Fixed Universal Transformers

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce *universal transformers*: fixed transformers that can simulate any transformer in a given class via a suitable input embedding. Analogous to a universal Turing machine, the input embedding encodes a description of the target model while all internal parameters remain fixed. We provide explicit sparse constructions achieving universality when the embedding dimension is sufficiently large, and further show that universality is generic: randomly initialized transformers are universal almost surely, which aligns with recent empirical results of Zhong and Andreas (2024). We empirically validate our theory on the algorithmic tasks of parenthesis balancing and multi-hop reasoning. Our results suggest that much of a transformer's expressive power may reside in its input representation rather than its learned weights.

1 Introduction

We introduce the notion of a *universal transformer*: a fixed transformer that can simulate any transformer in a prescribed class by changing only the input embedding. In a universal transformer, the embedding plays the role of a program: it encodes the parameters of the target transformer, while the attention and value parameters of the universal transformer remain fixed. This notion can be regarded as a deep learning analogue of the concept of universality from the theory of computation, of which universal Turing machines [22] and universal circuits [23] are the prime examples. A universal Turing machine is a single Turing machine that is capable of simulating any other Turing machine (provided in a suitable encoding) as input, much like how we regard general purpose computers today [22]. A universal circuit is a single circuit that can simulate any target circuit of comparable size by simply fixing additional special "control" inputs to an appropriate values specific to the target circuit [23]. Note that universality here is not the the same as universal approximation, which refers to the ability to choose the model parameters to approximate an arbitrary function. Instead, the architecture and non-embedding parameters are fixed, and the question is, for any target transformer of comparable form, whether there is a suitable representation of the input that makes this fixed computation emulate the target transformer.

Our notion of universality is partly motivated by recent experimental findings of [28] on *random transformers*, where they found that many basic capabilities of language models are already present in randomly initialized transformers, with only trained embedding and unembedding layers. Our results provide a theoretical lens on this phenomenon: random transformers are universal almost surely. Moreover, randomness is not essential, since universal transformers can also be *explicitly* constructed. These constructions give insight into how such emulation can be carried out, and they do not resemble, and are much sparser than, commonly used initialization schemes for transformers. More broadly, our results highlight the role of embedding layers: features should not be identified solely with learned internal weights, since a fixed set of internal parameters can support many distinct computations depending on the representation supplied at the input. Universal transformers are also related to model reprogramming, a transfer-learning paradigm that adapts a pre-trained model to new

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

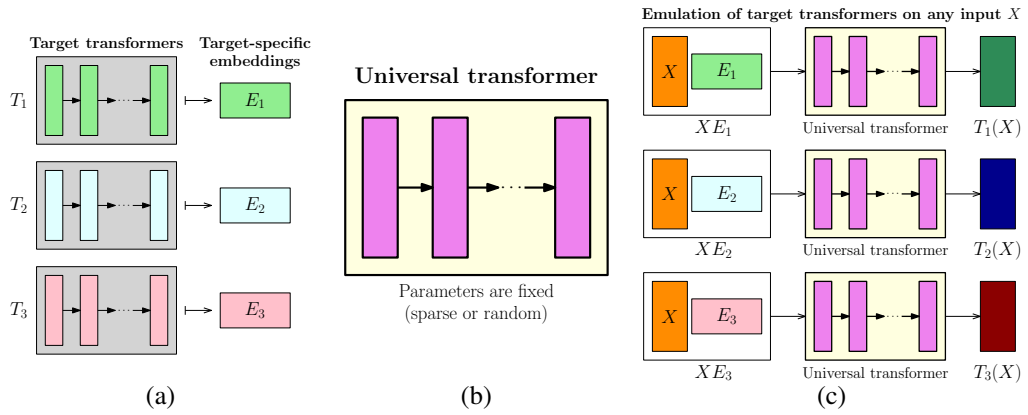


Figure 1: Schematic overview of universal transformers. (a) Each target transformer corresponds to a target-specific embedding (based on the target's model parameters). (b) The parameters of the universal transformer are fixed, either using the sparse deterministic construction from Theorem 3.1, or chosen randomly as in Theorem 3.3. (c) The composition of the universal transformer with the target-specific embedding emulates the target transformer on all inputs.

domains by changing only the embedding layers [3]; the existence of universal transformers vastly expands the potential scope of that approach.

Our contributions. We formalize the notion of a universal transformer, which can simulate any transformer within a given class via a suitable input embedding. We provide explicit constructions of such models and further show that randomly initialized transformers are universal.

- **Explicit sparse universal transformers.** We show that there exists a fixed H-head, L-layer universal transformer with embedding dimension m and $\{0,1\}$-valued parameters that can simulate any H-head, L-layer transformer with embedding dimension d, provided that $m = O(Ld)$ when $H = 1$ and $m = O(H^L d)$ when $H \geq 2$. Moreover, this result extends to the looped (weight-tied) setting, where the same parameters are reused across layers in the universal transformer.

  Theorem 1.1 (Informal version of Theorems 3.1 and 3.2)**.** *There exists fixed $\{0,1\}$-valued parameters defining a H-head L-layer transformer (or looped transformer) UT with embedding dimension $m = O(Ld)$ when $H = 1$ and $m = O(H^L d)$ when $H \geq 2$, such that there are at most m nonzero entries in each parameter matrix, and for any H-head L-layer transformer with embedding dimension d, there exists an embedding matrix $E \in \mathbb{R}^{d \times m}$, we have $\mathrm{UT}(XE) = T(X)$ for any input X.*

- **Random universal transformers.** We prove that universality is in abundance: almost surely, a randomly initialized H-head, L-layer transformer can simulate any H-head, L-layer transformer with embedding dimension d, when the embedding dimension satisfies $m = O(H^L d)$.

  Theorem 1.2 (Informal version of Theorems 3.3 and 3.4)**.** *Draw random parameters from an absolutely continuous distribution. With probability 1, the resulting H-head L-layer transformer (or looped transformer) UT with embedding dimension $m = O(H^L d)$ such that for any H-head L-layer transformer with embedding dimension d, there exists an embedding matrix $E \in \mathbb{R}^{d \times m}$, we have $\mathrm{UT}(XE) = T(X)$ for any input X.*

- **Lower bounds.** We establish that, when the number of heads is $H \geq 2$, the exponential dependence on depth L is unavoidable, even when the embedding dimension is 1, for any approach based on "attention-matching", where the universal transformer reproduces the attention matrices of the transformer being simulated.

  Theorem 1.3 (Informal version of Theorem 4.1)**.** *If an H-head, L-layer transformer UT with embedding dimension m can simulate any H-head, L-layer transformer T with embedding dimension 1 via "attention-matching", then $m \geq H^{L/2}$.*

- **Empirical evaluation.** We evaluate sparse and random universal transformers on parenthesis balancing and k-hop induction head tasks, achieving perfect or near-perfect accuracy. These

findings complement our representational results and demonstrate that standard gradient-based optimization can effectively learn the required embeddings.

1.1 Related work

Universality in computation. Our notion of a universal transformer takes inspiration from classical notions of universality in the theory of computation [22, 23], but adapts it to the deep learning context. Specifically, the description of the target transformer is provided as an input embedding layer for the universal transformer.

As mentioned in the introduction, universal transformers can also be regarded as a form of model reprogramming [2]. Theoretical analyses on model reprogramming rely on distributional assumptions and offer average case error guarantees [25], much like other works on transfer learning. Our results offer a complementary type of theoretical guarantee for model reprogramming when the "pre-trained model" is a universal transformer, and the target tasks are solvable using task-specific transformers.

Universal approximation and Turing-completeness. The notion of universality we study in this work is distinct from the universal approximation property of various types of neural networks (including transformers) [e.g., 6, 13, 11, 27, 24], which is concerned with the ability to arbitrarily-well approximate broad classes of functions by entire families of neural networks. In particular, for any given target function, the approximating network's size and internal parameters may all be specific to the target function. Similarly, many network families have been shown to satisfy Turing-completeness, which (in this context) means that any Turing machine can be simulated by an appropriate member of the network family [21, 7, 5, 24, 12]. (We have an unfortunate clash of terminology with [7], who use the term "Universal Transformer" for a recurrent extension of the standard transformer architecture that the authors argue is Turing-complete; our work does not change the transformer architecture.) Again, the particular network that simulates a given Turing machine may have size and internal parameters that depend on that Turing machine.

The Turing-completeness of transformers offers one avenue to construct universal transformers: use the fact that every transformer can be simulated by a Turing machine (e.g., one that implements a Python interpreter with PyTorch) when provided in a suitable encoding (e.g., model `.pt` file), and then obtain a transformer that simulates *that* Turing machine. However, such an approach would not (at least readily) come with any efficiency or succinctness guarantees about the constructed universal transformer, even for the classes of transformers we consider in our results.

Random networks. Random networks have long served as a lens for studying the capabilities of a wide range of machine learning models (especially at random initialization), including kernel machines [17, 19], feedforward neural networks [14, 10], recurrent neural networks [15, 16, 1], and—more recently—transformers [3, 8]. As mentioned in the introduction, our work is, in part, inspired by the empirical study of [28] on random transformers, which tested and validated the hypothesis that many capabilities of language models are already present in randomly-initialized transformers, prior to any training. That work also empirically observed that the token representations at different levels of a random transformers seem to lie close to low-dimensional subspaces corresponding to the current task at hand, but also that these representations could not be sparsified by selecting individual neurons in the multi-layer Perceptron (MLP) layers; this is taken as evidence against a "lottery ticket hypothesis" [10] for the success of random transformers.

The work of [8] continues the investigation of random transformers by further empirically investigating the relative importance of the self-attention parameters and MLP parameters. Theoretically, they attribute the expressivity of "Frozen-QK" transformers—where only the self-attention parameters are fixed at their initialization, but the MLP parameters are trained—to the universal approximation capabilities of MLPs and an embedding dimension that is sufficiently large for self-attention to gather the entire input sequence into a single token embedding. This explanation for the capabilities of (a different type of) random transformers is complementary to that provided by universality. Our theoretical work focuses on attention-only transformers (as opposed to transformers with MLP layers), considers exact representation (as opposed to approximation), and does not use any target-specific layers besides the embedding layers.

2 Preliminaries

Notation. For a positive integer k, we denote $[k] := \{1, 2, \ldots, k\}$. We use $\{0,1\}^{m\times n}$ to denote the class of $m \times n$ matrices with entries in $\{0, 1\}$. For a matrix $X \in \mathbb{R}^{n\times d}$, we interpret the n rows as tokens, where n is the context length, and d as the feature dimension. We use $\mathrm{softmax}$ to denote the row-wise softmax function.

2.1 Standard transformer

We consider a standard multi-head self-attention transformer without MLP layers.

Transformer class. We use $\mathrm{TF}_{H,L,d_{\mathrm{in}},d}$ to denote the class of H-head L-layer transformers, with the input and output dimension d_{in}, and each head uses an internal head dimension d.

A transformer $T \in \mathrm{TF}_{H,L,d_{\mathrm{in}},d}$ is defined by parameters including query and key matrices $W_Q^{l,h}, W_K^{l,h} \in \mathbb{R}^{d_{\mathrm{in}}\times d}$, value matrix $W_V^{l,h} \in \mathbb{R}^{d_{\mathrm{in}}\times d}$, and output matrix $W_O^{l,h} \in \mathbb{R}^{d\times d_{\mathrm{in}}}$, for each layer $l \in [L]$ and head $h \in [H]$, and the following computation:

Given an input sequence $X^0 \in \mathbb{R}^{n\times d_{\mathrm{in}}}$, T computes for each layer $l \in [L]$:

$$A^{l,h} := \mathrm{softmax}\Big((X^{l-1}W_Q^{l,h})(X^{l-1}W_K^{l,h})^\top\Big) \in \mathbb{R}^{n\times n},$$

$$X^l := \sum_{h=1}^{H} A^{l,h} X^{l-1} W_V^{l,h} W_O^{l,h} \in \mathbb{R}^{n\times d_{\mathrm{in}}}.$$

The final output is $T(X) := X^L \in \mathbb{R}^{n\times d_{\mathrm{in}}}$.

2.2 Universal transformer

We now define the *universal transformer* model. Like universal circuits, we define a universal transformer with respect to a class of transformers of certain set of hyper-parameters.

Universal transformer model. We define a parametric model $\mathrm{UTF}^m_{H,L,d_{\mathrm{in}},d}$ which is an H-head, L-layer transformer with embedding dimension m. It is parameterized by matrices $R_Q^{l,h}, R_K^{l,h} \in \mathbb{R}^{m\times d}$ and $R_V^{l,h} \in \mathbb{R}^{m\times m}$ for each $(l, h) \in [L] \times [H]$, together with an unembedding matrix $U \in \mathbb{R}^{m\times d_{\mathrm{in}}}$.

Given an input sequence $X^0 \in \mathbb{R}^{n\times d_{\mathrm{in}}}$ and an embedding matrix $E \in \mathbb{R}^{d_{\mathrm{in}}\times m}$, $\mathrm{UTF}^m_{H,L,d_{\mathrm{in}},d}$ first maps the input into the m-dimensional space:

$$\tilde{X}^0 := X^0 E \in \mathbb{R}^{n\times m}.$$

For each layer $l \in [L]$, it then computes

$$\tilde{A}^{l,h} := \mathrm{softmax}\Big((\tilde{X}^{l-1}R_Q^{l,h})(\tilde{X}^{l-1}R_K^{l,h})^\top\Big) \in \mathbb{R}^{n\times n},$$

$$\tilde{X}^l := \sum_{h=1}^{H} \tilde{A}^{l,h} \tilde{X}^{l-1} R_V^{l,h} \in \mathbb{R}^{n\times m}.$$

Finally, the output is projected back to the input space:

$$\mathrm{UTF}^m_{H,L,d_{\mathrm{in}},d}(X, E) := \tilde{X}^L U \in \mathbb{R}^{n\times d_{\mathrm{in}}}.$$

The key property of the universal transformer is that it can simulate any transformer in $\mathrm{TF}_{H,L,d_{\mathrm{in}},d}$ via a suitable choice of embedding.

Definition 2.1 (Universal transformer)**.** *We call a* $\mathrm{UTF}^m_{H,L,d_{\mathrm{in}},d}$*, parametrized by fixed* $\{R_Q^{l,h}, R_K^{l,h}, R_V^{l,h}, U\}$*, a* universal transformer *for the class* $\mathrm{TF}_{H,L,d_{\mathrm{in}},d}$ *if, for any* $T \in \mathrm{TF}_{H,L,d_{\mathrm{in}},d}$*, there exists an embedding matrix* E *such that*

$$\mathrm{UTF}^m_{H,L,d_{\mathrm{in}},d}(X, E) = T(X) \quad \textit{for all } X \in \mathbb{R}^{n\times d_{\mathrm{in}}} \textit{ and all } n \in \mathbb{N}.$$

Note that the universal transformer must be *uniform over the context length* n: the parameters $\{R_Q^{l,h}, R_K^{l,h}, R_V^{l,h}, U\}$ are fixed and do not depend on n, and the same universal transformer can be applied to inputs of arbitrary length.

This concept is analogous to concepts of universal Turing machines [22] and universal circuits of [23], adapted to the deep learning context. In our setting, the embedding matrix E serves as a *description* (or encoding) of the target transformer being simulated, while the parameters $\{R_Q^{l,h}, R_K^{l,h}, R_V^{l,h}, U\}$ define a fixed computational template that performs the simulation. The embedding dimension m therefore quantifies the expressive capacity of the universal transformer, i.e., the amount of information needed to encode a target transformer within this fixed architecture. In Sections 3 and 4, we study the minimal m required for universality over the class $\mathrm{TF}_{H,L,d_{\mathrm{in}},d}$.

We also consider a *looped* (or weight-tied) variant of the universal transformer, in which the same attention parameters are reused across iterations.

Looped universal transformer model. We define a looped parametric model looped-$\mathrm{UTF}^m_{H,L,d_{\mathrm{in}},d}$ which is an H-head transformer unrolled for L iterations, with embedding dimension m. It is parameterized by shared matrices $R_Q^h, R_K^h \in \mathbb{R}^{m\times d}$ and $R_V^h \in \mathbb{R}^{m\times m}$ for each $h \in [H]$, together with an unembedding matrix $U \in \mathbb{R}^{m\times d_{\mathrm{in}}}$. The looped-$\mathrm{UTF}^m_{H,L,d_{\mathrm{in}},d}$ follows the same computation path as $\mathrm{UTF}^m_{H,L,d_{\mathrm{in}},d}$ but replacing $R_Q^{l,h}, R_K^{l,h}, R_V^{l,h}$ with R_Q^h, R_K^h, R_V^h.

3 Constructing universal transformers

In this section we characterize when a transformer can serve as a *universal transformer* for the class $\mathrm{TF}_{H,L,d_{\mathrm{in}},d}$, and to determine the minimal embedding dimension m required for universality.

3.1 Fixed sparse universal transformer

In the following theorem, we give an explicit construction of a *sparse* universal transformer, which establishes an upper bound on the required embedding dimension and serves as the basis for our experiments. The upper bound is given in terms of the following quantity $\overline{m}$:

$$\overline{m} := \begin{cases} (L+1)\max\{2d, d_{\mathrm{in}}\}, & \text{if } H = 1, \\ \frac{2H(H^L-1)}{H-1} d + H^L d_{\mathrm{in}}, & \text{if } H > 1. \end{cases}$$

Theorem 3.1 (Sparse universal transformer)**.** *There exist fixed parameters* $\{R_Q^{l,h}, R_K^{l,h}, R_V^{l,h}\}_{l\in[L],\, h\in[H]}$ *and an unembedding matrix* U*, with at most* m *nonzero entries in each parameter matrix, defining a* $\mathrm{UTF}^m_{H,L,d_{\mathrm{in}},d}$ *such that it is a universal transformer for the class* $\mathrm{TF}_{H,L,d_{\mathrm{in}},d}$*, and*

$$R_Q^{l,h}, R_K^{l,h} \in \{0,1\}^{m\times d}, \quad R_V^{l,h} \in \{0,1\}^{m\times m}, \quad U \in \{0,1\}^{m\times d_{\mathrm{in}}},$$

if the embedding dimension satisfies $m \geq \overline{m}$*.*

Proof sketch. The proof proceeds by showing that the universal transformer can simultaneously match (i) the attention patterns and (ii) the value transformations of any transformer. The outputs of any target transformer T in the class $\mathrm{TF}_{H,L,d_{\mathrm{in}},d}$ and universal transformer can be written as

$$T(X^0) = \sum_{(h_1,\ldots,h_L)\in[H]^L} A^{L,h_L}\cdots A^{1,h_1} X^0 W_V^{1,h_1} W_O^{1,h_1} \cdots W_V^{L,h_L} W_O^{L,h_L}$$

$$\mathrm{UTF}^m_{H,L,d_{\mathrm{in}},d}(X^0, E) = \sum_{(h_1,\ldots,h_L)\in[H]^L} \tilde{A}^{L,h_L}\cdots \tilde{A}^{1,h_1} X^0 E R_V^{1,h_1} \cdots R_V^{L,h_L} U$$

First, we ensure that for every layer $l \in [L]$ and head $h \in [H]$, the attention matrices match: $\tilde{A}^{l,h} = A^{l,h}$. This guarantees that the universal transformer follows the same attention pattern over tokens as the target transformer. Second, conditioned on matching the attention matrices, we construct the value transformations so that $ER_V^{1,h_1}\cdots R_V^{L,h_L}U = W_V^{1,h_1}W_O^{1,h_1}\cdots W_V^{L,h_L}W_O^{L,h_L}$. Therefore, the overall input-output map is identical. This reduces the problem to implementing the correct linear transformations along each attention path.

Construction. We explicitly construct the parameters to realize this behavior. The key idea is to encode the weights of the target transformer into the embedding matrix E, and each parameter matrix is either routing or reading out from certain block of E. The sparsity follows from the fact that each matrix only interacts with a small number of coordinates.

We interpret the embedding space as an H-ary tree, indexed by $(h_1, \ldots, h_l) \in [H]^l$ for $l \leq L$, each corresponding to a head sequence. These blocks represent all possible computation paths through the transformer. For each layer l and head h, the matrix $R_V^{l,h}$ selects the child block corresponding to $(h_1, \ldots, h_{l-1}, h)$ from each parent block corresponding to a prefix $(h_1, \ldots, h_{l-1})$, and is zero elsewhere. Thus, $R_V^{l,h}$ deterministically routes representations along a tree of depth L. $R_Q^{l,h}$ and $R_K^{l,h}$ are sub-block selectors: they extract the hth sub-block for all the internal blocks indexed by the prefix $(h_1, \ldots, h_{l-1}) \in [H]^{l-1}$, and all other coordinates are mapped to zero. Finally, the unembedding matrix U reads out the appropriate coordinates from the leaf blocks indexed by $(h_1, \ldots, h_L) \in [H]^L$.

We further provide a sparse construction for looped (weight-tied) transformers, where the same parameters are reused across layers. This leads to a more succinct universal architectural description where the number of parameter matrices is $O(H)$ instead of $O(HL)$ in the standard setting. Let $\overline{m}_{\text{looped}}$ denote the embedding dimension that we show is sufficient for the looped transformer case:

$$\overline{m}_{\text{looped}} := \begin{cases} (L+1)\max\{2d, d_{\text{in}}\}, & \text{if } H = 1, \\ \frac{H^{L+1}-1}{H-1}\max\{2d, d_{\text{in}}\}, & \text{if } H > 1. \end{cases}$$

Theorem 3.2 (Sparse looped universal transformer)**.** *There exist fixed parameters $\{R_Q^h, R_K^h, R_V^h\}_{h\in[H]}$ and an unembedding matrix U, with at most m nonzero entries in each parameter matrix, defining a* looped-UTF$^m_{H,L,d_{\text{in}},d}$ *such that it is a looped universal transformer for the class* TF$_{H,L,d_{\text{in}},d}$*, and*

$$R_Q^h, R_K^h \in \{0,1\}^{m\times d}, \quad R_V^h \in \{0,1\}^{m\times m}, \quad U \in \{0,1\}^{m\times d_{\text{in}}},$$

when the embedding dimension satisfies $m \geq \overline{m}_{\text{looped}}$.

Note that all the above constructions work even if there is attention masking, e.g., causal masking for decoder type of transformer, since we are matching all the attention matrices.

3.2 Random universal transformer

The sparse constructions above demonstrate that universality is achievable and establish a sufficient embedding dimension. However, they rely on carefully designed, highly structured parameters. This raises a natural question: *How special are universal transformers?* We now show that universality is in fact a generic property. In particular, a randomly initialized transformer is universal almost surely.

Theorem 3.3 (Random universal transformer)**.** *Let $\{R_Q^{l,h}, R_K^{l,h}, R_V^{l,h}\}_{l\in[L],\, h\in[H]}$ and U be drawn independently from an absolutely continuous distribution over $\mathbb{R}^{m\times d}$, $\mathbb{R}^{m\times d}$, $\mathbb{R}^{m\times m}$, and $\mathbb{R}^{m\times d_{\text{in}}}$, respectively (e.g., with i.i.d. Gaussian entries). Then, with probability 1, the resulting transformer* UTF$^m_{H,L,d_{\text{in}},d}$ *is a universal transformer for the class* TF$_{H,L,d_{\text{in}},d}$*, provided that $m \geq \overline{m}$.*

The universality condition can be reduced to a collection of algebraic non-degeneracy conditions on the parameters $\{R_Q, R_K, R_V, U\}$. These conditions fail only when certain matrices are rank-deficient or satisfy polynomial constraints. The theorem follows from the fact that such degeneracies form a measure-zero set. This also applies to the looped setting.

Theorem 3.4 (Random looped universal transformer)**.** *Let $\{R_Q^h, R_K^h, R_V^h\}_{h\in[H]}$ and U be drawn independently from an absolutely continuous distribution over $\mathbb{R}^{m\times d}$, $\mathbb{R}^{m\times d}$, $\mathbb{R}^{m\times m}$, and $\mathbb{R}^{m\times d_{\text{in}}}$, respectively (e.g., with i.i.d. Gaussian entries). Then, with probability 1, the resulting looped transformer* looped-UTF$^m_{H,L,d_{\text{in}},d}$ *is a looped universal transformer for the class* TF$_{H,L,d_{\text{in}},d}$*, when the embedding dimension satisfies $m \geq \overline{m}_{\text{looped}}$.*

These results show that the embedding dimension $m = \Theta(H^L \max\{d, d_{\text{in}}\})$ is sufficient for universality, and that almost all transformers of this size are universal.

4 Optimality

In this section, we establish lower bounds on the embedding dimension m required for universality. Our upper bounds in Section 3 are achieved by explicitly matching the attention matrices of the target transformer, i.e., enforcing $\tilde{A}^{l,h} = A^{l,h}$ for all (l, h). We prove a lower bound on m for such "attention-matching" universal transformers. To do this, we prove a lower bound on m for a strictly stronger class of models that is additionally *given* the desired attention matrices as input. This, in turn, implies an $H^{\Omega(L)}$ lower bound on the embedding dimension for attention-matching approaches.

Shared-attention universal computation. We formalize this stronger model as follows. The shared-attention universal transformer, denoted shared-attn-UTF$^m_{H,L,d_{\text{in}},d}$, is parameterized only by value matrices $\{R_V^{l,h}\}_{l\in[L],h\in[H]}$. It takes as input a sequence $X^0 \in \mathbb{R}^{n\times d_{\text{in}}}$, an embedding matrix $E \in \mathbb{R}^{d_{\text{in}}\times m}$, an unembedding matrix $U \in \mathbb{R}^{m\times d_{\text{in}}}$, and attention matrices $\{A^{l,h}\}_{l\in[L],h\in[H]}$. The computation proceeds identically to UTF$^m_{H,L,d_{\text{in}},d}$, except that the attention matrices are externally provided and fixed, i.e., $\tilde{A}^{l,h}$ is replaced by $A^{l,h}$ for all (l, h).

Theorem 4.1 (Lower bound for universal transformers)**.** *Let $d_{\text{in}} = 1$. Consider any transformer class* $\text{TF}_{H,L,1,d}$*. If there exists a* shared-attn-UTF$^m_{H,L,d_{\text{in}},d}$ *(or* looped-shared-attn-UTF$^m_{H,L,d_{\text{in}},d}$*) such that for any $T \in \text{TF}_{H,L,1,d}$ (or* looped-TF$_{H,L,1,d}$*) with attention matrices $\{A^{l,h}\}_{l\in[L],h\in[H]}$,*

$$\text{shared-attn-UTF}^m_{H,L,d_{\text{in}},d}(X, E, U, \{A^{l,h}\}_{l\in[L],h\in[H]}) = T(X)$$

for all $X \in \mathbb{R}^{n\times d_{\text{in}}}, n \in \mathbb{N}$, and $n \times n$ row-stochastic matrices $A^{l,h}$, then $m \geq H^{L/2}$.

Theorem 4.1 shows that the embedding dimension must grow exponentially in the depth L, even in the simplest setting $d_{\text{in}} = 1$. Notably, this lower bound holds in the shared-attention model, where the desired attention matrices are given as part of the input. Thus, the complexity arises from representing the exponentially-many computation paths induced by the multi-head attention structure.

In particular, across L layers and H heads, there are H^L possible sequences of attention heads, each corresponding to a distinct computation path. The universal transformer must be able to simultaneously represent and combine these paths. The lower bound shows that compressing this exponential-size family of behaviors into a low-dimensional embedding is impossible.

5 Experimental studies

In this section, we experimentally evaluate the universal transformers constructed in Section 3 on two canonical reasoning tasks—parenthesis balancing and k-hop induction heads—which are widely used as toy benchmarks for studying algorithmic reasoning in language models [26, 20]. These experiments complement our theoretical results on representational power by demonstrating empirically that standard gradient-based optimization can successfully learn the required embeddings for universal transformers to solve these tasks.

5.1 Tasks

Parenthesis Balancing (Dyck-1). In this task, the model is given a sequence of left- and right-parentheses and must output whether the parentheses in the sequence are properly balanced. Such sequences form the Dyck-1 language, a canonical example of a context-free language [4]. In our setup, the vocabulary size is 4: two parenthesis tokens and two labels. The maximum length for the input parentheses sequences is 60.

k-hop induction heads. This task generalizes the standard induction head mechanism studied in language models [9, 18]. In the standard induction heads, the model outputs the token that follows the most recent previous occurrence of the current token in the context. The k-hop variant extends this mechanism by recursively composing k such retrieval steps. For example, for an input $X = aabccabca$, the 2-hop induction head output for the last token a is c. Under a widely-believed complexity assumption, $\Theta(\log k)$ layers are necessary and sufficient to solve this task [20]. Therefore, we use this task to evaluate the performance of deeper universal transformers. For this task, the vocabulary size is 30, including the number of hops tokens, 4 character tokens and label tokens. The maximum sequence length is 100.

Task	Sparse	Random	Fully-trained
Parenthesis Balancing	100.0%	94.4%	100.0%
1-hop Induction Head	30.2%	29.4%	8.4%
2-hop Induction Head	33.2%	30.5%	51.8%
3-hop Induction Head	39.2%	32.6%	99.5%
4-hop Induction Head	39.4%	35.2%	95.6%

(a) Attention-only.

Task	Sparse	Random	Fully-trained
Parenthesis Balancing	100.0%	100.0%	93.0%
1-hop Induction Head	100.0%	96.1%	100.0%
2-hop Induction Head	100.0%	85.7%	100.0%
3-hop Induction Head	100.0%	92.6%	100.0%
4-hop Induction Head	100.0%	89.2%	100.0%

(b) Adding residual and layer normalization.

Table 1: Accuracy on parenthesis balancing and k-hop induction-head tasks. Parenthesis balancing uses 4-head, 2-layer transformers with head dimension 24 and embedding dimension 1024. The 1-, 2-, 3-, and 4-hop induction-head tasks use 2-head transformers with depths $2, 3, 4, 4$, head dimensions $28, 28, 30, 30$, and embedding dimensions $456, 1024, 2280, 2280$, respectively.

Model	Sparse	Random	Fully-trained
Attention-only	100%	94.4%	100%
+ residual	100%	95.4%	100%
+ residual and LN	100%	100%	93.0%
+ MLP	100%	100%	100%
+ MLP and LN	100%	100%	94.4%

(a) Parenthesis Balancing.

Model	Sparse	Random	Fully-trained
Attention-only	33.2%	30.5%	51.8%
+ residual	53.8%	54.3%	58.4%
+ residual and LN	100%	85.7%	100%
+ MLP	54.6%	54.1%	100%
+ MLP and LN	100%	66.2%	100%

(b) 2-hop Induction Heads.

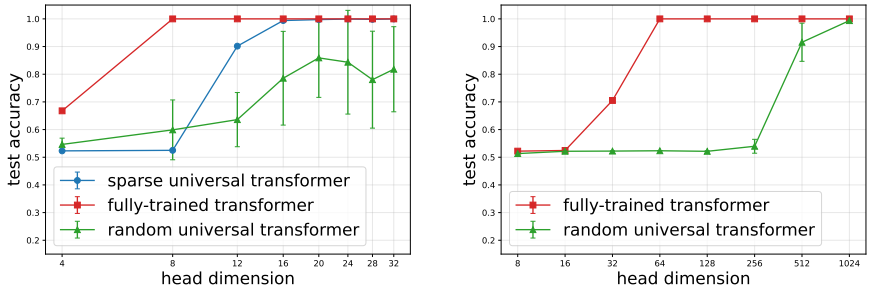
Table 2: Ablation study on parenthesis balancing and 2-hop induction-head tasks. Accuracies are reported for 4-head, 2-layer transformers with head dimension 24 and embedding dimension 1024 on parenthesis balancing, and for 2-head, 3-layer transformers with head dimension 28 and embedding dimension 1024 on 2-hop induction heads.

Architectures. We implement the universal transformer $\mathrm{UTF}^m_{H,L,d_{\mathrm{in}},d}$ using one-hot token representations, so that d_{in} corresponds to the vocabulary size. Following the construction in Theorem 3.1, for head dimension d, the embedding dimension is given by $m = \overline{m}$. We compare three models: (i) the constructed sparse universal transformer from Theorem 3.1, (ii) a random universal transformer with parameters initialized i.i.d. from $\mathrm{Uniform}(-1/\sqrt{m},\, 1/\sqrt{m})$ (the default PyTorch initialization), and (iii) a fully trained transformer. Following [28], for both the sparse and random universal transformers, we train only the embedding matrix E and the unembedding matrix U, while keeping all other parameters fixed. For random universal transformers, all the results are averaged across 5 seeds.

5.2 Results

Table 1 summarizes the main results. We first consider the attention-only setting, which directly matches our theoretical model $\mathrm{UTF}^m_{H,L,d_{\mathrm{in}},d}$. On parenthesis balancing, the sparse universal transformer reaches 100.0% accuracy, matching the fully trained transformer and the random universal also reaches non-trivial accuracy 94.4%. This shows that the explicit construction from Theorem 3.1 is not merely a worst-case representation result: the embedding and unembedding matrices can be learned by gradient descent to solve a nontrivial algorithmic task.

The induction-head tasks are more challenging in the attention-only setting. Both sparse and random universal transformers obtain nontrivial accuracies (random guessing gives 20%) but far from optimal. This suggests that while the attention-only universal transformers have the representational capacity, optimization becomes increasingly difficult as the task requires more depth. Adding standard components that stabalize training can substantially improve performance. With residual connections and layer normalization, both sparse and random universal transformers reach much higher accuracy. In particular, the sparse universal transformer reaches 100.0% accuracy on all reported induction-head tasks, while the random universal transformer achieves between 85.7% and 96.1% accuracy.



(a) Constructed parameterization $m = \overline{m}$ (b) Standard parametrization $m = Hd$

Figure 2: **Embedding dimension scaling for 2-hop induction heads.** Results are shown for 2-head, 3-layer transformers. *Left:* Performance under constructed parameterization from Theorem 3.1, where $m = \overline{m}$. *Right:* Performance under standard parameterization, where $m = Hd$. Error bars indicate ± 1 standard deviation across 5 random seeds.

5.2.1 Ablation studies

We further study how different components of standard transformers can affect the training optimization performance. Table 2 studies the effect of adding residual connections, layer normalization, and MLP layers. The MLP layers are randomly initialized using the PyTorch default initialization scheme. For the sparse and random universal transformers, the MLP layers are also kept fixed and untrained.

On parenthesis balancing, the sparse universal transformer is robust across all variants and achieves 100.0% accuracy in every setting. The random universal transformer also improves from 94.4% in the attention-only setting to 100.0% once residual connections and layer normalization or MLP layers are included. The 2-hop induction-head task reveals a sharper distinction between architectural components. Adding layer normalization together with residual connections can largely improve the performance, and this is also robust when adding frozen random MLP layers.

Overall, the sparse construction gives a useful inductive bias for optimization: it consistently matches or exceeds the random universal transformer when only the embedding and unembedding are trained.

5.2.2 Embedding dimension scaling

We next study how performance empirically depends on the embedding dimension m. Our theory shows that the universal transformer exists when the embedding dimension grows with the number of attention paths, on the order of H^L up to factors depending on d and d_{in}, but this does not account for the optimization difficulty of finding the desired embedding. Therefore, we experimentally investigate how large m must be for standard gradient-based training to find a good solution. Figure 2 compares two parameterizations on the 2-hop induction-head task. In the constructed parameterization, we set $m = \overline{m}$ according to Theorem 3.1; in the standard parameterization, we instead use $m = Hd$, matching the usual transformer hidden dimension. This gives the empirical evidence that universal transformer can be found when the embedding dimension m is sufficiently large.

6 Conclusion and future work

In this paper, we study the universal transformer: a fixed-parameter transformer that can simulate any transformer in a prescribed class by changing only the input embedding. In this view, the embedding matrix acts as a description of the target model, while the internal attention and value parameters define a fixed computational template. We gave explicit sparse constructions of universal transformers, extended the construction to the looped weight-tied setting, and showed that universality is ubiquitous: randomly initialized transformers are universal almost surely.

Our theory focuses on attention-only transformers. Extending these results to more realistic architectures with MLP layers, residual connections, and normalization is an important direction for future work. Our empirical results suggest that this extension is especially promising, since these components substantially improve performance in our experiments.

References

[1] Vaggos Chatziafratis, Ioannis Panageas, Clayton Sanford, and Stelios Stavroulakis. On scrambling phenomena for randomly initialized recurrent networks. *Advances in Neural Information Processing Systems*, 35:18501–18513, 2022.

[2] Pin-Yu Chen. Model reprogramming: Resource-efficient cross-domain machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 22584–22591, 2024.

[3] Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. The lottery ticket hypothesis for pre-trained bert networks. In *Advances in Neural Information Processing Systems 33*, pages 15834–15846, 2020.

[4] Noam Chomsky and Marcel P. Schützenberger. The algebraic theory of context-free languages. *Studies in Logic and the Foundations of Mathematics*, 35:118–161, 1963.

[5] Stephen Chung and Hava Siegelmann. Turing completeness of bounded-precision recurrent neural networks. In *Advances in Neural Information Processing Systems 34*, pages 28431–28441, 2021.

[6] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

[7] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019.

[8] Yihe Dong, Lorenzo Noci, Mikhail Khodak, and Mufan Li. Is random attention sufficient for sequence modeling? disentangling trainable components in the transformer. *arXiv preprint arXiv:2506.01115*, 2025.

[9] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

[10] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

[11] Ken-Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):183–192, 1989.

[12] Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *International Conference on Machine Learning*, 2023.

[13] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[14] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems 31*, 2018.

[15] Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks. GMD Report 148, GMD - German National Research Institute for Computer Science, 2001.

[16] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002.

[17] Radford M Neal. Priors for infinite networks. In *Bayesian learning for neural networks*, pages 29–53. Springer, 1996.

[18] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[19] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20*, 2007.

[20] Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Transformers, parallel computation, and logarithmic depth. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[21] Hava T Siegelmann and Eduardo D Sontag. On the computational power of neural nets. In *Fifth Annual ACM Conference on Computational Learning Theory*, pages 440–449, 1992.

[22] Alan M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42):230–265, 1936.

[23] Leslie G Valiant. Universal circuits (preliminary report). In *Proceedings of the eighth annual ACM symposium on Theory of computing*, pages 196–203, 1976.

[24] Colin Wei, Yining Chen, and Tengyu Ma. Statistically meaningful approximation: a case study on approximating turing machines with transformers. *Advances in Neural Information Processing Systems*, 35:12071–12083, 2022.

[25] Chao-Han Huck Yang, Yun-Yun Tsai, and Pin-Yu Chen. Voice2series: Reprogramming acoustic models for time series classification. In *International conference on machine learning*, pages 11808–11819. PMLR, 2021.

[26] Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. Self-attention networks can process bounded hierarchical languages. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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)*, pages 3770–3785, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.292. URL `https://aclanthology.org/2021.acl-long.292/`.

[27] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020.

[28] Ziqian Zhong and Jacob Andreas. Algorithmic capabilities of random transformers. In *Advances in Neural Information Processing Systems 37*, 2024.

A Proof of Theorem 3.1 (universal transformer)

Given an input X^0, let $Y := X^L$ denote the output of the target transformer, and $Z := \tilde{X}^L U$ denote the output of the universal transformer. We prove that for any target transformer, there exists a construction of E such that our constructed universal transformer outputs $Z = Y$ for any input X^0.

A.1 Warmup: single head per layer

In this section, we assume $H = 1$ and drop the h superscript in the notations.

Lemma A.1 (Unrolling)**.** *Assume $H = 1$. For each $l \in [L]$,*

$$X^l = A^l \cdots A^1 X^0 W_V^1 W_O^1 \cdots W_V^l W_O^l$$
$$\tilde{X}^l = \tilde{A}^l \cdots \tilde{A}^1 X^0 E R_V^1 \cdots R_V^l.$$

Lemma A.2 (Equal attention)**.** *Assume $H = 1$, and also that*

$$\begin{cases} W_V^1 W_O^1 \cdots W_V^{l-1} W_O^{l-1} W_Q^l = E R_V^1 \cdots R_V^{l-1} R_Q^l \\ W_V^1 W_O^1 \cdots W_V^{l-1} W_O^{l-1} W_K^l = E R_V^1 \cdots R_V^{l-1} R_K^l \end{cases} \quad \textit{for each } l \in [L]. \tag{1}$$

Then for any $X^0 \in \mathbb{R}^{n \times d_{\text{in}}}$, we have $A^l = \tilde{A}^l$ for each $l \in [L]$.

Proof. The proof is by induction. The base case is $L = 1$. We have

$$X^0 W_Q^1 = X^0 E R_Q^1 = \tilde{X}^0 R_Q^1$$
$$X^0 W_K^1 = X^0 E R_K = \tilde{X}^0 R_K^1$$

by (1) for $l = 1$ and the definition of $\tilde{X}^0$. Using these identities with the definitions of A^1 and $\tilde{A}^1$, we have

$$A^1 = \mathrm{softmax}((X^0 W_Q^1)(X^0 W_K^1)^\intercal) = \mathrm{softmax}((\tilde{X}^0 R_Q^1)(\tilde{X}^0 R_K^1)^\intercal) = \tilde{A}^1,$$

This proves the claim for $L = 1$.

For the inductive step, consider $L > 1$. We assume, as the inductive hypothesis, that for each $l \in [L-1]$, $A^l = \tilde{A}^l$. We need to show that $A^L = \tilde{A}^L$. We have

$$\begin{aligned} X^{L-1} W_Q^L &= A^{L-1} \cdots A^1 X^0 W_V^1 W_O^1 \cdots W_V^{L-1} W_O^{L-1} W_Q^L && \text{(by Lemma A.1)} \\ &= \tilde{A}^{L-1} \cdots \tilde{A}^1 X^0 W_V^1 W_O^1 \cdots W_V^{L-1} W_O^{L-1} W_Q^L && \text{(by inductive hypothesis)} \\ &= \tilde{A}^{L-1} \cdots \tilde{A}^1 X^0 E R_V^1 \cdots R_V^{L-1} R_Q^L && \text{(by (1) for } l = L) \\ &= \tilde{X}^{L-1} R_Q^L && \text{(by Lemma A.1),} \end{aligned}$$

and similarly,

$$X^{L-1} W_K^L = \tilde{X}^{L-1} R_K^L.$$

Therefore

$$A^L = \mathrm{softmax}((X^{L-1} W_Q^L)(X^{L-1} W_K^L)^\intercal) = \mathrm{softmax}((\tilde{X}^{L-1} R_Q^L)(\tilde{X}^{L-1} R_K^L)^\intercal) = \tilde{A}^L.$$

This completes the inductive step. □

Lemma A.3 (Equal input-output behavior)**.** *Assume $H = 1$, and also that both (1) and*

$$W_V^1 W_O^1 \cdots W_V^L W_O^L = E R_V^1 \cdots R_V^L U \tag{2}$$

hold. Then for any $X^0 \in \mathbb{R}^{n \times d_{\text{in}}}$, we have $Y = Z$.

Proof. We have

$$\begin{aligned} Y = X^L &= A^L \cdots A^1 X^0 W_V^1 W_O^1 \cdots W_V^L W_O^L && \text{(by Lemma A.1)} \\ &= \tilde{A}^L \cdots \tilde{A}^1 X^0 W_V^1 W_O^1 \cdots W_V^L W_O^L && \text{(by Lemma A.2)} \\ &= \tilde{A}^L \cdots \tilde{A}^1 X^0 E R_V^1 \cdots R_V^L U && \text{(by (2))} \\ &= \tilde{X}^L U = Z && \text{(by Lemma A.1).} \end{aligned}$$

□

Lemma A.4 (Construction for single head)**.** *Assume $H = 1$ and $m \geq (L+1)\max\{2d, d_{\text{in}}\}$. Then there exists fixed $\{R_Q^l, R_K^l, R_V^l\}_{l\in[L]}$ and U such that for any transformer $T \in \text{TF}_{1,L,d_{\text{in}},d}$, there exists an E so that (1) and (2) hold.*

Proof. Let $b := \max\{2d, d_{\text{in}}\}$. Since $m \geq (L+1)b$, we partition the first $(L+1)b$ coordinates of the embedding dimension into $L+1$ consecutive blocks

$$B_1, \ldots, B_{L+1}, \qquad |B_i| = b.$$

Any remaining coordinates are unused.

We choose R_Q^l, R_K^l and U to be selector matrices. More precisely, R_Q^l selects the first d coordinates of the block B_l, R_K^l selects the next d coordinates of the block B_l, and U selects the first d_{in} coordinates of the block B_{L+1}. These matrices are the same for every target transformer.

We choose R_V^l to be the block-shift matrix satisfying $ER_V^1 \cdots R_V^{l-1}$ has, in block B_l, the contents originally stored in block B_1 after the appropriate shifts. Equivalently, for each l, multiplying by R_V^l shifts the information one block to the left, so that the block used by R_Q^{l+1} and R_K^{l+1} contains the next stored matrices. All other unused coordinates may be set to zero.

Now fix an arbitrary target transformer $T \in \text{TF}_{1,L,d_{\text{in}},d}$. Let

$$W_{VO}^{1:l} := W_V^1 W_O^1 \cdots W_V^l W_O^l, \qquad W_{VO}^{1:0} := I_{d_{\text{in}}}.$$

We construct $E \in \mathbb{R}^{d_{\text{in}}\times m}$ block by block as follows. For each $l \in [L]$, fill the first d columns of block B_l with $W_{VO}^{1:l-1}W_Q^l$, and fill the next d columns of block B_l with $W_{VO}^{1:l-1}W_K^l$. Finally, fill the first d_{in} columns of block B_{L+1} with $W_{VO}^{1:L}$. All remaining entries of E are set arbitrarily, say to zero.

By construction, for every $l \in [L]$, after applying $R_V^1 \cdots R_V^{l-1}$, the block selected by R_Q^l contains $W_{VO}^{1:l-1}W_Q^l$, and the block selected by R_K^l contains $W_{VO}^{1:l-1}W_K^l$. Hence

$$ER_V^1 \cdots R_V^{l-1}R_Q^l = W_{VO}^{1:l-1}W_Q^l$$

and

$$ER_V^1 \cdots R_V^{l-1}R_K^l = W_{VO}^{1:l-1}W_K^l.$$

Therefore (1) holds.

Similarly, after applying $R_V^1 \cdots R_V^L$, the block selected by U contains $W_{VO}^{1:L}$. Therefore

$$ER_V^1 \cdots R_V^L U = W_{VO}^{1:L} = W_V^1 W_O^1 \cdots W_V^L W_O^L,$$

so (2) also holds.

Since the matrices R_Q^l, R_K^l, R_V^l and U depend only on L, d, d_{in} and not on the target transformer T, the construction is universal. This completes the proof. □

A.2 Multiple heads per layer

Lemma A.5 (Multi-headed Unrolling)**.** *For each $l \in [L]$,*

$$X^l = \sum_{(h_1,\ldots,h_l)\in[H]^l} A^{l,h_l} \cdots A^{1,h_1} X^0 W_V^{1,h_1} W_O^{1,h_1} \cdots W_V^{l,h_l} W_O^{l,h_l}$$
$$\tilde{X}^l = \sum_{(h_1,\ldots,h_l)\in[H]^l} \tilde{A}^{l,h_l} \cdots \tilde{A}^{1,h_1} X^0 E R_V^{1,h_1} \cdots R_V^{l,h_l}.$$

Proof. We prove the first identity; the second is identical with $(W_V^{t,h_t}W_O^{t,h_t})$ replaced by ER_V^{t,h_t} and (A^{t,h_t}) replaced by $(\tilde{A}^{t,h_t})$. For $l = 1$, the update rule gives

$$X^1 = \sum_{h_1=1}^{H} A^{1,h_1} X^0 W_V^{1,h_1} W_O^{1,h_1},$$

which is exactly the stated formula. Assume the identity holds for $l-1 \geq 1$. Then

$$\begin{aligned} X^l &= \sum_{h_l=1}^{H} A^{l,h_l} X^{l-1} W_V^{l,h_l} W_O^{l,h_l} \\ &= \sum_{h_l=1}^{H} A^{l,h_l} \left(\sum_{(h_1,\ldots,h_{l-1})\in[H]^{l-1}} A^{l-1,h_{l-1}} \cdots A^{1,h_1} X^0 \prod_{t=1}^{l-1} W_V^{t,h_t} W_O^{t,h_t} \right) W_V^{l,h_l} W_O^{l,h_l} \\ &= \sum_{(h_1,\ldots,h_l)\in[H]^l} A^{l,h_l} \cdots A^{1,h_1} X^0 \prod_{t=1}^{l} W_V^{t,h_t} W_O^{t,h_t}, \end{aligned}$$

which is the desired expansion. □

Lemma A.6 (Equal attention for multi-head)**.** *For each $l \in [L]$,*

$$\begin{cases} W_V^{1,h_1} W_O^{1,h_1} \cdots W_V^{l-1,h_{l-1}} W_O^{l-1,h_{l-1}} W_Q^{l,h_l} = E R_V^{1,h_1} \cdots R_V^{l-1,h_{l-1}} R_Q^{l,h_l} \\ W_V^{1,h_1} W_O^{1,h_1} \cdots W_V^{l-1,h_{l-1}} W_O^{l-1,h_{l-1}} W_K^{l,h_l} = E R_V^{1,h_1} \cdots R_V^{l-1,h_{l-1}} R_K^{l,h_l} \end{cases} \quad \textit{for each } (h_1,\ldots,h_l) \in [H]^l. \tag{3}$$

Then for any $X^0 \in \mathbb{R}^{n\times d_{\text{in}}}$, we have $A^{l,h_l} = \tilde{A}^{l,h_l}$ for each $h_l \in [H], l \in [L]$.

Proof. Fix a layer $l \in [L]$ and a head $h_l \in [H]$. It suffices to show that

$$X^{l-1} W_Q^{l,h_l} = \tilde{X}^{l-1} R_Q^{l,h_l} \quad \text{and} \quad X^{l-1} W_K^{l,h_l} = \tilde{X}^{l-1} R_K^{l,h_l},$$

since then by the definitions of attention,

$$A^{l,h_l} = \operatorname{softmax}\big((X^{l-1} W_Q^{l,h_l})(X^{l-1} W_K^{l,h_l})^\top\big) = \operatorname{softmax}\big((\tilde{X}^{l-1} R_Q^{l,h_l})(\tilde{X}^{l-1} R_K^{l,h_l})^\top\big) = \tilde{A}^{l,h_l}.$$

By Lemma A.5, we can write

$$X^{l-1} W_Q^{l,h_l} = \sum_{(h_1,\ldots,h_{l-1})\in[H]^{l-1}} A^{l-1,h_{l-1}} \cdots A^{1,h_1}\, X^0 \Big(W_V^{1,h_1} W_O^{1,h_1} \cdots W_V^{l-1,h_{l-1}} W_O^{l-1,h_{l-1}} W_Q^{l,h_l} \Big).$$

Using the assumption (3), each parenthesized product equals

$$E\, R_V^{1,h_1} \cdots R_V^{l-1,h_{l-1}} R_Q^{l,h_l}.$$

Substituting and factoring out the rightmost R_Q^{l,h_l} gives

$$X^{l-1} W_Q^{l,h_l} = \sum_{(h_1,\ldots,h_{l-1})\in[H]^{l-1}} A^{l-1,h_{l-1}} \cdots A^{1,h_1}\, X^0 E\, R_V^{1,h_1} \cdots R_V^{l-1,h_{l-1}} R_Q^{l,h_l}.$$

If we assume (inductively in t) that for all $t \leq l-1$ and all heads h_t, we already have $A^{t,h_t} = \tilde{A}^{t,h_t}$, then the above becomes

$$\begin{aligned} X^{l-1} W_Q^{l,h_l} &= \sum_{(h_1,\ldots,h_{l-1})\in[H]^{l-1}} \tilde{A}^{l-1,h_{l-1}} \cdots \tilde{A}^{1,h_1}\, X^0 E\, R_V^{1,h_1} \cdots R_V^{l-1,h_{l-1}} R_Q^{l,h_l} \\ &= \tilde{X}^{l-1} R_Q^{l,h_l}, \end{aligned}$$

by the second identity in Lemma A.5. The W_K/R_K case is identical.

Finally, we prove the claim layer-by-layer. For $l = 1$, the computation of A^{1,h_1} only depends on X^0 and the pair $(W_Q^{1,h_1}, W_K^{1,h_1})$, and the assumption (3) (with $l = 1$) directly gives $X^0 W_Q^{1,h_1} = \tilde{X}^0 R_Q^{1,h_1}$ and $X^0 W_K^{1,h_1} = \tilde{X}^0 R_K^{1,h_1}$, hence $A^{1,h_1} = \tilde{A}^{1,h_1}$. Now assume for some $l \geq 2$ that $A^{t,h} = \tilde{A}^{t,h}$ holds for all $t \leq l-1$ and all heads $h \in [H]$. Repeating the argument above with this inductive hypothesis yields $X^{l-1} W_Q^{l,h_l} = \tilde{X}^{l-1} R_Q^{l,h_l}$ and $X^{l-1} W_K^{l,h_l} = \tilde{X}^{l-1} R_K^{l,h_l}$ for each h_l, and therefore $A^{l,h_l} = \tilde{A}^{l,h_l}$ for each h_l. This completes the induction. □

Lemma A.7 (Equal input-output behavior for multi-head)**.** *Assume (3) and*

$$W_V^{1,h_1} W_O^{1,h_1} \cdots W_V^{L,h_L} W_O^{L,h_L} = E R_V^{1,h_1} \cdots R_V^{L,h_L} U \quad \textit{for each } (h_1, \ldots, h_L) \in [H]^L \tag{4}$$

hold. Then for any $X^0 \in \mathbb{R}^{n \times d_{\text{in}}}$, we have $Y = Z$.

Proof. Fix $X^0 \in \mathbb{R}^{n \times d_{\text{in}}}$. By Lemma A.6, we have

$$A^{l,h} = \tilde{A}^{l,h} \qquad \text{for all } l \in [L],\ h \in [H].$$

Using the multi-headed unrolling Lemma A.5 at $l = L$, we can write

$$\begin{aligned} Y = X^L &= \sum_{(h_1,\ldots,h_L) \in [H]^L} A^{L,h_L} \cdots A^{1,h_1}\, X^0 \Big(W_V^{1,h_1} W_O^{1,h_1} \cdots W_V^{L,h_L} W_O^{L,h_L} \Big) \\ &= \sum_{(h_1,\ldots,h_L) \in [H]^L} \tilde{A}^{L,h_L} \cdots \tilde{A}^{1,h_1}\, X^0 \Big(W_V^{1,h_1} W_O^{1,h_1} \cdots W_V^{L,h_L} W_O^{L,h_L} \Big). \end{aligned}$$

By the assumption (4), each product in parentheses equals

$$E\, R_V^{1,h_1} \cdots R_V^{L,h_L} U.$$

Substituting this in, and then applying Lemma A.5 to recognize the resulting sum as $\tilde{X}^L$, yields

$$\begin{aligned} Y &= \sum_{(h_1,\ldots,h_L) \in [H]^L} \tilde{A}^{L,h_L} \cdots \tilde{A}^{1,h_1}\, X^0 E\, R_V^{1,h_1} \cdots R_V^{L,h_L} U \\ &= \tilde{X}^L U \\ &= Z, \end{aligned}$$

completing the proof. □

Theorem A.8 (Fixed sparse R_Q, R_K, R_V)**.** *Assume $H > 1$ and*

$$m = \sum_{t=0}^{L-1} H^t \cdot (2Hd) \ + \ H^L d_{\text{in}} \ = \ O(H^L d + H^L d_{\text{in}}).$$

There exist matrices $\{R_Q^{l,h}, R_K^{l,h}\}_{l \in [L], h \in [H]}$ with $R_Q^{l,h}, R_K^{l,h} \in \mathbb{R}^{m \times d}$, routing maps $\{R_V^{l,h}\}_{l \in [L], h \in [H]}$ with $R_V^{l,h} \in \mathbb{R}^{m \times m}$ and $U \in \mathbb{R}^{m \times d_{\text{in}}}$ such that for any target transformer $T \in \mathrm{TF}_{H,L,d_{\text{in}},d}$, there exists $E \in \mathbb{R}^{d_{\text{in}} \times m}$ so that (3) and (4) hold.

Proof. We first state the high level idea for the contruction of E.

Construction for E We divide E into different chucks to construct an H-ary tree (See Figure 3)

- a chunk for storing the parameters of W_Q's for all H heads in the first layer
- a chunk for storing the parameters of W_K's for all H heads in the first layer
- chunks for others parameters of different heads: $E_{l=1,h_1=1}, \ldots, E_{l=1,h_1=H}$

Correspondingly, let R_Q^{1,h_l} and R_K^{1,h_l} be the linear transformation that selects the columns where W_Q^{1,h_l} and W_K^{1,h_l} are respectively. These satisfy the equations in (3) for $l = 1$. Let R_V^{1,h_l} be the linear transformation that selects the columns where E_{1,h_1} is.

Define the next layer similarly, for each $E_{l=1,h_1}$,

- the first chunk store $W_V^{(1,h_1)} W_O^{(1,h_1)} W_Q^{(2,1)}, \ldots, W_V^{(1,h_1)} W_O^{(1,h_1)} W_Q^{(2,H)}$
- the second chunk store $W_V^{(1,h_1)} W_O^{(1,h_1)} W_K^{(2,1)}, \ldots, W_V^{(1,h_1)} W_O^{(1,h_1)} W_K^{(2,H)}$
- The rest of the chunks are divided into $E_{l=2,h_1,h_2=1}, \ldots, E_{l=2,h_1,h_2=H}$

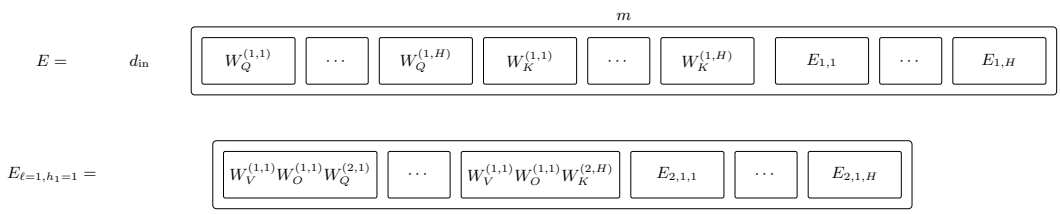


Figure 3: Block structure of E and a schematic recursive expansion.

Likewise, R_Q^{2,h_2} and R_K^{2,h_2} selects the corresponding Q, K part in each $E_{l=1,h_1}$, and R_V^{2,h_2} selects $E_{l=2,h_1,h_2}$.

Do this recursively until the L-th layer, where we set each $E_{L,h_1,\ldots,h_L} = W_V^{1,h_1} W_O^{1,h_1} \cdots W_V^{L,h_L} W_O^{L,h_L}$.

We formalize it below. For convenience define

$$M^{l,h} \;=\; W_V^{l,h} W_O^{l,h} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}, \qquad M^{1:t,(h_1,\ldots,h_t)} \;=\; \prod_{s=1}^{t} M^{s,h_s} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}},$$

with the convention $M^{1:0,\emptyset} = I_{d_{\text{in}}}$.

Coordinate blocks (tree indexing). We partition the m embedding coordinates into disjoint blocks indexed by prefixes. For each depth $t \in \{0, 1, \ldots, L-1\}$ and each prefix $\mathbf{h}_{1:t} = (h_1, \ldots, h_t) \in [H]^t$, allocate an *internal block* $B_{t,\mathbf{h}_{1:t}} \subset [m]$ of width $2Hd$. For each leaf $\mathbf{h}_{1:L} \in [H]^L$, allocate a *leaf block* $C_{\mathbf{h}_{1:L}} \subset [m]$ of width d_{in}. All these blocks are disjoint and cover $[m]$.

Inside each internal block $B_{t,\mathbf{h}_{1:t}}$, further partition into $2H$ subblocks:

$$B_{t,\mathbf{h}_{1:t}} = \Big(\bigsqcup_{h=1}^{H} B^{Q}_{t,\mathbf{h}_{1:t};h}\Big) \;\sqcup\; \Big(\bigsqcup_{h=1}^{H} B^{K}_{t,\mathbf{h}_{1:t};h}\Big), \qquad |B^{Q}_{t,\mathbf{h}_{1:t};h}| = |B^{K}_{t,\mathbf{h}_{1:t};h}| = d.$$

Intuitively, $B^{Q}_{t,\mathbf{h}_{1:t};h}$ stores a $d_{\text{in}} \times d$ matrix that will act like $M^{1:t,\mathbf{h}_{1:t}} W_Q^{t+1,h}$, and similarly for K.

Define routing maps $R_V^{l,h}$. For each layer $l \in [L]$ and head $h \in [H]$, define $R_V^{l,h} \in \mathbb{R}^{m \times m}$ as a sparse block-routing operator that moves the representation from depth $l-1$ to its h-child at depth l:

- For each prefix $\mathbf{h}_{1:l-1} \in [H]^{l-1}$, $R_V^{l,h}$ maps the internal block $B_{l-1,\mathbf{h}_{1:l-1}}$ into the child internal block $B_{l,(\mathbf{h}_{1:l-1},h)}$ if $l < L$, and into the leaf block $C_{(\mathbf{h}_{1:l-1},h)}$ if $l = L$.
- It sends all other blocks (all depths $\neq l-1$, and all leaves) to 0.

Concretely, one may take $R_V^{l,h}$ to be a block-embedding map that copies the entire block $B_{l-1,\mathbf{h}_{1:l-1}}$ into the designated child block (padding with zeros as needed).

Define the unembedding U. Define $U \in \mathbb{R}^{m \times d_{\text{in}}}$ to read the leaf blocks and ignore internal blocks: on each leaf block $C_{\mathbf{h}_{1:L}}$ it acts as the $d_{\text{in}} \times d_{\text{in}}$ identity, and on each internal block $B_{t,\mathbf{h}_{1:t}}$ it is zero. Equivalently, for $E \in \mathbb{R}^{d_{\text{in}} \times m}$,

$$E R_V^{1,h_1} \cdots R_V^{L,h_L} U \;=\; E_{C_{\mathbf{h}_{1:L}}} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}},$$

where E_S denotes the sub-matrix of E consisting of columns indexed by S.

Define selectors $R_Q^{l,h}, R_K^{l,h}$. For each (l,h) define $R_Q^{l,h} \in \mathbb{R}^{m \times d}$ and $R_K^{l,h} \in \mathbb{R}^{m \times d}$ to select (and sum across prefixes) the appropriate Q-subblocks and K-subblocks at depth $l-1$:

$$E R_V^{1,h_1} \cdots R_V^{l-1,h_{l-1}} R_Q^{l,h} \;=\; E_{B^{Q}_{l-1,\mathbf{h}_{1:l-1};h}} \in \mathbb{R}^{d_{\text{in}} \times d}$$

$$E R_V^{1,h_1} \cdots R_V^{l-1,h_{l-1}} R_K^{l,h} \;=\; E_{B^{K}_{l-1,\mathbf{h}_{1:l-1};h}} \in \mathbb{R}^{d_{\text{in}} \times d}$$

Equivalently, as matrices, $R_Q^{l,h}$ (resp. $R_K^{l,h}$) has an I_d on each chosen subblock and zeros elsewhere.

Define the embedding E by filling each block. We now construct $E \in \mathbb{R}^{d_{\text{in}} \times m}$ block-by-block.

(Internal blocks.) For each depth $t \in \{0, \dots, L-1\}$, prefix $\mathbf{h}_{1:t} \in [H]^t$, and head $h \in [H]$, set

$$E_{B^Q_{t,\mathbf{h}_{1:t};h}} \;=\; M^{1:t,\mathbf{h}_{1:t}}\, W_Q^{t+1,h} \in \mathbb{R}^{d_{\text{in}} \times d}, \qquad E_{B^K_{t,\mathbf{h}_{1:t};h}} \;=\; M^{1:t,\mathbf{h}_{1:t}}\, W_K^{t+1,h} \in \mathbb{R}^{d_{\text{in}} \times d}.$$

(For $t = 0$ this means the root block stores $\{W_Q^{1,h}, W_K^{1,h}\}_{h=1}^H$.)

(Leaf blocks.) For each leaf $\mathbf{h}_{1:L} \in [H]^L$, set

$$E_{C_{\mathbf{h}_{1:L}}} \;=\; M^{1:L,\mathbf{h}_{1:L}} \;=\; \prod_{s=1}^{L} W_V^{s,h_s} W_O^{s,h_s} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}.$$

Verify the multi-head QK equalities. Fix a layer $l \in [L]$ and a head $h_l \in [H]$. Let $\mathbf{h}_{1:l-1} \in [H]^{l-1}$ be an arbitrary prefix. By construction of the routing operators, the product $R_V^{1,h_1} \cdots R_V^{l-1,h_{l-1}}$ routes the representation into the internal block $B_{l-1,\mathbf{h}_{1:l-1}}$ (and annihilates other internal blocks). Then R_Q^{l,h_l} selects exactly the subblock $B^Q_{l-1,\mathbf{h}_{1:l-1};h_l}$. Hence

$$E\, R_V^{1,h_1} \cdots R_V^{l-1,h_{l-1}}\, R_Q^{l,h_l} \;=\; E_{B^Q_{l-1,\mathbf{h}_{1:l-1};h_l}} \;=\; M^{1:l-1,\mathbf{h}_{1:l-1}}\, W_Q^{l,h_l}.$$

The same argument yields

$$E\, R_V^{1,h_1} \cdots R_V^{l-1,h_{l-1}}\, R_K^{l,h_l} \;=\; E_{B^K_{l-1,\mathbf{h}_{1:l-1};h_l}} \;=\; M^{1:l-1,\mathbf{h}_{1:l-1}}\, W_K^{l,h_l}.$$

Therefore the assumptions of Lemma A.6 (equation (3)) hold, and for every input X^0,

$$A^{l,h} = \tilde{A}^{l,h} \qquad \text{for all } l \in [L],\ h \in [H].$$

Verify the multi-head OV equalities on leaves. Fix a full path $\mathbf{h}_{1:L} \in [H]^L$. By the routing construction, $R_V^{1,h_1} \cdots R_V^{L,h_L}$ routes the representation into the leaf block $C_{\mathbf{h}_{1:L}}$, and then U reads that leaf block as identity. Thus

$$E\, R_V^{1,h_1} \cdots R_V^{L,h_L}\, U \;=\; E_{C_{\mathbf{h}_{1:L}}} \;=\; M^{1:L,\mathbf{h}_{1:L}} \;=\; W_V^{1,h_1} W_O^{1,h_1} \cdots W_V^{L,h_L} W_O^{L,h_L}.$$

Hence equation (4) holds.

Since (3) and (4) hold, Lemma A.2 (multi-head version) and Lemma A.5 imply that for every input X^0, the target transformer output $Y = X^L$ equals the universal transformer output $Z = \tilde{X}^L U$. This proves the theorem. □

B Proof of Theorem 3.3 (random universal transformer)

Theorem B.1 (Random parameters)**.** *Assume $m \geq O(H^L d + H^L d_{\text{in}})$. Draw $\{R_Q^{l,h}, R_K^{l,h}, R_V^{l,h}\}_{l \in [L],\, h \in [H]}$ and U randomly with each entry i.i.d. from an absolute continuous distribution. Then with probability 1, for any target transformer $T \in \mathrm{TF}_{H,L,d_{\text{in}},d}$, there exists E such that (3) and (4) hold.*

Proof. (3) and (4) have $\frac{H(H^L-1)}{H-1}$ and H^L linear equations for E. Let $M = \frac{H(H^L-1)}{H-1} d + H^L d_{\text{in}}$. We can rewrite this system of equations into the following form

$$E R_{\text{all}} = W_{\text{all}}, \tag{5}$$

where $R_{\text{all}} \in \mathbb{R}^{m\times M}$ and $W_{\text{all}} \in \mathbb{R}^{d_{\text{in}}\times M}$ are formed by horizontal concatenation of the corresponding blocks:

$$\begin{aligned} R_{\text{all}} = \big[\ & \{\, R_V^{1,h_1}\cdots R_V^{l-1,h_{l-1}} R_Q^{l,h_l} \,\}_{(h_1,\dots,h_l,l)} \\ & \{\, R_V^{1,h_1}\cdots R_V^{l-1,h_{l-1}} R_K^{l,h_l} \,\}_{(h_1,\dots,h_l,l)} \\ & \{\, R_V^{1,h_1}\cdots R_V^{L,h_L} U \,\}_{(h_1,\dots,h_L)} \,\big], \\ W_{\text{all}} = \big[\ & \{\, W_V^{1,h_1} W_O^{1,h_1}\cdots W_V^{l-1,h_{l-1}} W_O^{l-1,h_{l-1}} W_Q^{l,h_l} \,\}_{(h_1,\dots,h_l,l)} \\ & \{\, W_V^{1,h_1} W_O^{1,h_1}\cdots W_V^{l-1,h_{l-1}} W_O^{l-1,h_{l-1}} W_K^{l,h_l} \,\}_{(h_1,\dots,h_l,l)} \\ & \{\, W_V^{1,h_1} W_O^{1,h_1}\cdots W_V^{L,h_L} W_O^{L,h_L} \,\}_{(h_1,\dots,h_L)} \,\big], \end{aligned}$$

with $(h_1,\dots,h_l)\in[H]^l$ for each $l\in[L]$ in the first two groups, and $(h_1,\dots,h_L)\in[H]^L$ in the last group.

If R_{all} has left pseudoinverse, denoted by $R_{\text{all}}^{\dagger}$, then E has a solution of the form $E = W_{\text{all}} R_{\text{all}}^{\dagger}$, which requires R_{all} to have linearly independent columns.

Reduce to a nonvanishing determinant polynomial. Since $m \ge M$, R_{all} has full column rank iff there exists an $M\times M$ minor with nonzero determinant. Fix any deterministic rule that selects M distinct rows of R_{all} (for example the first M rows), and let

$$\bar{R}_{\text{all}} \in \mathbb{R}^{M\times M}$$

denote the resulting square submatrix. Define

$$p(\omega) \;=\; \det(\bar{R}_{\text{all}}),$$

viewed as a function of the underlying random draw ω (i.e. the entries of all random matrices $\{R_Q^{l,h}, R_K^{l,h}, R_V^{l,h}\}$). Each entry of R_{all} is a polynomial (in fact, a multilinear polynomial) in the entries of these random matrices, because R_{all} is assembled from products of the form $R_V^{1,h_1}\cdots R_V^{t,h_t} R_Q^{t+1,h}$, $R_V^{1,h_1}\cdots R_V^{t,h_t} R_K^{t+1,h}$, and $R_V^{1,h_1}\cdots R_V^{L,h_L} U$. Hence $p(\omega)$ is a (real) polynomial in finitely many independent Gaussian variables (and any additional independent continuous variables used to sample the $R_V^{l,h}$).

If p is not the zero polynomial, then its zero set has Lebesgue measure 0, and thus

$$\Pr\big[p(\omega)=0\big] = 0.$$

In particular, with probability 1 we have $p(\omega)\neq 0$, so $\bar{R}_{\text{all}}$ is invertible and therefore $\operatorname{rank}(R_{\text{all}}) = M$. It remains to prove that p is not the identically-zero polynomial.

One assignment with full rank using the explicit construction. We show there exists a deterministic choice of matrices $\{R_Q^{l,h}, R_K^{l,h}, R_V^{l,h}, U\}$ for which R_{all} has full column rank. This implies that p cannot be identically zero, because p evaluates to a nonzero number on that assignment.

Consider the explicit *designed* construction from Theorem A.8 (the previous theorem): there we fixed

$$m_0 \;=\; \sum_{t=0}^{L-1} H^t\cdot(2Hd) \;+\; H^L d_{\text{in}} \;=\; M$$

and built block-routing matrices $\{\widehat{R}_V^{l,h}\}$, block-selectors $\{\widehat{R}_Q^{l,h}, \widehat{R}_K^{l,h}\}$, and an unembedding $\widehat{U}$, together with a block-structured embedding $\widehat{E}$, such that (3) and (4) hold for *every* target transformer. In particular, in that construction, the block-columns appearing in

$$\widehat{R}_{\text{all}} \;=\; \big[\ \{\widehat{R}_V^{1,h_1}\cdots\widehat{R}_V^{l-1,h_{l-1}}\widehat{R}_Q^{l,h_l}\}_{(h_1,\dots,h_l,l)} \quad \{\widehat{R}_V^{1,h_1}\cdots\widehat{R}_V^{l-1,h_{l-1}}\widehat{R}_K^{l,h_l}\}_{(h_1,\dots,h_l,l)} \quad \{\widehat{R}_V^{1,h_1}\cdots\widehat{R}_V^{L,h_L}\widehat{U}\}_{(h_1,\dots,h_L)}\ \big]$$

are supported on *disjoint coordinate blocks* of $\mathbb{R}^m$: each column block corresponds to a unique node (or leaf) in the H-ary prefix tree and occupies a unique set of rows. Consequently, after a suitable permutation of the rows of $\widehat{R}_{\text{all}}$ (which does not change its rank), the matrix becomes block diagonal with identity blocks on the diagonal:

$$P\,\widehat{R}_{\text{all}} \;=\; \operatorname{diag}\big(I_d,\dots,I_d,\ I_d,\dots,I_d,\ I_{d_{\text{in}}},\dots,I_{d_{\text{in}}}\big) \in \mathbb{R}^{M\times M},$$

where the first group of I_d's corresponds to all Q-constraints, the second group to all K-constraints, and the final group of $I_{d_{\text{in}}}$'s to all OV-constraints. In particular,

$$\operatorname{rank}(\widehat{R}_{\text{all}}) = M, \qquad \text{and hence} \qquad \det(\widehat{\bar{R}}_{\text{all}}) \neq 0$$

for the corresponding $M \times M$ minor $\widehat{\bar{R}}_{\text{all}}$ (take $\widehat{\bar{R}}_{\text{all}}$ to be the full matrix when $m_0 = M$).

Thus there exists at least one assignment of the underlying variables for which $p(\omega) \neq 0$. Therefore p is not the zero polynomial.

Full column rank almost surely. Since p is a nonzero polynomial in continuously distributed random variables, its zero set has measure 0. Hence $\Pr[p(\omega) = 0] = 0$, and with probability 1 we have $p(\omega) \neq 0$. On this event, $\bar{R}_{\text{all}}$ is invertible and thus R_{all} has full column rank M. Therefore (5) has a solution; taking any left pseudoinverse $R_{\text{all}}^{\dagger}$ and setting $E = W_{\text{all}} R_{\text{all}}^{\dagger}$ yields (3) and (4). This completes the proof. □

C Proof of Theorem 3.2 (looped universal transformer)

We further improve our result to Looped Transformer (weight-tying for each layer).

The multi-head unrolling and equal attention, equal input-output behavior works here by weight-tying. Since the construction in Lemma A.4 already satisfies the weight-typing property, we focus on multiple heads ($H > 1$) in the proof.

Theorem C.1 (Designed R_Q, R_K, R_V for looped transformer)**.** *Assume $H > 1$. Let*

$$m = \sum_{t=1}^{L-1} H^t \cdot \max\{d_{\text{in}}, 2d\} + (H^L + 1)\max\{d_{\text{in}}, 2d\} = O(H^L \max\{d_{\text{in}}, 2d\})$$

Then there exist matrices $\{R_V^h, R_Q^h, R_K^h\}_{h\in[H]}$ and U such that for any target transformer $T \in \mathrm{TF}_{H,L,d_{\text{in}},d}$, there exists $E \in \mathbb{R}^{d_{\text{in}}\times m}$ so that (3) and (4) hold.

Proof. Let $M^h := W_V^h W_O^h \in \mathbb{R}^{d_{\text{in}}\times d_{\text{in}}}$ and for any prefix $p = (h_1, \ldots, h_t) \in [H]^t$, define

$$M_p := M^{h_1} \cdots M^{h_t}, \qquad M_\phi := I_{d_{\text{in}}}.$$

Coordinate blocks and selectors. Partition the m coordinates into disjoint blocks indexed by prefixes

$$[m] = \bigsqcup_{t=0}^{L} \bigsqcup_{p\in[H]^t} B_p,$$

where each block has width d_{in}. For each p, define $B_p \in \mathbb{R}^{m\times d_{\text{in}}}$ to be the selector matrix extracting the coordinates of block B_p.

Define embedding E. Define $E \in \mathbb{R}^{d_{\text{in}}\times m}$ blockwise as follows.

- For the empty prefix ϕ, set
$$E_{B_\phi} = M_\phi$$
This block has dimension d_{in}.
- For each prefix p with $|p| < L$ and each $h \in [H]$, set
$$E_{B_{(p,h)}} = \begin{bmatrix} M_p W_Q^h & M_p W_K^h \end{bmatrix} \in \mathbb{R}^{d_{\text{in}}\times 2d}$$
Each of these blocks takes dimension $2d$.
- For each prefix p with $|p| = L$, set
$$E_{B_p} = M_p$$
Each of these blocks takes dimension d_{in}.

Assume $d_{\text{in}} = 2d$ to have each block to be the same size. Otherwise, we can pad each block with 0 columns.

Therefore, the total embedding dimension m needed is

$$\sum_{t=1}^{L-1} H^t \cdot \max\{d_{\text{in}}, 2d\} + (H^L + 1)\max\{d_{\text{in}}, 2d\} = O(H^L \max\{d_{\text{in}}, 2d\})$$

Define the readout matrices

$$U := B_\phi = \begin{bmatrix} I_{d_{\text{in}}} \\ 0 \end{bmatrix} \in \mathbb{R}^{m \times d_{\text{in}}}, \qquad U_Q := \begin{bmatrix} I_d \\ 0 \end{bmatrix} \in \mathbb{R}^{m \times d}, \qquad U_K := \begin{bmatrix} 0 \\ I_d \\ 0 \end{bmatrix} \in \mathbb{R}^{m \times d}.$$

where U is the final un-embedding layer that readout the value and output; U_Q and U_K are intermediate readout matrices for W_Q and W_K

Define routing operators R_V^h, R_Q^h and R_K^h. By Lemma C.2, for each $h \in [H]$ there exists a matrix $R_V^h \in \mathbb{R}^{m \times m}$ such that

$$R_V^h B_p = B_{(h,p)} \quad \text{for all } |p| < L, \qquad R_V^h B_p = 0 \quad \text{for all } |p| = L.$$

Therefore, by induction on t, for any prefix $p = (h_1, \ldots, h_t)$,

$$R_V^{h_1} \cdots R_V^{h_t} U = B_p.$$

Lemma C.2 (Existence of tied routing operators)**.** *Let*

$$[m] = \bigsqcup_{t=0}^{L} \bigsqcup_{p \in [H]^t} B_p$$

be a partition of the coordinates into disjoint blocks, where each block B_p has width d_{in}. For each prefix p, let $B_p \in \mathbb{R}^{m \times d_{\text{in}}}$ denote the selector matrix extracting the coordinates of block B_p. Then for each $h \in [H]$, there exists a matrix $R_V^h \in \mathbb{R}^{m \times m}$ such that

$$R_V^h B_p = B_{(h,p)} \qquad \textit{for every } p \textit{ with } |p| < L,$$

and

$$R_V^h B_p = 0 \qquad \textit{for every } p \textit{ with } |p| = L.$$

Proof. Fix $h \in [H]$. Since the blocks $\{B_p\}$ are disjoint and each has width d_{in}, for every prefix p with $|p| < L$ there is a unique coordinate-wise identification between the coordinates of B_p and those of its child block $B_{(h,p)}$.

Define R_V^h on the standard basis of $\mathbb{R}^m$ as follows. For each prefix p with $|p| < L$, and each of the d_{in} coordinates inside block B_p, send the j-th coordinate of B_p to the j-th coordinate of block $B_{(h,p)}$. For coordinates belonging to leaf blocks B_p with $|p| = L$, send them to 0. For all remaining coordinates (there are none, since the blocks partition $[m]$), define the action arbitrarily.

Equivalently, if we write

$$B_p = [\, e_{j_1(p)}, \ldots, e_{j_{d_{\text{in}}}(p)} \,], \qquad B_{(h,p)} = [\, e_{j_1(h,p)}, \ldots, e_{j_{d_{\text{in}}}(h,p)} \,].$$

where e_i denotes the i-th standard basis vector of $\mathbb{R}^m$, then set

$$R_V^h e_{i_j^{(p)}} := e_{i_j^{(h,p)}} \quad (|p| < L,\ j \in [d_{\text{in}}]),$$

and

$$R_V^h e_{i_j^{(p)}} := 0 \quad (|p| = L,\ j \in [d_{\text{in}}]).$$

Extending linearly defines a matrix $R_V^h \in \mathbb{R}^{m \times m}$.

By construction, for every p with $|p| < L$,

$$R_V^h B_p = [\, R_V^h e_{j_1(p)}, \ldots, R_V^h e_{j_{d_{\text{in}}}(p)} \,] = [\, e_{j_1(h,p)}, \ldots, e_{j_{d_{\text{in}}}(h,p)} \,] = B_{(p,h)},$$

and for every p with $|p| = L$,

$$R_V^h B_p = 0.$$

This proves the claim. □

Define

$$R_Q^h := R_V^h U_Q, \qquad R_K^h := R_V^h U_K.$$

which route to the corresponding block first and read out the Q and K part respectively.

Verify QK equalities. Fix any prefix $p = (h_1, \ldots, h_{l-1})$ and head h_l. Then

$$\begin{aligned} E R_V^{h_1} \cdots R_V^{h_{l-1}} R_Q^{h_l} &= E R_V^{h_1} \cdots R_V^{h_{l-1}} R_V^{h_l} U_Q \\ &= E R_V^{h_1} \cdots R_V^{h_{l-1}} B_{h_l}^{:d} \\ &= E_{B_{p,h_l}^{:d}} \\ &= M_p W_Q^{h_l}. \end{aligned}$$

where $B_p^{:d}$ denotes the first d columns of B_p. Similarly,

$$E R_V^{h_1} \cdots R_V^{h_{l-1}} R_K^{h_l} = M_p W_K^{h_l}.$$

Thus, the looped analog of (3) holds, implying

$$A^{l,h} = \tilde{A}^{l,h} \quad \text{for all } l, h.$$

Verify OV equalities. For any full path $p = (h_1, \ldots, h_L)$,

$$\begin{aligned} E R_V^{h_1} \cdots R_V^{h_L} U &= E R_V^{h_1} \cdots R_V^{h_{l-1}} B_{h_L} \\ &= E_{B_p} \\ &= M_p = W_V^{h_1} W_O^{h_1} \cdots W_V^{h_L} W_O^{h_L}. \end{aligned}$$

Thus, the looped analog of (4) holds.

By the multi-head unrolling and the equality of attention matrices,

$$Z = \tilde{X}^L U = X^L = Y.$$

□

D Proof of Theorem 3.4 (random looped universal transformer)

Theorem D.1 (Random looped parameters)**.** *Assume*

$$m \;\geq\; M_{\text{loop}} \;:=\; 2d \sum_{t=0}^{L-1} H^t \;+\; d_{\text{in}} H^L.$$

Let $\{R_Q^h, R_K^h\}_{h\in[H]} \in \mathbb{R}^{m\times d}$, $\{R_V^h\}_{h\in[H]} \in \mathbb{R}^{m\times m}$ and U be independent random matrices drawn from absolutely continuous distributions (for example, Gaussian). Then with probability one, for any target transformer $T \in \mathrm{TF}_{H,L,d_{\text{in}},d}$, there exist $E \in \mathbb{R}^{d_{\text{in}}\times m}$ such that

$$M_p W_Q^h = E\, R_p\, R_Q^h, \tag{6}$$

$$M_p W_K^h = E\, R_p\, R_K^h, \tag{7}$$

$$M_p = E\, R_p\, U, \tag{8}$$

for every prefix $p \in [H]^t$ with $0 \leq t \leq L-1$ and every $h \in [H]$ in (6)–(7)*, and for every full path $p \in [H]^L$ in* (8)*. Here*

$$R_p := R_V^{h_1} \cdots R_V^{h_t} \quad \textit{for } p = (h_1, \ldots, h_t), \qquad R_\phi := I_m.$$

Proof. We will show that, for almost every draw of the tied random backbone $\{R_V^h, R_Q^h, R_K^h\}_{h\in[H]}$, there exists at least one U for which the stacked linear system has full column rank. We may then solve for E by pseudoinverse.

Stack all constraints into one linear system. Introduce an auxiliary matrix $U \in \mathbb{R}^{m \times d_{\text{in}}}$, to be chosen later. Define

$$R_{\text{all}}(U) := \Big[\{ R_p R_Q^h \}_{\substack{p \in [H]^t,\ 0 \le t \le L-1 \\ h \in [H]}} \quad \{ R_p R_K^h \}_{\substack{p \in [H]^t,\ 0 \le t \le L-1 \\ h \in [H]}} \quad \{ R_p U \}_{p \in [H]^L} \Big] \in \mathbb{R}^{m \times M_{\text{loop}}},$$

where the blocks are concatenated horizontally in any fixed order.

Likewise define

$$W_{\text{all}} := \Big[\{ M_p W_Q^h \}_{\substack{p \in [H]^t,\ 0 \le t \le L-1 \\ h \in [H]}} \quad \{ M_p W_K^h \}_{\substack{p \in [H]^t,\ 0 \le t \le L-1 \\ h \in [H]}} \quad \{ M_p \}_{p \in [H]^L} \Big] \in \mathbb{R}^{d_{\text{in}} \times M_{\text{loop}}}.$$

Then (6)–(8) are equivalent to

$$E\, R_{\text{all}}(U) = W_{\text{all}}. \tag{9}$$

Therefore, if $R_{\text{all}}(U)$ has full column rank M_{loop}, then

$$E = W_{\text{all}}\, R_{\text{all}}(U)^{\dagger}$$

is a solution of (9).

Reduce full column rank to a nonvanishing polynomial. Let x denote the collection of entries of all tied random matrices

$$\{R_V^h, R_Q^h, R_K^h\}_{h \in [H]}$$

together with the entries of U. Every entry of $R_{\text{all}}(U)$ is a polynomial in x, since each block is a product of matrices of the form $R_p R_Q^h$, $R_p R_K^h$, or $R_p U$.

Let $\mathcal{I}$ be the collection of all M_{loop}-subsets of rows of $[m]$. For each $I \in \mathcal{I}$, let $R_{\text{all}}(U)[I]$ denote the corresponding $M_{\text{loop}} \times M_{\text{loop}}$ submatrix. Define

$$p(x) := \sum_{I \in \mathcal{I}} \det\big(R_{\text{all}}(U)[I]\big)^2.$$

Then $p(x) = 0$ if and only if every $M_{\text{loop}} \times M_{\text{loop}}$ minor vanishes, i.e. if and only if

$$\operatorname{rank}\big(R_{\text{all}}(U)\big) < M_{\text{loop}}.$$

Thus it suffices to show that p is not the zero polynomial.

One tied assignment with full column rank. We use the explicit tied construction from Theorem C.1. In that construction, the coordinates are partitioned into disjoint blocks $\{B_p\}_{p \in \cup_{t=0}^{L} [H]^t}$ of width d_{in}, and the tied routing maps satisfy

$$R_V^h B_p = B_{(p,h)} \qquad \text{for every } |p| < L.$$

Moreover,

$$U = B_{\phi}, \qquad R_Q^h = R_V^h U_Q, \qquad R_K^h = R_V^h U_K,$$

where $U_Q, U_K \in \mathbb{R}^{d_{\text{in}} \times d}$ select the first and last d coordinates of a block, respectively.

For this tied assignment, one has

$$R_p U = B_p \quad \text{for every } p \in [H]^L,$$

and, for every prefix p with $|p| < L$,

$$R_p R_Q^h = B_{(p,h)} U_Q, \qquad R_p R_K^h = B_{(p,h)} U_K.$$

Hence each block-column of $R_{\text{all}}(U)$ is supported on a unique coordinate block: the Q-constraints occupy the first d coordinates of the block $B_{(p,h)}$, the K-constraints occupy the last d coordinates of the block $B_{(p,h)}$, and the OV-constraints occupy the full leaf block B_p.

These supports are pairwise disjoint. Therefore, after a suitable row permutation,

$$P\, R_{\text{all}}(U) = \operatorname{diag}\big(I_d, \ldots, I_d,\ I_d, \ldots, I_d,\ I_{d_{\text{in}}}, \ldots, I_{d_{\text{in}}}\big),$$

where the first group of I_d's corresponds to all Q-constraints, the second group to all K-constraints, and the final group to all $I_{d_{\text{in}}}$'s to the OV-constraints. In particular,

$$\operatorname{rank}\big(R_{\text{all}}(U)\big) = M_{\text{loop}}.$$

Thus p is not the zero polynomial.

Full column rank almost surely. Now draw the tied random backbone

$$\{R_V^h, R_Q^h, R_K^h\}_{h\in[H]}$$

from the distributions in the statement, and independently draw an auxiliary matrix U with an absolutely continuous distribution on $\mathbb{R}^{m\times d_{\text{in}}}$ (for example, Gaussian). Since p is a nonzero polynomial in jointly continuous random variables, its zero set has Lebesgue measure zero. Therefore,

$$\operatorname{rank}\big(R_{\text{all}}(U)\big) = M_{\text{loop}} \qquad \text{with probability one}$$

under the joint draw of the tied random backbone and the auxiliary U.

By Fubini's theorem, for almost every realization of the tied random backbone alone, the set of U for which $\operatorname{rank}(R_{\text{all}}(U)) = M_{\text{loop}}$ has full measure in $\mathbb{R}^{m\times d_{\text{in}}}$; in particular, it is nonempty. Hence, with probability one over the tied random backbone, there exists at least one U such that $R_{\text{all}}(U)$ has full column rank. Fix such a U, and define

$$E = W_{\text{all}}\, R_{\text{all}}(U)^{\dagger}.$$

Then (9) holds, and therefore so do (6)–(8). □

E Lower bound on embedding dimension

We prove a lower bound for our approach even when all the attention matrices are matched.

Assumption E.1. *For each attention head in the random/designed transformer, the attention matrix is equal to that of the target transformer, i.e. $\tilde{A}^{l,h} = A^{l,h}$, and $\forall l \in [L], h \in [H], A^{l,h}$ is arbitrary row-stochastic matrix, i.e. each row of $A^{l,h}$ sum up to one.*

Theorem E.2 (Lower bound assuming matching attention)**.** *Let $d_{\text{in}} = 1$. If there exist matrices $\{R_V^{l,h} \in \mathbb{R}^{m\times m}\}_{l\in[L],h\in[H]}$, for any target transformer with $\{A^{l,h}, W_V^{l,h}, W_O^{l,h}\}_{l\in[L],h\in[H]}$, there exist $E \in \mathbb{R}^{d_{\text{in}}\times m}$ and $U \in \mathbb{R}^{m\times d_{\text{in}}}$ such that for any $X^0 \in \mathbb{R}^{n\times d_{\text{in}}}$, the output $Z = \tilde{X}^L U$ of the constructed transformer equals the target output $Y = X^L$, then $m \geq H^{L/2}$.*

Proof. We prove by contradiction. Recall that

$$X^L = \sum_{(h_1,\ldots,h_L)\in[H]^L} A^{L,h_L}\cdots A^{1,h_1} X^0 W_V^{1,h_1} W_O^{1,h_1}\cdots W_V^{L,h_L} W_O^{L,h_L}$$
$$\tilde{X}^L = \sum_{(h_1,\ldots,h_L)\in[H]^L} \tilde{A}^{L,h_L}\cdots \tilde{A}^{1,h_1} X^0 E R_V^{1,h_1}\cdots R_V^{L,h_L}$$

Assuming the attention matrices are the same, for $\tilde{X}^L U = X^L$,

$$\sum_{(h_1,\ldots,h_L)\in[H]^L} A^{L,h_L}\cdots A^{1,h_1} X^0 \left(\prod_{l=1}^{L} W_V^{l,h_l} W_O^{l,h_l} - E\prod_{l=1}^{L} R_V^{l,h_l} U\right) = 0 \tag{10}$$

Let $\mathbf{h} := (h_1,\ldots,h_L)$ and

$$\alpha_{\mathbf{h}} := \left(\prod_{l=1}^{L} W_V^{l,h_l} W_O^{l,h_l} - E\prod_{l=1}^{L} R_V^{l,h_l} U\right)$$
$$A_{\mathbf{h}} := A^{L,h_L}\cdots A^{1,h_1} = \begin{pmatrix} (A_{\mathbf{h}})_{11} & (A_{\mathbf{h}})_{12} \\ (A_{\mathbf{h}})_{21} & (A_{\mathbf{h}})_{22}\end{pmatrix}$$

Consider the context length $n = 2$ and $X^0 = \begin{pmatrix}1\\0\end{pmatrix}$. Then, $A_{\mathbf{h}} X^0 = \begin{pmatrix}(A_{\mathbf{h}})_{11}\\(A_{\mathbf{h}})_{21}\end{pmatrix}$ and (10) is a linear combination of $\{A_{\mathbf{h}} X^0\}_{\mathbf{h}}$. Specifically, let $\alpha := (\cdots,\alpha_{\mathbf{h}},\cdots)_{\mathbf{h}\in[H]^L} \in \mathbb{R}^{H^L}$ and $v(\{A_{i,h}\}) := (\cdots,(A_{\mathbf{h}})_{11},\cdots)_{\mathbf{h}\in[H]^L} \in \mathbb{R}^{H^L}$, be the vector indexed by $(h_1,\ldots,h_L) \in \mathbb{R}^{H^L}$. The first row of (10) is $v(\{A_{i,h}\})\cdot\alpha$. We will show that if $v(\{A_{i,h}\})\cdot\alpha = 0$ for all sets of attention matrices $\{A^{l.h}\}_{l\in[L],h\in[H]}$, then α must be the 0 vector, by showing $v(\{A_{i,h}\})$ can span the entire $\mathbb{R}^{H^L}$ space.

Lemma E.3. *Let $v(\{A_{i,h}\}) \in \mathbb{R}^{H^L}$ be the vector indexed by $h = (h_1, \dots, h_L) \in [H]^L$, whose h-th coordinate is*

$$v(\{A_{i,h}\})_{h_1,\dots,h_L} := \big(A_{1,h_1} A_{2,h_2} \cdots A_{L,h_L}\big)_{1,1},$$

where each $A_{i,h}$ is a row-stochastic matrix. Then, the span of all such vectors is all of $\mathbb{R}^{H^L}$.

Proof. It suffices to show that every standard basis vector in $\mathbb{R}^{H^L}$ can be realized. Fix any sequence

$$s = (s_1, \dots, s_L) \in [H]^L.$$

We construct 2×2 row-stochastic matrices $\{A^{(s)}_{i,h}\}_{i\in[L],\, h\in[H]}$ such that the resulting vector $v^{(s)}$ satisfies

$$v^{(s)}_{h_1,\dots,h_L} = \mathbf{1}\big[(h_1, \dots, h_L) = s\big].$$

This will imply that $v^{(s)}$ is exactly the standard basis vector e_s.

Consider the two 2×2 row-stochastic matrices

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}.$$

For the fixed sequence s, define

$$A^{(s)}_{i,h} = \begin{cases} I, & h = s_i, \\ B, & h \neq s_i. \end{cases}$$

Now let $h = (h_1, \dots, h_L) \in [H]^L$. If $h = s$, then every factor equals I, and hence

$$A^{(s)}_{1,h_1} A^{(s)}_{2,h_2} \cdots A^{(s)}_{L,h_L} = I,$$

so

$$v^{(s)}_{h_1,\dots,h_L} = \big(A^{(s)}_{1,h_1} A^{(s)}_{2,h_2} \cdots A^{(s)}_{L,h_L}\big)_{1,1} = I_{1,1} = 1.$$

On the other hand, if $h \neq s$, then there exists some $i \in [L]$ such that $h_i \neq s_i$, so the product contains at least one factor equal to B. Observe that

$$IB = B, \qquad BI = B, \qquad BB = B.$$

Therefore, once a factor B appears in the product, the entire product is equal to B. Consequently,

$$v^{(s)}_{h_1,\dots,h_L} = \big(A^{(s)}_{1,h_1} A^{(s)}_{2,h_2} \cdots A^{(s)}_{L,h_L}\big)_{1,1} = B_{1,1} = 0.$$

Thus,

$$v^{(s)}_{h_1,\dots,h_L} = \mathbf{1}\big[(h_1, \dots, h_L) = s\big],$$

so indeed $v^{(s)} = e_s$.

Since $s \in [H]^L$ was arbitrary, every standard basis vector e_s belongs to the set of realizable vectors. Hence these vectors span all of $\mathbb{R}^{H^L}$, and the span has full dimension H^L. □

If there exist $\alpha_{\mathbf{h}} \neq 0$, then picking the $A^{(\mathbf{h})}_{i,h}$ results in $v(\{A_{i,h}\}) \cdot \alpha \neq 0$. Therefore, each term $\alpha_{\mathbf{h}} = 0$ for all $\mathbf{h} \in [H]^L$. Next, we show that if $m < H^{L/2}$, then α can not be 0.

Lemma E.4. *If $m < H^{L/2}$, then for each collection of $m \times m$ matrices $\{R^{l,h}\}_{(l,h)\in[L]\times[H]}$, there exists a collection of $d_{\text{in}} \times d_{\text{in}}$ matrices $\{W^{l,h}\}_{(l,h)\in[L]\times[H]}$ such that, for each pair of $d_{\text{in}} \times m$ matrices E, U, there exists $(h_1, \dots, h_L) \in [H]^L$ such that*

$$E \prod_{l=1}^{L} R^{l,h_l} U^{\top} \neq \prod_{l=1}^{L} W^{l,h_l}.$$

Proof. Fix a collection of $m \times m$ matrices $\{R^{l,h}\}_{(l,h)\in[L]\times[H]}$. Consider the system of H^L linear equations $Ax = b$ in m^2 unknowns x, where the equation corresponding to $(h_1, \ldots, h_L) \in [H]^L$ is

$$\operatorname{tr}\left(\prod_{l=1}^{L} R^{l,h_l} \operatorname{mat}(x)\right) = \prod_{l=1}^{L} W^{l,h_l}.$$

Here, $\operatorname{mat}(x)$ reshapes x to a $m \times m$ matrix. The matrix A is determined by $\{R^{l,h}\}_{(l,h)\in[L]\times[H]}$; below, we show how to choose $\{W^{l,h}\}_{(l,h)\in[L]\times[H]}$, which in turn determines the right-hand side vector b.

The rank of A is at most the number of columns, which is m^2; that number is at most $H^L - 1$ by the assumption $m < H^{L/2}$. So by the rank-nullity theorem, the left nullspace of A has dimension at least 1. Let y be a nonzero vector in the left nullspace of A, and let $y_{h_1^*,\ldots,h_L^*}$ be a nonzero component of y. For each $(l, h) \in [L] \times [H]$, choose

$$W^{l,h} := \begin{cases} 1 & \text{if } h = h_l^*; \\ 0 & \text{otherwise;} \end{cases}$$

and let b be the corresponding right-hand side vector. We have

$$\langle y, b\rangle = \sum_{(h_1,\ldots,h_L)\in[H]^L} y_{h_1,\ldots,h_L} \times \prod_{l=1}^{L} W^{l,h_l} = y_{h_1^*,\ldots,h_L^*} \times 1 \neq 0.$$

Therefore, b has a nonzero component in this left nullspace, which implies that b is not in the column space of A, i.e., $Ax = b$ has no solution. Since $U^\top E$ is a $m \times m$ matrix, it follows that there exists some $(h_1, \ldots, h_L) \in [H]^L$ such that

$$E \prod_{l=1}^{L} R^{l,h_l} U^\top = \operatorname{tr}\left(\prod_{l=1}^{L} R^{l,h_l} U^\top E\right) \neq \prod_{l=1}^{L} W^{l,h_l}$$

□

The proof of Theorem E.2 follows from this. □

The lower bound also works for the weight-tying (looped) transformer, by choosing

$$W^{h} := \begin{cases} 1 & \text{if } h = h_l^*; \\ 0 & \text{otherwise.} \end{cases}$$

F Additional ablation studies

Besides parenthesis balancing and 2-hop induction heads, we also perform ablation studies on 1-hop, 3-hop, and 4-hop induction heads tasks. These experiments follow the same protocol as in the parenthesis balancing and 2-hop induction heads: for the sparse and random universal transformers, only the embedding and unembedding matrices are trained, while all internal attention parameters are kept fixed. When MLP layers are added, they are randomly initialized and also kept fixed. The results are shown in Tables 3 and 4. These results are consistent with Table 2b, that while it's hard for the attention-only universal transformers to find the optimal solution, adding residual connections and layer normalization can make the optimization much easier.

G Experimental details

G.1 Data generation

Parenthesis balancing. The data is generated following the same way as [28]. We generate sequences over a four-symbol vocabulary consisting of padding/EOF, left parenthesis, right parenthesis, and a query token, encoded as $0, 1, 2, 3$, respectively. Each example is constructed by first sampling a parenthesis string of length at most $2\,\texttt{max_len}$. With probability $1/3$, we sample a completely

Table 3: Induction head (1-hop) results for $H = 2$, $L = 2$, head_dim $= 28$, embedding_dim $= 456$.

Model	Sparse	Random	Fully-trained
Attention-only	30.2%	29.4%	8.4%
+ residual	42.1%	5.5%	67.5%
+ residual and LN	100%	96.1%	100%
+ MLP	61.0%	51.3%	100%
+ MLP and LN	100%	96.0%	100%

Table 4: 3-hop and 4-hop results for $H = 2$, $L = 4$, head_dim $= 30$, embedding_dim $= 2280$.

3-hop

Model	Sparse	Random	Fully-trained
Attention-only	39.2%	32.6%	99.5%
+ residual	53.6%	43.9%	99.9%
+ residual and LN	100%	92.6%	100%
+ MLP	56.9%	57.5%	100%
+ MLP and LN	100%	98.7%	100%

4-hop

Model	Sparse	Random	Fully-trained
Attention-only	39.4%	35.2%	95.6%
+ residual	43.9%	41.6%	97.9%
+ residual and LN	100%	89.2%	100%
+ MLP	44.7%	48.8%	97.4%
+ MLP and LN	82.8%	70.9%	100%

random parenthesis string of uniformly random length; otherwise, we recursively generate a balanced parenthesis string with a uniformly sampled number of matching pairs. To obtain both positive and negative instances, the sampled string is optionally corrupted by local mutations. These mutations include swapping two positions in the string and flipping parentheses from left to right or right to left, with a geometrically distributed number of repetitions. After the parenthesis string s is generated, we compute whether it is balanced using the standard stack-depth criterion: the running depth must never become negative and must end at zero. We then append a query token "?" followed by the target answer token. The answer is ")" if s is balanced and "(" otherwise. Finally, the sequence is padded with zeros to a fixed length $2\,\texttt{max_len} + 3$. The model is trained in a next-token prediction format: for each padded sequence, the input is the sequence excluding the final token, and the target is the same sequence shifted one position to the left.

k-hop induction heads. For this task, we generate the data following [20]. We generate sequence-modeling examples over a vocabulary consisting of character tokens, induction-query tokens, a blank token, and a special "does-not-exist" token. The character vocabulary contains `char_tokens` symbols, denoted by consecutive lowercase letters. For each example, we first sample a random character string $x_1, \ldots, x_{\texttt{seq_len}}$, where the first character is sampled uniformly and each subsequent character is sampled uniformly subject to being different from the previous character. This prevents trivial adjacent repetitions. We then construct a sequence of induction-hop targets. The 0-hop string is the original string itself. Given the current hop string and its associated source indices, the next-hop string is obtained positionwise as follows: for position i, we look backward in the original random string for the most recent previous occurrence of the current symbol before its current source index. If such an occurrence exists, the next-hop target is the character immediately following that occurrence; otherwise, the target is the special "does-not-exist" symbol. Repeating this procedure produces targets for all hop depths up to `max_hops`. For each training example, the queried hop depth is either fixed or sampled uniformly from $\{\texttt{min_hops}, \ldots, \texttt{max_hops}\}$ and encoded as a leading induction-query token. The model input consists of this query token followed by the random character string with the last two characters removed. The output is a blank token followed by the corresponding hop-target string, also truncated by two characters. Thus the task is trained in a full-sequence prediction format:

after reading the hop query and the character context, the model must output, at every position, the result of applying the specified number of induction steps to the underlying sequence.

G.2 Training details

All the experiments are conducted on one single NVIDIA RTX PRO 6000 Blackwell 96GB. This project roughly takes 14 GPU days.

The embedding and unembedding matrices are initialized from standard normal distribution with standard deviation 0.02. We use AdamW as optimizer and LambdaLR as learning rate scheduler. We adopt online training, i.e. at each step we generate fresh data.

The hyperparameters for parenthesis balancing and k-hop induction heads task can be found in Tables 5 and 6. For $k = 1, 2, 3, 4$, the training steps are $10000, 15000, 60000, 60000$ respectively.

Hyperparameter	**Value**
Embedding dimension m	1024
Head dimension d	24
Depth L	2
Number of heads H	4
Vocabulary size	4
Activation function of MLP	GeLU
Layer norm ϵ	10^{-5}
Learning rate	10^{-3}
Warmup steps	50
Training steps	10^4
Batch size	1000

Table 5: Model and training hyperparameters for parenthesis balancing.

Hyperparameter	**Value**
Number of heads H	2
Number of character tokens	4
Vocabulary size	30
Activation function of MLP	GeLU
Layer norm ϵ	10^{-5}
Learning rate	10^{-4}
Warmup steps	1000
Batch size	128

Table 6: Model and training hyperparameters for k-hop induction heads.