heads for $k > 3$, under Assumption 1. We subsequently extend it to the general L-layer transformer (i.e., Corollary 1) in Appendix E.2 and to the general case with H_ℓ heads in layer $\ell \in [L]$ in Appendix E.3.

E.1 Lower bounds on 2-layer 1-head attention-only transformers

In this section we show that under Assumption 1, a 2-layer 1-head attention-only transformer cannot represent k^{th}-order induction heads for any $k \geqslant 4$. We will prove lower bounds on the transformer when the input is binary, i.e., $S = \{0, 1\}$. With relative position embeddings, observe that the first layer of the transformer model learns representations of the form,

$$\boldsymbol{x}_n^{(2)} = \texttt{Emb}(x_n) + \sum_{i \leqslant n} \text{att}_{n,i}^{(1)} \boldsymbol{W}_V^{(1)} \texttt{Emb}(x_i) + \sum_{i \leqslant n} \boldsymbol{W}_V^{(1)} \boldsymbol{p}_{n-i}^{V,(1)} \tag{80}$$

where note that the attention pattern only depends on n and i and not on x_i or x_n. These representations are input into the second layer, which realizes the attention pattern $\text{att}_{n,i}^{(2)}$, which is proportional to,

$$\exp\left(\left\langle \boldsymbol{W}_K^{(2)}\big(\boldsymbol{x}_i^{(2)} + \boldsymbol{p}_{n-i}^{K,(2)}\big), \boldsymbol{W}_Q^{(2)}\boldsymbol{x}_n^{(2)}\right\rangle\right). \tag{81}$$

We need this function to be maximized uniquely when $x_{i-1} = x_n, \cdots, x_{i-k} = x_{n-k+1}$. Denoting $\phi(0) = \boldsymbol{W}_V^{(1)}\texttt{Emb}(0)$ and $\phi(1) = \boldsymbol{W}_V^{(1)}\texttt{Emb}(1)$,

$$\boldsymbol{x}_n^{(2)} = \texttt{Emb}(x_n) + \sum_{i\leqslant n} \mathrm{att}_{n,i}^{(1)}\, \boldsymbol{W}_V^{(1)}\texttt{Emb}(x_i) + \sum_{i\leqslant n} \boldsymbol{W}_V^{(1)}\boldsymbol{p}_{n-i}^{(1),V} \tag{82}$$

$$= x_n\texttt{Emb}(1) + (1-x_n)\texttt{Emb}(0) + \sum_{i\leqslant n} \mathrm{att}_{n,i}^{(1)}\big(x_i\cdot\phi(1) + (1-x_i)\cdot\phi(0)\big) + \sum_{i\leqslant n}\boldsymbol{W}_V^{(1)}\boldsymbol{p}_{n-i}^{(1),V} \tag{83}$$

$$= \left(\frac{\texttt{Emb}(1)+\texttt{Emb}(0)}{2} + x_n'\cdot\frac{\texttt{Emb}(1)-\texttt{Emb}(0)}{2}\right) + \sum_{i\leqslant n}\mathrm{att}_{n,i}^{(1)}\left(\frac{\phi(1)+\phi(0)}{2} + x_i'\cdot\frac{\phi(1)-\phi(0)}{2}\right) + \sum_{i\leqslant n}\boldsymbol{W}_V^{(1)}\boldsymbol{p}_{n-i}^{(1),V} \tag{84}$$

where $x_i' \leftarrow 2x_i - 1$. We can write this down as,

$$\boldsymbol{x}_n^{(2)} = \boldsymbol{m}_n^{(1)} + \boldsymbol{M}_n^{(1)}\begin{bmatrix} x_n' & x_{n-1}' & \cdots & x_1'\end{bmatrix}^T \tag{85}$$

where $\boldsymbol{M}_n^{(1)}$ is a matrix of rank at most 2 and of the form,

$$\boldsymbol{M}_n^{(1)} = \left(\frac{\phi(1)-\phi(0)}{2}\right)\begin{bmatrix}\mathrm{att}_{n,n}^{(1)} & \cdots & \mathrm{att}_{n,1}^{(1)}\end{bmatrix} + \left(\frac{\texttt{Emb}(1)-\texttt{Emb}(0)}{2}\right)\begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix} \tag{86}$$

which is independent of $x_1', \cdots, x_n'$. Likewise $\boldsymbol{m}_n^{(1)}$ collects all the vectors in the sum that don't depend on $x_1', \cdots, x_n'$. Now, observe that in the next layer, we wish to show that an induction head cannot be realized by $\mathrm{att}_{n,i}^{(2)}$ for each $i \leqslant n$. We will show this for any value of $i \leqslant n-k$.

In the second layer, we may write down the key vectors as,

$$\boldsymbol{W}_K^{(2)}\left(\boldsymbol{x}_i^{(2)} + \boldsymbol{p}_{n-i}^{(2),K}\right) = \boldsymbol{W}_K^{(2)}\boldsymbol{m}_i^{(1)} + \boldsymbol{W}_K^{(2)}\boldsymbol{M}_i^{(1)}\begin{bmatrix}x_i' & x_{i-1}' & \cdots & x_1'\end{bmatrix}^T + \boldsymbol{W}_K^{(2)}\boldsymbol{p}_{n-i}^{(2),K}. \tag{87}$$

Again, defining the vector $\overline{\boldsymbol{m}}_i^{(1)}$ and the matrix $\overline{\boldsymbol{M}}_i^{(1)}$ appropriately (having rank at most 2), this equals,

$$\overline{\boldsymbol{m}}_i^{(1)}(\{x_i'\}\cup\{x_{-k-1}', \cdots, x_1'\}) + \overline{\boldsymbol{M}}_i^{(1)}\boldsymbol{y} \tag{88}$$

where $\boldsymbol{y} \triangleq \begin{bmatrix}x_{i-1}' & \cdots & x_{i-k}'\end{bmatrix}^T$ and the vector $\overline{\boldsymbol{m}}_i^{(1)}$ depends on x_i' as well as the inputs $x_{i-k-1}', \cdots, x_1'$, which in this context, are treated as nuisance variables since they do not intersect with $\{x_{i-1}', \cdots, x_{i-k}'\} \cup \{x_n, \cdots, x_{n-k+1}\}$. Henceforth we will avoid explicitly stating the dependency of $\overline{\boldsymbol{m}}_i^{(1)}$ on the x_j's. Similarly, the query vector can be written down as,

$$\boldsymbol{W}_Q^{(2)}\boldsymbol{x}_n^{(2)} = \widehat{\boldsymbol{m}}_n^{(1)} + \breve{\boldsymbol{M}}_n^{(1)}\boldsymbol{x} + \widehat{\boldsymbol{M}}_n^{(1)}\boldsymbol{y} \tag{89}$$

where $\widehat{\boldsymbol{m}}_n^{(1)}$, $\breve{\boldsymbol{M}}_n^{(1)}$ and $\widehat{\boldsymbol{M}}_n^{(1)}$ are defined appropriately, with $\widehat{\boldsymbol{M}}_n^{(1)}$ and $\widehat{\boldsymbol{M}}_n^{(1)}$ of rank at most 2, and $\boldsymbol{x}$ is defined as $\begin{bmatrix}x_n' & \cdots & x_{n-k+1}'\end{bmatrix}^T$. For an appropriate matrix $\boldsymbol{M}_{n,i}^{\times}$, vectors $\boldsymbol{m}_{n,i}^{\times}$ and $\widetilde{\boldsymbol{m}}_{n,i}^{\times}$ and scalar $m_{n,i}^{\times}$, the dot-product of the key and query vectors can be written as,

$$\left\langle \boldsymbol{W}_K^{(2)}\big(\boldsymbol{x}_i^{(2)} + \boldsymbol{p}_{n-i}^{(2),K}\big), \boldsymbol{W}_Q^{(2)}\boldsymbol{x}_n^{(2)}\right\rangle = \boldsymbol{x}^T\boldsymbol{M}_{n,i}^{\times}\boldsymbol{y} + \boldsymbol{y}^T\big(\overline{\boldsymbol{M}}_{n,i}^{\times}\big)\boldsymbol{y} + (\boldsymbol{m}_{n,i}^{\times})^T\boldsymbol{x} + (\overline{\boldsymbol{m}}_{n,i}^{\times})^T\boldsymbol{y} + m_{n,i}^{\times} \triangleq f_{n,i}(\boldsymbol{x}, \boldsymbol{y}), \tag{90}$$

Which is a linear function in $\boldsymbol{x}$ and quadratic in $\boldsymbol{y}$, both of which lie on $\{\pm 1\}^k$. Note that the matrix $\boldsymbol{M}_{n,i}^{\times}$ has rank at most 2 since it is a product of $\overline{\boldsymbol{M}}_i^{(1)}$ and $\breve{\boldsymbol{M}}_n^{(1)}$, each with rank at most 2. Next we introduce a lemma showing that if $\boldsymbol{M}_{n,i}^{\times}$ is inherently low rank, the quadratic form in Equation (90) which captures the dot-product between the key and value vectors cannot satisfy the property that for every $\boldsymbol{y}$, the function is uniquely maximized at $\boldsymbol{x} = \boldsymbol{y}$. In particular, this means that for any $i \leqslant n-k$, there is some choice of $x_n, x_{n-1}, \cdots, x_{n-k+1}$ such that there are $x_{i-1}, \cdots, x_{i-k}$ such that for at least one $j \in [k]$, x_{i-j} and x_{n-j-1} are not equal, but the attention score is larger than the case when x_{i-j} were equal to x_{n-j-1} for each $j \in [k]$.

Lemma 2. *If $M^{\times}_{n,i}$ has rank $\leqslant k-2$, it is impossible for $f_{n,i}(\boldsymbol{x},\boldsymbol{y})$ to satisfy the property that for every $\boldsymbol{y} \in \{\pm 1\}^k$, the maximizer is uniquely $\boldsymbol{x} = \boldsymbol{y}$.*

The proof is almost complete: if $k \geqslant 4$, then the rank of $\boldsymbol{M}^{\times}_{n,i}$, which is at most 2, does not exceed $k-2$. This means that when $k \geqslant 4$, any attention pattern realized in the second layer must satisfy the property that there exists a string such that the attention is no longer uniquely maximized when $x_n = x_{i-1}, \cdots, x_{n-k+1} = x_{i-k}$.

Proof. For the purpose of brevity, define $\mathcal{H}_k = \{\pm 1\}^k$. First consider the reparameterization,

$$\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{M}}^{\times}_{n,i}\boldsymbol{x}, \text{ where } \widetilde{\boldsymbol{M}}^{\times}_{n,i} = \begin{bmatrix} (\boldsymbol{M}^{\times}_{n,i})^T \\ (\boldsymbol{m}^{\times}_{n,i})^T \end{bmatrix}. \tag{91}$$

Then, the dot-product of the key and query matrices can be written as,

$$\begin{bmatrix} \boldsymbol{y}^T & 1 \end{bmatrix} \widetilde{\boldsymbol{x}} + \boldsymbol{y}^T (\overline{\boldsymbol{M}}^{\times}_{n,i}) \boldsymbol{y} + (\overline{\boldsymbol{m}}^{\times}_{n,i})^T \boldsymbol{y} + m^{\times}_{n,i} \tag{92}$$

Note that this function is linear in $\widetilde{\boldsymbol{x}}$ and therefore must be maximized on a vertex of the convex hull of the domain, $\widetilde{\boldsymbol{M}}^{\times}_{n,i}\mathcal{H}_k \triangleq \{\widetilde{\boldsymbol{M}}^{\times}_{n,i}\boldsymbol{h} : \boldsymbol{h} \in \mathcal{H}_k\}$. If $\boldsymbol{M}^{\times}_{n,i}$ has rank at most $k-2$, the rank of $\widetilde{\boldsymbol{M}}^{\times}_{n,i}$ is at most $k-1$ and cannot be full rank. We show that this must imply that there is a vertex $\boldsymbol{v} \in \mathcal{H}_k$ such that $\widetilde{\boldsymbol{M}}^{\times}_{n,i}\boldsymbol{v}$ is not a unique vertex of the convex hull of $\widetilde{\boldsymbol{M}}^{\times}_{n,i}\mathcal{H}_k$. This means that $\boldsymbol{v}$ cannot be a unique maximizer for $\widetilde{\boldsymbol{x}}$ when maximizing over all strings in Equation (92), and specifically $\boldsymbol{y} = \boldsymbol{v}$ is a witness to Lemma 2.

Below we discuss how to find such a vector $\boldsymbol{v}$. Note that $\widetilde{\boldsymbol{M}}^{\times}_{n,i}$ is not full rank, which implies that there exists a vector $\boldsymbol{n}$ such that $\widetilde{\boldsymbol{M}}^{\times}_{n,i}\boldsymbol{n} = \boldsymbol{0}$. Without loss of generality, let $\boldsymbol{n}_1$ be the smallest non-zero coordinate of $\boldsymbol{n}$ in absolute value. Then the vector $\boldsymbol{n}_1^{-1}\boldsymbol{n}$ has no non-zero coordinates in the interval $(-1,1)$. We will show that $\mathrm{sign}(\boldsymbol{n}_1^{-1}\boldsymbol{n})$ is a good choice for $\boldsymbol{v}$.

Consider two cases,

Case I. Every non-zero coordinate of $\boldsymbol{n}_1^{-1}\boldsymbol{n}$ is in $\{\pm 1\}$. Consider any $\boldsymbol{x} \in \mathcal{H}_k$ which matches with $\boldsymbol{n}$ on the non-zero coordinates. Consider $\boldsymbol{x}'$ which is the same as $\boldsymbol{x}$, except a negation is taken on the coordinates where $\boldsymbol{n}$ is non-zero. Note that $\widetilde{\boldsymbol{M}}^{\times}_{n,i}\boldsymbol{x} = \widetilde{\boldsymbol{M}}^{\times}_{n,i}\boldsymbol{x}'$, for the same value of $\boldsymbol{x}$. This means that for any $\boldsymbol{y}$. In particular, from Equation (92), both $\boldsymbol{x}$ and $\boldsymbol{x}'$ are maximizers, showing that Lemma 2 is true in this case. We circumvent having to find such a vector $\boldsymbol{v}$ in this case.

Case II. $\boldsymbol{n}_1^{-1}\boldsymbol{n}$ has non-zero coordinates which are not all in $\{\pm 1\}$. In particular, at least one coordinate where this vector is strictly less than -1 or strictly greater than $+1$. In this case, observe that the sign vector $\widetilde{\boldsymbol{n}} = \mathrm{sign}(\boldsymbol{n}_1^{-1}\boldsymbol{n}) \in \mathcal{H}_k$ lies within, but is not a vertex of the convex hull of the set $\mathcal{H}_k \cup \{\boldsymbol{n}_1^{-1}\boldsymbol{n}\}$. The reason for this is simple to see when we assume that $\boldsymbol{n}_1^{-1}\boldsymbol{n}$ has only one coordinate which is not in $[-1,1]$, say, the coordinate $j=2$: here, $\widetilde{\boldsymbol{n}}$ can be written down as a convex combination (with non-zero coefficients) of $\boldsymbol{n}_1^{-1}\boldsymbol{n}$ and $\widetilde{\boldsymbol{n}}^{(2)}$; the latter vector is obtained by flipping coordinate 2 of $\widetilde{\boldsymbol{n}}$. When there is more than one coordinate not in $[-1,1]$, we can peel away these large coordinates in $\boldsymbol{n}_1^{-1}\boldsymbol{n}$ by taking a convex combination of this vector with the vectors $\widetilde{\boldsymbol{n}}^{(j)}$ for the appropriate values of j, to return the sign vector $\widetilde{\boldsymbol{n}}$. Here, $\widetilde{\boldsymbol{n}}^{(j)}$ is the version of $\widetilde{\boldsymbol{n}}$ where the j^{th}-coordinate is flipped. This results in the following claim.

Claim 1. *The sign vector $\widetilde{\boldsymbol{n}}$ lies within the convex hull of the points $\mathcal{H}_k \cup \{\boldsymbol{n}_1^{-1}\boldsymbol{n}\}$, but is not a vertex of this set.*

In particular, we may write,

$$\widetilde{\boldsymbol{n}} = \alpha_0 \boldsymbol{n}_1^{-1}\boldsymbol{n} + \sum_{j \in [n]} \alpha_j \widetilde{\boldsymbol{n}}^{(j)}. \tag{93}$$

where $\alpha_0 > 0$ and $\sum_{j=0}^{n} \alpha_i = 1$. By left-multiplying this on both sides by $\widetilde{\boldsymbol{M}}^{\times}_{n,i}$ and noting that $\boldsymbol{n}$ lies in the null-space of this matrix, we get,

$$\widetilde{\boldsymbol{M}}^{\times}_{n,i}\widetilde{n} = \sum_{j \in [n]} \alpha_j \widetilde{\boldsymbol{M}}^{\times}_{n,i}\widetilde{\boldsymbol{n}}^{(j)} \tag{94}$$

where note that $\sum_{j\in[n]} \alpha_j$ is strictly less than 1, since $\alpha_0 > 0$. We may write this vector as,

$$\begin{aligned}\widetilde{\boldsymbol{M}}^{\times}_{n,i}\tilde{n} &= \alpha_0 \mathbf{0} + \sum_{j\in[n]} \alpha_j \widetilde{\boldsymbol{M}}^{\times}_{n,i}\tilde{\boldsymbol{n}}^{(j)} \\ &= \frac{\alpha_0}{2^k} \sum_{\boldsymbol{h}\in\mathcal{H}_k} \widetilde{\boldsymbol{M}}^{\times}_{n,i}\boldsymbol{h} + \sum_{j\in[n]} \alpha_j \widetilde{\boldsymbol{M}}^{\times}_{n,i}\tilde{\boldsymbol{n}}^{(j)} \end{aligned} \tag{95}$$

Since $\alpha_0 > 0$, this equation implies that the image of $\tilde{\boldsymbol{n}}$ under $\widetilde{\boldsymbol{M}}^{\times}_{n,i}$ itself falls within $\mathrm{conv}(\widetilde{\boldsymbol{M}}^{\times}_{n,i}\mathcal{H}_k)$, but is itself not a vertex of this set. This means that $\tilde{\boldsymbol{n}}$ can never be a maximizer of $f_{n,i}\big(\cdot, \boldsymbol{y}\big)$ for any $\boldsymbol{y}$, and in particular when $\boldsymbol{y} = \tilde{\boldsymbol{n}}$, thereby proving Lemma 2. □

E.2 L-layer attention-only transformers with 1 head per layer: Proof of Corollary 1

Proof. The proof largely tracks the 2-layer case, with the main exception that we keep track of how the maximum possible rank of the matrix $\boldsymbol{M}^{\times}_{n,i}$ grows as a function of the depth of the transformer. In the case the 2-layer transformer, we show that it cannot exceed 2. With the addition of more layers, we show that it cannot exceed 2^{L-1}.

Recall from the notation in Equation (85) that the output of the first attention layer is,

$$\boldsymbol{x}^{(2)}_n = \boldsymbol{m}^{(1)}_n + \boldsymbol{M}^{(1)}_n \begin{bmatrix} x'_n & x'_{n-1} & \cdots & x'_1 \end{bmatrix}^T \tag{96}$$

where $\boldsymbol{M}^{(1)}_n \in \mathbb{R}^{d\times n}$ has rank at most 2. Let us rewrite this as,

$$\boldsymbol{x}^{(2)}_n = \boldsymbol{m}^{(1)}_n + \overline{\boldsymbol{M}}^{(1)}_n \begin{bmatrix} x'_T & x'_{T-1} & \cdots & x'_1 \end{bmatrix}^T \tag{97}$$

where $\boldsymbol{M}^{(1)}_n \in \mathbb{R}^{d\times T}$ is causally masked to be 0's when it operates on x_i for all indices $i > n$. Note that even with this causal masking, $\overline{\boldsymbol{M}}^{(1)}_n$ has rank at most 2, as discussed in Equation (85).

By induction, assume that the output of the $(\ell-1)^{\text{th}}$ attention layer is of the form,

$$\boldsymbol{x}^{(\ell)}_n = \boldsymbol{m}^{(\ell-1)}_n + \overline{\boldsymbol{M}}^{(\ell-1)}_n \boldsymbol{x}_{1:T} \tag{98}$$

where $\boldsymbol{x}_{1:T} \triangleq \begin{bmatrix} x'_T & x'_{T-1} & \cdots & x'_1 \end{bmatrix}^T$. Passing $\boldsymbol{x}^{(\ell)}_n$ through the ℓ^{th} attention layer, we get,

$$\boldsymbol{x}^{(\ell+1)}_n = \boldsymbol{x}^{(\ell)}_n + \sum_{i\leqslant n} \mathrm{att}^{(\ell)}_{n,i}\, \boldsymbol{W}^{(\ell)}_V \left(\boldsymbol{x}^{(\ell)}_i + \boldsymbol{p}^{(\ell),V}_{n-i}\right) \tag{99}$$

$$\begin{aligned} &= \boldsymbol{m}^{(\ell-1)}_n + \overline{\boldsymbol{M}}^{(\ell-1)}_n \boldsymbol{x}_{1:T} + \sum_{i\leqslant n} \mathrm{att}^{(\ell)}_{n,i}\, \boldsymbol{W}^{(\ell)}_V \boldsymbol{m}^{(\ell-1)}_i + \sum_{i\leqslant n} \mathrm{att}^{(\ell)}_{n,i}\, \boldsymbol{W}^{(\ell)}_V \overline{\boldsymbol{M}}^{(\ell-1)}_i \boldsymbol{x}_{1:T} \\ &\qquad + \sum_{i\leqslant n} \mathrm{att}^{(\ell)}_{n,i}\, \boldsymbol{W}^{(\ell)}_V \boldsymbol{p}^{(\ell),V}_{n-i} \end{aligned} \tag{100}$$

Define,

$$\boldsymbol{m}^{(\ell)}_n = \boldsymbol{m}^{(\ell-1)}_n + \sum_{i\leqslant n} \mathrm{att}^{(\ell)}_{n,i}\, \boldsymbol{W}^{(\ell)}_V \boldsymbol{m}^{(\ell-1)}_i + \sum_{i\leqslant n} \mathrm{att}^{(\ell)}_{n,i}\, \boldsymbol{W}^{(\ell)}_V \boldsymbol{p}^{(\ell),V}_{n-i}, \text{ and,} \tag{101}$$

$$\overline{\boldsymbol{M}}^{(\ell)}_n = \overline{\boldsymbol{M}}^{(\ell-1)}_n + \sum_{i\leqslant n} \mathrm{att}^{(\ell)}_{n,i}\, \boldsymbol{W}^{(\ell)}_V \overline{\boldsymbol{M}}^{(\ell-1)}_i \tag{102}$$

Then, we can write down,

$$\boldsymbol{x}^{(\ell+1)}_n = \boldsymbol{m}^{(\ell)}_n + \overline{\boldsymbol{M}}^{(\ell)}_n \boldsymbol{x}_{1:T} \tag{103}$$

We also inductively assume that for every $i \leqslant n$,

(i) $\overline{\boldsymbol{M}}^{(\ell-1)}_i$ has rank $R \leqslant 2^{\ell-1}$, and,

(ii) $\overline{\boldsymbol{M}}^{(\ell-1)}_i$ can be factorized in the form $\sum_{r=1}^R \boldsymbol{u}_r \cdot \boldsymbol{v}^T_{i,r}$, where only the $\boldsymbol{v}_{i,r}$'s depend on i, but the $\boldsymbol{u}_r$'s do not depend on i.

Both of these conditions are true when $\ell - 1 = 1$ as evidenced by the structure of $\boldsymbol{M}_i^{(1)}$ in Equation (86) and noting that $\overline{\boldsymbol{M}}_i^{(1)}$ is obtained from $\boldsymbol{M}_i^{(1)}$ by right multiplying by a diagonal mask matrix. Using the recursion in Equation (101), we prove that the induction hypotheses (i) and (ii) are true at layer ℓ as well. In particular using the decomposition in (ii), observe that,

$$\overline{\boldsymbol{M}}_n^{(\ell)} = \sum_{r=1}^{R} \boldsymbol{u}_r \cdot \boldsymbol{v}_{n,r}^T + \sum_{i \leqslant n} \mathrm{att}_{n,i}^{(\ell)} \boldsymbol{W}_V^{(\ell)} \sum_{r=1}^{R} \boldsymbol{u}_r \cdot \boldsymbol{v}_{i,r}^T \tag{104}$$

$$= \sum_{r=1}^{R} \boldsymbol{u}_r \cdot \boldsymbol{v}_{n,r}^T + \sum_{r=1}^{R} \boldsymbol{W}_V^{(\ell)} \boldsymbol{u}_r \cdot \left(\sum\nolimits_{i \leqslant n} \mathrm{att}_{n,i}^{(\ell)} \boldsymbol{v}_{i,r} \right)^T \tag{105}$$

$$= \sum_{r=1}^{2R} \boldsymbol{u}_r \cdot \boldsymbol{v}_{n,r}^T \tag{106}$$

where for $r' \in [R]$, $\boldsymbol{u}_{R+r'} \triangleq \boldsymbol{W}_V^{(\ell)} \boldsymbol{u}_r$ and $\boldsymbol{v}_{n,r'} \triangleq \sum_{i \leqslant n} \mathrm{att}_{n,i}^{(\ell)} \boldsymbol{v}_{i,r}$. Since $\boldsymbol{M}_n^{(\ell)}$ is the sum of $2R$ rank 1 matrices and therefore has rank at most $2R \leqslant 2^\ell$, proving both parts of the induction hypothesis.

By induction, at the end of the $(L-1)^{\text{th}}$ layer, we have an output which looks like,

$$\boldsymbol{x}_n^{(L)} = \boldsymbol{m}_n^{(L-1)} + \overline{\boldsymbol{M}}_n^{(L-1)} \boldsymbol{x}_{1:T} \tag{107}$$

where $\boldsymbol{M}_n^{(L-1)}$ has rank at most 2^{L-1}. More importantly, note that by the causal masking, even though it appears to depend on the whole input sequence through $\boldsymbol{x}_{1:T}$, note that $\boldsymbol{x}_n^{(L)}$ only depends on $x_1, \cdots, x_n$ and not on the future inputs to this time n. In particular, by a similar argument as in the 2-layer case (cf. Equation (85) to Equation (90)), for any $i \leqslant n-k$ we can decompose the dot-product of the key and query vectors at the L^{th} layer as a bilinear form which looks like,

$$\begin{aligned} &\left\langle \boldsymbol{W}_K^{(L)} \big(\boldsymbol{x}_i^{(L)} + \boldsymbol{p}_{n-i}^{(L),K} \big), \boldsymbol{W}_Q^{(L)} \boldsymbol{x}_n^{(L)} \right\rangle \\ &\qquad = \boldsymbol{x}^T \boldsymbol{M}_{n,i}^{\times} \boldsymbol{y} + \boldsymbol{y}^T \big(\overline{\boldsymbol{M}}_{n,i}^{\times} \big) \boldsymbol{y} + (\boldsymbol{m}_{n,i}^{\times})^T \boldsymbol{x} + (\overline{\boldsymbol{m}}_{n,i}^{\times})^T \boldsymbol{y} + m_{n,i}^{\times} \triangleq f_{n,i}^{(L+1)}(\boldsymbol{x}, \boldsymbol{y}) \end{aligned} \tag{108}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are defined as $\begin{bmatrix} x'_n & \cdots & x'_{n-k+1} \end{bmatrix}^T$ and $\begin{bmatrix} x'_{i-1} & \cdots & x'_{i-k} \end{bmatrix}^T$ respectively, and $\boldsymbol{M}_{n,i}^{\times}$ has rank at most that of $\boldsymbol{M}_n^{(L-1)}$, which is 2^{L-1}. In particular, if $2^{L-1} \leqslant k-2$, by Lemma 2 the proof concludes. □

E.3 The general case: Transformers with H_ℓ heads in layer ℓ: Proof of Theorem 6

The h^{th} head of the first layer of the attention-only transformer learns patterns of the form,

$$\widetilde{\boldsymbol{x}}_n^{(1,h)} = \sum_{i \leqslant n} \mathrm{att}_{n,i}^{(1,h)} \boldsymbol{W}_V^{(1,h)} \mathtt{Emb}(x_i) + \sum_{i \leqslant n} \boldsymbol{W}_V^{(1,h)} \boldsymbol{p}_{n-i}^{V,(1,h)} \tag{109}$$

$$= \sum_{i \leqslant n} \mathrm{att}_{n,i}^{(1,h)} \left(\frac{\phi^h(0) + \phi^h(1)}{2} + x'_i \cdot \frac{\phi^h(1) - \phi^h(0)}{2} \right) + \sum_{i \leqslant n} \boldsymbol{W}_V^{(1,h)} \boldsymbol{p}_{n-i}^{V,(1,h)} \tag{110}$$

where the last equation assumes a binary input sequence, defines $x'_i = 2x_i - 1$ and uses the notation $\phi^h(0) = \boldsymbol{W}_V^{(1,h)} \mathtt{Emb}(0)$ and $\phi^h(1) = \boldsymbol{W}_V^{(1,h)} \mathtt{Emb}(1)$. We can further rewrite this as,

$$\widetilde{\boldsymbol{x}}_n^{(1,h)} = \boldsymbol{m}_n^{(1,h)} + \boldsymbol{M}_n^{(1,h)} \boldsymbol{x}_{1:T} \tag{111}$$

where each $\boldsymbol{M}_n^{(1,h)} \in \mathbb{R}^{d \times T}$ is rank 1 and applies a causal mask on the inputs x_i for $i > n$. Recall that the output of the first attention layer applies a projection matrix on the concatentation of $\widetilde{\boldsymbol{x}}_n^{(1,h)}$ across $h \in [H_1]$ and then adds a residual connection. The output can be written down as,

$$\widetilde{\boldsymbol{x}}_n^{(2)} = \mathtt{Emb}(x_n) + \boldsymbol{W}_O^{(1)} \begin{bmatrix} \boldsymbol{m}_n^{(1,1)} \\ \vdots \\ \boldsymbol{m}_n^{(1,H_1)} \end{bmatrix} + \boldsymbol{W}_O^{(1)} \begin{bmatrix} \boldsymbol{M}_n^{(1,1)} \\ \vdots \\ \boldsymbol{M}_n^{(1,H_1)} \end{bmatrix} \boldsymbol{x}_{1:T} \tag{112}$$

$$= \boldsymbol{m}_n^{(1)} + \boldsymbol{M}_n^{(1)} \boldsymbol{x}_{1:T}, \tag{113}$$

where,

$$\boldsymbol{M}_n^{(1)} = \left(\frac{\texttt{Emb}(1) + \texttt{Emb}(0)}{2} \right) e_n^T + \boldsymbol{W}_O^{(1)} \begin{bmatrix} \boldsymbol{M}_n^{(1,1)} \\ \vdots \\ \boldsymbol{M}_n^{(1,H_1)} \end{bmatrix}, \text{ and,} \tag{114}$$

$$\boldsymbol{m}_n^{(1)} = \left(\frac{\texttt{Emb}(1) - \texttt{Emb}(0)}{2} \right) + \boldsymbol{W}_O^{(1)} \begin{bmatrix} \boldsymbol{m}_n^{(1,1)} \\ \vdots \\ \boldsymbol{m}_n^{(1,H_1)} \end{bmatrix} \tag{115}$$

Notice that the rank of the matrix $\boldsymbol{M}_n^{(1)}$ is at most $H_1 + 1$. This is because the concatenation operation can increase the rank at most additively, and since each of the $\boldsymbol{M}_n^{(1,h)}$ matrices are rank at most 1.

Following through the proof in Appendix E.2 for the L-layer case, we can prove inductively that at any layer ℓ, the output looks like,

$$\boldsymbol{x}_n^{(\ell)} = \boldsymbol{m}_n^{(\ell)} + \boldsymbol{M}_n^{(\ell)} \boldsymbol{x}_{1:T} \tag{116}$$

where the rank of $\boldsymbol{M}_n^{(\ell)}$ is $\prod_{i=1}^{\ell} (H_i + 1)$. Invoking Lemma 2, if $\prod_{i=1}^{L-1} (H_i + 1) \leqslant k - 2$, the attention-only transformer cannot realize a k^{th}-order induction head at layer L.

F Model architecture and hyper-parameters

The experiments were run on one $8 \times A100$ GPU node.

Parameter	Matrix shape
transformer.wte	$2 \times d$
transformer.wpe	$N \times d$
transformer.h.ln_1 $(\times \ell)$	$d \times 1$
transformer.h.attn.c_attn $(\times \ell)$	$3d \times d$
transformer.h.attn.c_proj $(\times \ell)$	$d \times d$
transformer.h.ln_2 $(\times \ell)$	$d \times 1$
transformer.h.mlp.c_fc $(\times \ell)$	$4d \times d$
transformer.h.mlp.c_proj $(\times \ell)$	$d \times 4d$
transformer.ln_f	$d \times 1$

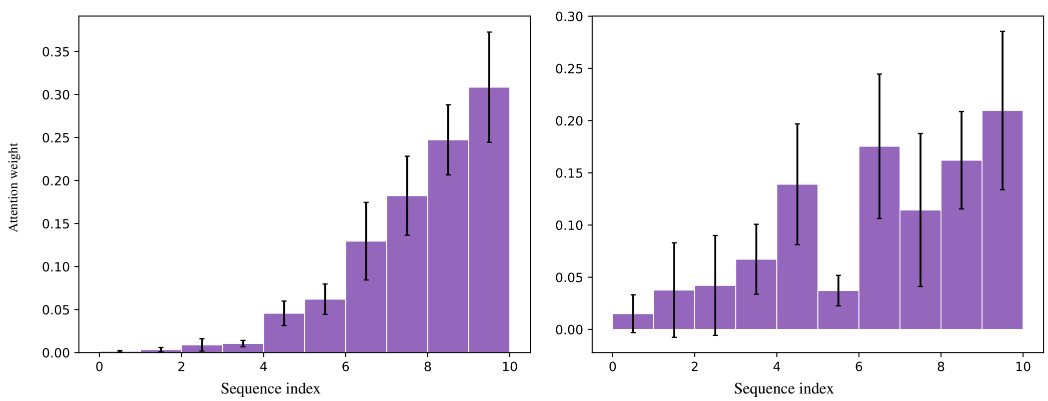
Table 2: Parameters in the transformer architecture with their shape.

G Additional experimental results

Assumption 1 suggests that the attention patterns $\text{att}_{n,i}^{(\ell)}$ in layers $\ell = 1, 2, \cdots, L-1$, as learnt by an L-layer attention-only transformers may only be a function of only the position indices n, i. In this section we run some additional experiments to test this conjecture. We train a 2 layer attention-only transformer with k heads in the first layer, on data drawn from a randomly sampled k^{th}-order Markov process, and focus on the learnt attention patterns as a function of in the input sequence. Figure 7 plots the results of this experiment for $k = 2$ and Figure 8 for $k = 3$. While in both cases there is some variance in the attention patterns learnt by the transformer in some of the heads, we believe that this is a consequence of the iteration budget of the transformer, and specifically the fact that even if the test loss appears to have converged, the transformer may still continue changing in the parameter space. Furthermore, when the attention patterns have some non-zero but small variance as a function of the input, a relaxation of Assumption 1, we also believe that the results we proved in Corollaries 1 and 2 and Theorem 6 should carry over approximately and leave this as an interesting question for future work. Conditional lower bounds of this nature, reliant on structural assumptions the transformer appears to demonstrate in practice are an interesting area of future research.

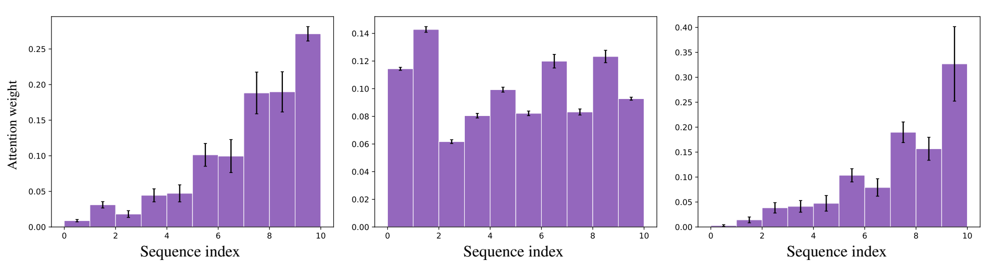
Dataset	k-th order binary Markov source
Architecture	Based on the GPT-2 architecture as implemented in [30]
Batch size	Grid-searched in $\{8, 16\}$
Accumulation steps	1
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.95$)
Learning rate	0.001
Scheduler	Cosine
# Iterations	Up to 25000
Weight decay	1×10^{-3}
Dropout	0
Sequence length	Grid-searched in $\{32, 64, 128, 256, 512, 1024\}$
Embedding dimension	Grid-searched in $\{16, 32, 64\}$
Transformer layers	Between 1 and 8
Attention heads	Up to k
Repetitions	3

Table 3: Settings and parameters for the transformer model used in the experiments.



(a) First head (b) Second head

Figure 7: Mean attention for column $n = 10$ of the two heads of the first attention layer, for a 2-layer 2-head transformer model trained on an order-3 Markov process, averaged across 100 input sequences of length 128.



(a) First head (b) Second head (c) Third head

Figure 8: Mean attention for column $n = 10$ of the three heads of the first attention layer, for a 2-layer 3-head transformer model trained on an order-3 Markov process, averaged across 100 input sequences of length 128.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper presents experimental evidence about k^{th}-order Markov processes being effectively learnable by small-depth transformers. We also prove theorems about the representation power of low-depth transformers, which are stated in the main paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We state that our lower bound in Corollary 1 is a conditional result in Section 6.1, and that it is an important open problem to see whether the conditional statements can be removed. While our work talks about the representational power of transformers, we also mention that learning dynamics of gradient descent is an important direction that future work needs to address.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theorems in the paper are stated and cross-referenced with proofs in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experiment hyperparameters are presented in Tables 2 and 3 and the code has been provided along with the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All the code has been open sourced; hyperparameter choices have been provided in Tables 2 and 3.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Tables 2 and 3 cover this information.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Standard error bars are provided on all plots.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Details in Appendix F.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: No NeurIPS code of ethics were violated.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This work is a primarily theoretical study on the behavior of tokenization on toy problems (learning Markov chains). The societal impact of this research is not likely to be significant.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No models with a high risk for misuse were trained or released.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Code has been properly credited, via citing the relevant papers.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new assets released.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.