FIVE INVARIANTS GOVERN ATTENTION EXPRESSIVITY: A COMPLETENESS THEOREM

Anonymous authors
Paper under double-blind review

ABSTRACT

Dozens of impossibility and separation theorems now record what one attention variant can do and another cannot, each proved on its own terms. We ask whether they share a structure, and prove that on the canonical tasks they do. We identify five structural invariants of a single-head attention architecture, namely kernel-rank, piecewise-degree, boundedness, symmetry, and recurrent-state, and prove a completeness theorem about them. Each invariant is an obstruction on its own class of tasks. On the associative-recall, averaging, counting, and indexing tasks the five are also jointly sufficient. For the realization regimes covered by the theorem, every cost gap beyond polylogarithmic factors on these tasks has an invariant witness. Under a matrix-dimension cost measure, budgeted invariant profiles determine each canonical task's cost up to polylogarithmic factors. We tabulate that cost for softmax, hard, average-hard, polynomial, sigmoid, causal, and position-free attention, as well as state-space architectures, on every canonical task. Two consequences stand out. Single-layer linear attention needs width at least N on associative recall over N keys, at any head count, against $O(\log N)$ for softmax. The single-layer polynomial kernels form a strict hierarchy in the degree.

1 INTRODUCTION

Softmax, polynomial-linear, hard, average-hard, sigmoid, sliding-window and position-free (NoPE) attention, along with linear and selective state-space (SSM) models, make different trade-offs between efficiency and expressivity. Section 2 places their score functionals, embeddings, masks and recurrences in a common framework. A separate literature of impossibility and separation theorems has developed around these models (Hahn, 2020; Sanford et al., 2023; 2024a;b; Arora et al., 2024b; Merrill & Sabharwal, 2023a; Chen et al., 2025), often under different model definitions and resource budgets. We seek a small set of structural axes that places these separations in a common framework.

A single theorem organizes these separations through five structural invariants of single-head attention architectures: kernel-rank, piecewise-degree, boundedness, symmetry, and recurrent-state. We prove that each is an obstruction on its own class of tasks, and that on the canonical tasks of associative recall, averaging, counting and indexing the five are jointly sufficient. For the realization regimes covered by the theorem, every cost gap beyond polylogarithmic factors between the architectures of Table 1 on those tasks has an invariant witness. Each invariant captures a distinct failure mode in the cited separations. Below, d is the width of a model, L its depth, n the length of its input, and N the number of keys in associative recall.

- Kernel-rank (KRank): a polynomial score functional of degree p factors its kernel through a feature space of dimension $\binom{d+p}{p}$. This forces width $\Omega(N^{1/p})$ on associative recall.
- Piecewise-degree (PDeg): the argmax of hard attention partitions input space into finitely many regions, and on each region the depth-L output depends on at most 2^L positions. At depth below $\log_2 n$, this blocks the exact realization of tasks that depend on every position.
- Boundedness (Bound): a normalized layer outputs a convex combination of its inputs, so no fixed depth gives a range growing faster than a polylogarithm of the length. This blocks length-uniform counting, while an unnormalized layer counts at depth one.

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

- Symmetry (Sym): a permutation-equivariant architecture (NoPE), one whose outputs are permuted along with its input positions, cannot break input symmetry. This blocks position-sensitive tasks.
- Recurrent-state (State): a recurrence with state dimension d_h carries at most d_h independent values to the query. This blocks streaming retrieval tasks of higher rank.

1.1 Our Results

The paper proves one theorem, stated here in full. The five invariants are defined in Section 3, and we write $\mathcal{I}$ for any one of them. The class $\mathcal{T}_{\mathcal{I}}$ of tasks on which the invariant $\mathcal{I}$ is an obstruction is defined in Theorem 4.1. The cost C of a model (Theorem 2.2) is a matrix-dimension proxy. Fix an architecture A and a cost budget C. The value $\mathcal{I}(A; C)$ is the best value of the invariant $\mathcal{I}$ over the widths, depths, head counts and state dimensions that A can take at that cost or less (Theorem 3.2). Here *best* means the most capacity, in the orientation of Theorem D.3. The value $\mathcal{I}(T)$ measures how much of the invariant $\mathcal{I}$ a task T requires (Section 3). The canonical tasks AR_N, Avg_n, Count_n and Index_n of part (ii) are associative recall, averaging, counting and indexing (Theorem 2.3). The notation $\tilde{O}(\cdot)$ hides polylogarithmic factors.

Theorem 1.1 (Completeness theorem, restated as Theorem 4.2)**.** *Let A be a standard architecture (Theorem 2.4), with a fixed embedding as in Theorem 2.4(ii).*

(i) Soundness. *Fix one of the five invariants $\mathcal{I} \in \{\mathrm{KRank}, \mathrm{PDeg}, \mathrm{Bound}, \mathrm{Sym}, \mathrm{State}\}$ and a task T in its class $\mathcal{T}_{\mathcal{I}}$. Suppose that a model of A of depth L realizes T to error $\epsilon < 1/2$ at cost C. Then $\mathcal{I}(A; C) \geq \mathcal{I}(T)$, in the orientation of Theorem D.3. On the* KRank *axis this holds at $L = 1$ unconditionally, and at $L > 1$ for realizations that are layered in the sense of Theorem A.8, provided that the single layer extracted by the lemma reads a split embedding. On the* PDeg *axis it holds for exact realization. On the* Bound *and* Sym *axes it holds at every depth. On the* State *axis it holds for the single recurrence of Theorem 2.4(iv).*

(ii) Sufficiency on the canonical tasks. *Let T be one of the canonical tasks AR_N, Avg_n, Count_n and Index_n. Let A be a row of Table 1 whose mask lets the read position (Theorem 2.3) see every input position, and let C be a cost budget. Suppose that $\mathcal{I}(A; C) \geq \mathcal{I}(T)$ for all five invariants. Then A realizes T at cost $\tilde{O}(C)$. The realization is exact, except in the cells that Theorem D.20 names, where it is to every error $\epsilon \geq e^{-\mathrm{polylog}(n)}$.*

(iii) Witness. *Consequently, let T be such a task, and let A_1 and A_2 be two such rows that share a mask. Suppose that A_1 realizes T at cost C_1, by a realization to which part (i) applies on every axis, while A_2 requires cost $\omega(\tilde{O}(C_1))$ at the tolerance of part (ii). Then, for all large n, some invariant satisfies $\mathcal{I}(A_2; C_1) < \mathcal{I}(T) \leq \mathcal{I}(A_1; C_1)$.*

Four consequences follow. First, the least cost at which all five inequalities of part (ii) hold is the task's cost up to polylogarithmic factors (Theorem B.1, Table 2). Second, position-free softmax realizes precisely the permutation-invariant canonical tasks (Theorem 4.3). Third, single-layer linear attention needs width $\Omega(N)$ on AR_N at every head count (Theorem B.2). Fourth, single-layer polynomial kernels form a strict hierarchy in the degree (Theorem B.3).

1.2 Our Techniques

The proof rests on four elementary ingredients, all proved here rather than imported. The first is the bilinear kernel factorization (Theorem A.2), which caps the rank of a degree-p attention kernel by the dimension $\binom{d+p}{p}$ of its feature space. The second is the influence-set bound (Theorem A.6). It concerns a κ-selective functional, one that places every nonzero attention weight of a query on at most κ positions within each of finitely many regions of the input space (Theorem A.5). The bound shows that the output of such a functional at depth L depends on at most $(\kappa + 1)^L$ input positions within a region. The third is the linear-recurrence rank bound (Theorem A.7), which forces a state dimension at least the rank of a streaming retrieval task. The fourth is compositional inheritance (Theorem A.8), which lifts single-layer bounds through a layered realization. In a layered realization some layer receives each token through a fixed function of that token alone, and the output of

the model depends only on that layer's output at the read position. The literature already has the dimension count behind the first ingredient, as well as the separations that these ingredients explain (Section D). The contribution is their assembly, and Section E gives each ingredient in full.

1.3 Other Related Work

Our invariants assemble obstructions previously studied in isolation. Hahn (2020) proved the first impossibility results for hard attention, by a restriction argument that is the ancestor of the influence-set reasoning behind PDeg. Sanford, Hsu, and Telgarsky (2023) proved precision-sensitive size lower bounds for one-layer attention by communication complexity, which is the style the analytic branch of KRank uses. Arora et al. (2024b) proved, for every causally dependent recurrent model, the $\Omega(N)$-bit state bound that stands behind State. A parallel program develops logical and algebraic bounds and characterizations for specified transformer models (Chiang et al., 2023; Merrill & Sabharwal, 2023b; Yang et al., 2024; Li & Cotterell, 2025; 2026; Nowak et al., 2026; Sälzer et al., 2026; Svete et al., 2026). Those results characterize which functions an architecture computes; our theorem identifies structural properties that determine the cost of computing the canonical tasks. The dense-associative-memory line reached the shape of our KRank dichotomy between polynomial and analytic kernels first, in a different model (Krotov & Hopfield, 2016; Demircigil et al., 2017; Millidge et al., 2022; Ramsauer et al., 2021), and Section D says what remains ours. Section F places the paper against the rest of the field.

2 Preliminaries

We work over the reals and write $[n] = \{1, \dots, n\}$ for the set of the first n positive integers.

2.1 The Single-Head Attention Layer

A layer reads a sequence of n vectors, one at each position, and we call n the length of the sequence. Fix a width $d \in \mathbb{N}$, the dimension of the vector at each position. An input to a layer is then a sequence $X = (x_1, \dots, x_n)$ in $(\mathbb{R}^d)^n$, with the vector x_i at position i. A single-head attention layer with score functional σ and visibility mask $\mathcal{V} \subseteq [n] \times [n]$ is the map $\mathrm{Att}_\sigma : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n$ given by

$$\mathrm{Att}_\sigma(X)_i = \frac{1}{Z_i(X)} \sum_{j:(i,j)\in\mathcal{V}} \sigma(q_i, k_j)(W_V x_j), \qquad q_i = W_Q x_i, \quad k_j = W_K x_j, \tag{1}$$

with parameters $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ and a normalizer Z_i determined by σ. In words, position i forms a query vector q_i, and every position j forms a key vector k_j and a value vector $W_V x_j$. The score $\sigma(q_i, k_j)$ weighs position j for the query at position i. The output at position i is the sum of the value vectors of the positions that i may see, each weighted by its score, divided by the normalizer Z_i. The mask says which positions each position may see: position i attends to position j exactly when the pair (i, j) lies in $\mathcal{V}$. The mask is a fixed function of the position pair, not of the input.

We name three masks. The bidirectional mask $\mathcal{V} = [n] \times [n]$ lets every position see every position. The causal mask $\mathcal{V} = \{(i, j) : j \le i\}$ lets a position see itself and the positions before it. The sliding-window mask $\mathcal{V} = \{(i, j) : |i - j| \le w\}$ of half-width w lets a position see the positions within distance w of it. Unless a statement says otherwise the mask is bidirectional, and we write $\sum_{j=1}^{n}$ for the sum in Eq. (1). The mask carries the permutation structure that the symmetry invariant Sym of Section 3 reads. For that reason every comparison in this paper is made at a common mask.

The score functionals we consider are the following six. Softmax attention has the score $\sigma(q, k) = \exp\langle q, k\rangle$, and its normalizer is the sum of the scores of the visible positions, $Z_i = \sum_{j':(i,j')\in\mathcal{V}} \exp\langle q_i, k_{j'}\rangle$. Hard attention puts all the weight on one visible position, the one whose key vector has the largest inner product with the query vector. Its score is $\sigma(q_i, k_j) = \mathbf{1}[j = \mathrm{argmax}_{j':(i,j')\in\mathcal{V}}\langle q_i, k_{j'}\rangle]$, where $\mathbf{1}[\cdot]$ is 1 when the bracketed event holds and 0 otherwise, and its normalizer is $Z_i = 1$. Ties in the argmax are broken by a rule that depends only on the tied set and the query position. Average-hard attention spreads the weight uniformly over the argmax set, the set of visible positions at which that inner product is largest. Unlike the other scores, those of hard and average-hard attention depend on every visible key. Polynomial-linear attention of degree p has

the score $\sigma(q,k) = \langle \phi(q), \phi(k) \rangle$ and the normalizer $Z_i = 1$, for a feature map ϕ whose coordinates are polynomials of degree at most p. The feature map ϕ is chosen with the other parameters of the model. Sigmoid attention has the score $\sigma(q,k) = \mathrm{sigmoid}(\langle q, k \rangle)$ and the normalizer $Z_i = 1$. Sliding-window attention is softmax attention with the window mask of half-width w.

Softmax, hard and average-hard attention are the normalized functionals, in that their weights are nonnegative and sum to one. Polynomial-linear and sigmoid attention are unnormalized. A normalized functional is nonnegative, so that its weights form a distribution. An unnormalized one may take either sign, as polynomial-linear attention does. Normalization is the only property of the score functional that the boundedness invariant Bound of Section 3 reads, and it leaves the kernel rank unchanged. The reason is that dividing row i of the score matrix, the matrix of the scores $\sigma(q_i, k_j)$, by Z_i is a diagonal scaling. So normalization changes the output range but not the rank of the kernel.

A model is a tuple $\mathcal{M} = (\iota, \{\mathrm{Att}_\sigma^{(\ell)}\}_{\ell=1}^{L}, r)$ of three parts. The first is an input embedding ι, which maps each symbolic token to a vector in $\mathbb{R}^d$. The second is a stack of L attention layers $\mathrm{Att}_\sigma^{(1)}, \ldots, \mathrm{Att}_\sigma^{(L)}$ with or without residual connections. The third is a token-wise readout r, a map applied at each position separately. We elide token-wise feedforward sublayers to keep the analysis focused on attention. The width of the model is d and its depth is L.

Every parameter matrix, every embedded token and every readout in this paper is of bounded magnitude, in the following three senses, which together we call the norm convention. First, the matrices W_Q, W_K, W_V and any multi-head output maps have operator norm at most an absolute constant c_V. Second, the embedded tokens have norm at most $P(n)$, where P is a polylogarithmic function of the length. Third, the readout's value on the vectors the model produces is at most an absolute constant times their largest coordinate, plus an offset of at most $P(n)$. The same P and constants are fixed for all models before an invariant supremum is taken or two architectures are compared; they may depend on a fixed error-tolerance regime. The output range of a layer is read under the norm convention, and Theorem D.1 says why the Bound axis needs it. Every other axis is invariant under rescaling, and every construction in the paper satisfies the norm convention.

2.2 Realization and Parameter Cost

A model realizes a function when its output is close to the value of the function at every input, and the next definition makes that precise.

Definition 2.1 (Realization)**.** Let $\mathcal{M}$ be a model, $\mathcal{X}$ an input domain, f a function on it, and $\epsilon \geq 0$. The model realizes f on the domain to error ϵ if $\|\mathcal{M}(x)_\rho - f(x)\|_\infty \leq \epsilon$ for every input $x \in \mathcal{X}$. Here $\mathcal{M}(x)_\rho$ is the output at the read position ρ of Theorem 2.3. Exact realization is the case $\epsilon = 0$.

In words, the output of the model at the read position differs from the value of f by at most ϵ in every coordinate, at every input of the domain. Unless a statement says otherwise, lower bounds are for realization to error $\epsilon < 1/2$, which includes exact realization. Upper bounds are exact where the score functional allows it and ϵ-approximate otherwise. Theorem D.5 says which bounds are of which kind, and why.

The separations compare architectures by the cost proxy below. The state dimension d_h that it uses is the dimension of the hidden state of a recurrent architecture, Theorem 2.4(iv).

Definition 2.2 (Cost proxy)**.** The cost of a model with width d, depth L, and state dimension d_h, where $d_h = 0$ for a feedforward model, is

$$C = L \cdot (d^2 + d_h^2), \tag{2}$$

a matrix-dimension cost proxy that excludes embeddings, readouts, feature-map coefficients and input-dependent map coefficients. A model of an architecture A (Theorem 2.4) is one built from the components of A at some width, depth, head count and state dimension. The architecture A realizes a task T at cost C when some model of A with cost at most C realizes the task.

Realization is per length. A model realizing T at length n may have its embedding, its parameters and its readout depend on n, within the norm convention above. Per-length dependence within the norm convention is what lets a token-wise readout divide by n and forbids it to multiply by n. Length-uniform realization, by one model at every length, is strictly stronger and is named wherever it is meant. The Bound axis separates in that form.

2.3 TASKS

The completeness theorem is proved on four tasks: two retrieve a single value from the sequence, and two aggregate all of its values.

Definition 2.3 (Canonical tasks)**.** We use the following tasks.

AR_N **(associative recall).** The input is a sequence of length $N+1$. Each position $j \in [N]$ carries a key k_j and a payload $v_j \in [-1,1]$, and the keys $k_1, \ldots, k_N$ are a permutation of $[N]$. Position $N+1$ carries a query $q \in [N]$. The output is the payload of the position whose key is q.

Avg_n **(averaging).** Position $j \in [n]$ carries a scalar $v_j \in [-1,1]$. The output is the mean $\bar{v} = \frac{1}{n}\sum_{j=1}^{n} v_j$.

Count_n **(counting).** Position j carries $v_j \in [0,1]$. The output is the sum $\sum_{j=1}^{n} v_j$, which is the count of ones when $v \in \{0,1\}^n$.

Index_n **(indexing).** Position $j \in [n]$ carries $v_j \in [-1,1]$, and position $n+1$ carries a query $i^* \in [n]$. The output is the value v_{i^*} at the queried position.

The output is read at the query position where there is one and at position n otherwise, and we call that position the read position.

These four tasks are the canonical tasks of the completeness theorem. Each is the running example of at least one invariant's class of tasks (Theorem 4.1). Associative recall is the running example for KRank, averaging for PDeg, counting for Bound, and indexing for Sym. The running example for State is again AR_N, now read as a streaming task processed by a recurrence that meets the query last. Every coefficient vector of AR_N, the weights with which its target combines the values (Theorem A.1), is supported on the N key-value positions before the query. This support is the condition that the class $\mathcal{T}_{\mathrm{State}}$ of Theorem 4.1 asks for. In AR_N the key k_j and the query q are symbols in $[N]$, not the key vectors and query vectors of Eq. (1).

In both retrieval tasks, AR_N and Index_n, a key-value position is embedded by the split embedding, which gives its value a channel of its own. The split embedding is $\iota(k_j, v_j) = u_j + v_j w$, where the key part u_j is a fixed function of the key and the value direction w is fixed. The query and key maps of every model read the key channel alone, that is, $W_Q w = W_K w = 0$. The transition and input maps of a recurrence, defined in Theorem 2.4(iv), are likewise fixed or functions of the key channel alone. Theorem D.7 says what the split gives and what is known without it.

2.4 THE STANDARD ARCHITECTURE CLASS

The completeness theorem is stated for the following class of architectures, and every row of Table 1 belongs to it.

Definition 2.4 (Standard architecture)**.** A standard architecture A is specified by five components.

(i) For feedforward attention, a score functional σ_A, nonnegative where normalized, that is either real-analytic in its arguments or κ-selective in the sense of Theorem A.5; recurrent rows have no score functional. Hard attention is the case $\kappa = 1$.

(ii) An embedding map ι_A, chosen with the other parameters of a model, that depends only on token content and position. It is real-analytic in the real coordinates of the content, on all of their Euclidean space and not only on a task's domain.

(iii) A class $\mathcal{R}_A$ of token-wise readouts, linear or feedforward of width $O(d)$, within the norm convention of Section 2.1.

(iv) A stacking discipline. It is either feedforward, a stack of depth L with optional residual connections, or a single recurrent layer $h_t = R_t h_{t-1} + E_t \iota_t$ of state dimension d_h, for which $L = 1$. In the recurrent layer the vector $h_t \in \mathbb{R}^{d_h}$ is the hidden state after position t, and ι_t is the embedded token there. The readout r at position t reads h_t together with

the embedded token ι_t. The matrix R_t is the transition map, and the matrix E_t is the input map.

(v) A visibility mask $\mathcal{V}_A$, a fixed function of the position pair, applied in every attention layer; recurrent rows have causal visibility.

A κ-selective functional, by Theorem A.5, places every nonzero attention weight of a query on at most κ positions within each of finitely many regions that cover its inputs. Condition (ii) means that the embedding of a token never reads the other tokens of the sequence. A multi-head variant runs H heads of width d/H, maps each head output to $\mathbb{R}^d$ by $O_h \in \mathbb{R}^{d\times(d/H)}$, and sums those outputs (Section G).

3 The Five Structural Invariants

This section defines five structural invariants of an architecture and, for each of them, a matching quantity for a task. The score functional supplies the kernel rank and the piecewise degree, and the normalization supplies the output range. The embedding together with the mask supplies the symmetry group, and the stacking discipline supplies the state dimension (Theorem D.19). The completeness theorem of Section 4 compares the invariants of an architecture with the matching quantities of a task. To each standard architecture A of Theorem 2.4, at given parameters, we associate the following five invariants.

Definition 3.1 (Five structural invariants)**.**

- The kernel rank $\mathrm{KRank}(A) \in \mathbb{N} \cup \{\infty\}$ is the least dimension of a feature space through which the attention kernel of every model of A factors. The attention kernel is the score functional $\sigma(q, k)$, a function of a query and a key. It factors through a feature space when it is the inner product $\langle \phi(q), \phi'(k)\rangle$ of feature vectors of the query and of the key in that space, as in Theorem A.2. Polynomial-linear attention of degree p has $\mathrm{KRank}(A) = \binom{d+p}{p}$, the number of monomials of degree at most p in d variables. Analytic super-polynomial kernels such as softmax and sigmoid have $\mathrm{KRank} = \infty$. For hard, average-hard and recurrent rows, we set $\mathrm{KRank} = \infty$ by convention. The kernel rank caps the rank of the kernel matrix of Theorem A.2, whose entries are the scores between a family of queries and the keys, divided by the normalizer. It is the invariant that limits retrieval: a polynomial kernel of degree p forces width $\Omega(N^{1/p})$ on associative recall (Theorem A.3).

- The piecewise degree $\mathrm{PDeg}(A, L) \in \mathbb{N}\cup\{\infty\}$ counts how many input positions the output at the read position can depend on, piece by piece. To define it, regard the input, for each value of its discrete content, as a point of $\mathbb{R}^m$, where m is the number of its real coordinates. The input domain is then a subset of $\mathbb{R}^m$. For a model of A at depth L, consider the least k with the following property. This domain is covered by finitely many semi-analytic regions (Theorem A.5), on each of which the output at the read position is a function of at most k input positions. The piecewise degree $\mathrm{PDeg}(A, L)$ is the largest value of this k over the models of A at depth L.

  Single-head hard attention has $\mathrm{PDeg}(A, L) = 2^L$. Sliding-window attention of half-width w, read at the last position, has $\mathrm{PDeg}(A, L) = Lw + 1$. Analytic-kernel architectures read at a position that sees every position have $\mathrm{PDeg} = \infty$. Every value is at most the length n, so 2^L stands for $\min(2^L, n)$, and ∞ stands for n at every depth. Selectivity (Theorem A.5) bounds how many positions a query attends to: on each piece of a finite cover, the nonzero attention weights of a κ-selective functional at each query lie on at most κ positions. For such a functional, Theorem A.6 gives the general bound $(\kappa + 1)^L$. The piecewise degree is the invariant that limits tasks whose target depends on every position, such as Avg_n. By Theorem 4.2(i), an exact realization of a task in the matching class of Theorem 4.1 needs a piecewise degree at least the task's own, which is defined below.

- The output range $\mathrm{Bound}(A, L) \in \mathbb{R} \cup \{\infty\}$ is the supremum of the output magnitude over the models of A at depth L and the inputs on which they satisfy the norm convention of Section 2.1. It measures how large an output the architecture can produce from embedded tokens of bounded norm. A normalized layer multiplies the largest token norm by at most

$1 + Hc_V^2$, where H is the number of heads and c_V bounds the parameter matrices and head output maps. Softmax, hard, average-hard, sliding-window and NoPE softmax attention are normalized. A depth-L normalized architecture therefore has $\mathrm{Bound}(A, L) = 2^{\Theta(L)}$ at bounded head count, up to the polylogarithmic factor $P(n)$ of the norm convention. An unnormalized layer that gives every position the same score outputs a sum of n values. An unnormalized architecture, such as polynomial-linear attention, sigmoid attention or an accumulating recurrence, therefore has $\mathrm{Bound}(A, 1) \geq n$. Table 1 records this value as ∞, to say that the output range is unbounded in the length at fixed depth. At a fixed depth and head count, a normalized architecture cannot reach a range that grows faster than a polylogarithm of the length. Length-uniform realization of Count_n, by one model at every length, asks for such a range. The output range is therefore the invariant that limits length-uniform counting.

- The symmetry group $\mathrm{Sym}(A)$ is the largest group of permutations of the positions other than the read position under which A is equivariant. Equivariance under a permutation π means that permuting the input positions by π permutes the output positions by π as well. We write S_n for the group of all these permutations, and $\{e\}$ for the trivial group, which contains the identity alone. Architectures with a positional encoding, whether absolute, rotary or additive relative, have $\mathrm{Sym}(A) = \{e\}$. NoPE, the architecture with no positional encoding, has $\mathrm{Sym}(A) = S_n$ under a bidirectional mask. Equivariance is a property of the embedding *and* the mask together. A causal or windowed mask is not permutation-invariant, so a causally masked architecture has $\mathrm{Sym}(A) = \{e\}$ whether or not it carries positional encodings. For this reason the Sym column of Table 1 is read at a fixed mask. A larger group is a stronger constraint on the architecture, so S_n is the least capable value and $\{e\}$ the most capable one (Theorem D.3). The symmetry group is the invariant that limits position-sensitive tasks, such as Index_n, whose target changes when two value positions are exchanged.

- The state dimension $\mathrm{State}(A) \in \mathbb{N} \cup \{\infty\}$ is the hidden-state dimension in the recurrent setting, and ∞ for feedforward attention. In the recurrent setting it is the dimension d_h of the state h_t that the recurrence $h_t = R_t h_{t-1} + E_t \iota_t$ of Theorem 2.4(iv) carries from one position to the next. It is the invariant that limits streaming retrieval: a recurrence with state dimension d_h carries at most d_h independent values to the query (Theorem A.7).

The completeness theorem (Section 4) makes precise in what sense these five exhaust the structural axes for width separations. It does so within the fixed-embedding regime, in which the embedding of a token depends only on its content and its position. Beyond the canonical tasks we cannot yet prove that they exhaust these axes, and Theorem D.16 states what we conjecture.

Task-side counterparts. For each invariant $\mathcal{I}$ we define a task-side counterpart $\mathcal{I}(T)$, which measures how much of the invariant a task T requires. The completeness theorem compares this counterpart with the invariant of an architecture, axis by axis. The kernel rank $\mathrm{KRank}(T)$ of a linear-retrieval task T is its rank in the sense of Theorem A.1: the largest number of linearly independent coefficient vectors that its targets have at a single key set. For the canonical aggregation tasks Avg_n and Count_n, we set $\mathrm{KRank}(T) = 1$: each has a single nonzero coefficient vector.

For the piecewise degree we need a notion of dependence. The task T depends on position j at an input x if every neighborhood of x contains an input that differs from it at that position alone and on which the task takes a different value. The piecewise degree $\mathrm{PDeg}(T)$ is the largest k such that the inputs at which T depends on at least k positions form a set with nonempty interior. In words, there is a nonempty open set of inputs at each of which the target depends on at least $\mathrm{PDeg}(T)$ positions.

The output range $\mathrm{Bound}(T)$ is the supremum of the magnitude of T over its input domain, as a function of the length. The symmetry group $\mathrm{Sym}(T)$ is the largest group of permutations of the positions other than the read position under which T is invariant: applying such a permutation to the input never changes the target. The streaming rank $\mathrm{State}(T)$ is the rank of T as a streaming linear-retrieval task, that is, as a linear-retrieval task read at its last position. For Avg_n and Count_n, we likewise set $\mathrm{State}(T) = 1$. These aggregation conventions define their gates in Theorem 4.2(ii); they do not enlarge the task classes in its soundness claim.

Definition 3.2 (Invariant profile at a cost budget)**.** A resource setting of an architecture is a choice of its width d, depth L, head count H and state dimension d_h, written (d, L, H, d_h). Four of the quantities of Theorem 3.1 are functions of an architecture together with a resource setting. The kernel rank KRank reads the width, at the degree the architecture fixes. The piecewise degree PDeg and the output range Bound read the depth and the head count. The state dimension State reads the recurrent state dimension d_h. The symmetry group Sym depends on the architecture alone.

Let A be an architecture, and let C be a budget for the cost of Theorem 2.2. We write $\mathcal{I}(A; C)$ for the extreme value of the invariant $\mathcal{I}$ over the resource settings of A whose cost is at most C. On the four axes oriented upward, where a larger value means more capacity, the extreme value is the supremum. On the symmetry axis, where a smaller group means more capacity (Theorem D.3), it is the infimum under inclusion. On the Bound axis the comparison $\mathcal{I}(A; C) \geq \mathcal{I}(T)$ is read as $\mathrm{Bound}(A; C) > \mathrm{Bound}(T) - 1/2$. The slack of $1/2$ is there because every error the theorem allows is below $1/2$.

The profile at a cost budget says how much of each invariant an architecture can afford at a given cost. It is the reading at which the completeness theorem compares an architecture with a task, and Theorem D.2 works it out for each invariant. Table 1, in Section A, lists the invariants of the standard architectures at given parameters.

4 The Completeness Theorem

This section states the completeness theorem, the one result of the paper. Its proof, in Section C, rests on machinery that Section A collects, statement by statement. On the kernel-rank axis the machinery is the kernel-rank obstruction and the constructions that match it (Section A.1). On the piecewise-degree axis it is the influence-set bound (Section A.2). Three further ingredients complete the machinery (Section A.3). The first is a bound on the state of a linear recurrence, the second a lemma that extracts a single layer from a deeper model, and the third a construction that sums n values at logarithmic depth. The five classes of tasks on which the invariants act come first here, then the theorem, and then its first corollary.

The theorem ranges over five classes of tasks, one for each invariant. The class $\mathcal{T}_{\mathcal{I}}$ holds the tasks on which part (i) of the theorem shows that the invariant $\mathcal{I}$ is an obstruction, and its definition records what the proof on that axis needs of a task. On the Bound axis it records instead the tasks on which the axis has content.

Definition 4.1 (The five separation classes)**.** Let T be a task on inputs of length n. The task lies in the kernel-rank class $\mathcal{T}_{\mathrm{KRank}}$ if it is a linear-retrieval task (Theorem A.1) under the split embedding of Section 2.3. If it is realized to a positive error ϵ, it must also be of retrieval type, a further condition stated in the same definition. Its task-side complexity $\mathrm{KRank}(T)$ is then its rank. For the piecewise-degree class $\mathcal{T}_{\mathrm{PDeg}}$ the condition is that the real coordinates of T range over a product of nondegenerate intervals, for each value of its discrete coordinates. This condition makes the piecewise degree $\mathrm{PDeg}(T)$ of Section 3 defined: the domain then has nonempty interior, so the count $k = 0$ in its definition qualifies, and no count above the number of positions does.

The boundedness class $\mathcal{T}_{\mathrm{Bound}}$ contains the tasks that are defined at every length and whose output range grows faster than every polylogarithm of n. A task lies in the symmetry class $\mathcal{T}_{\mathrm{Sym}}$ if at least one permutation of the positions other than the read position changes it, and if every permutation π with $T \circ \pi \neq T$ satisfies $\sup_x |T(x) - T(\pi x)| \geq 1$. Finally, the recurrent-state class $\mathcal{T}_{\mathrm{State}}$ contains the linear-retrieval tasks under the split embedding that are read at their last position and whose coefficient vectors are supported on the positions before their first query token. As in the kernel-rank class, such a task must also be of retrieval type if it is realized to a positive error ϵ. Its task-side complexity $\mathrm{State}(T)$ is then its rank. The running examples of the five classes, in this order, are AR_N, Avg_n, Count_n, Index_n and AR_N again, read by a recurrence. A task may lie in several classes at once.

The theorem is stated next, and Theorem 1.1 in Section 1.1 is the same statement. Its part (ii) concerns the four canonical tasks of Theorem 2.3: associative recall AR_N, the mean Avg_n and the sum Count_n of n values, and indexing Index_n. A row of Table 1 is one of the architectures listed there.

Theorem 4.2 (Completeness theorem, restatement of Theorem 1.1)**.** *Let A be a standard architecture (Theorem 2.4), with a fixed embedding as in Theorem 2.4(ii).*

(i) Soundness. *Fix one of the five invariants $\mathcal{I} \in \{\mathrm{KRank}, \mathrm{PDeg}, \mathrm{Bound}, \mathrm{Sym}, \mathrm{State}\}$ and a task T in its class $\mathcal{T}_{\mathcal{I}}$. Suppose that a model of A of depth L realizes T to error $\epsilon < 1/2$ at cost C. Then $\mathcal{I}(A; C) \geq \mathcal{I}(T)$, in the orientation of Theorem D.3. On the* KRank *axis this holds at $L = 1$ unconditionally, and at $L > 1$ for realizations that are layered in the sense of Theorem A.8, provided that the single layer extracted by the lemma reads a split embedding. On the* PDeg *axis it holds for exact realization. On the* Bound *and* Sym *axes it holds at every depth. On the* State *axis it holds for the single recurrence of Theorem 2.4(iv).*

(ii) Sufficiency on the canonical tasks. *Let T be one of the canonical tasks AR_N, Avg_n, Count_n and Index_n. Let A be a row of Table 1 whose mask lets the read position (Theorem 2.3) see every input position, and let C be a cost budget. Suppose that $\mathcal{I}(A; C) \geq \mathcal{I}(T)$ for all five invariants. Then A realizes T at cost $\tilde{O}(C)$. The realization is exact, except in the cells that Theorem D.20 names, where it is to every error $\epsilon \geq e^{-\mathrm{polylog}(n)}$.*

(iii) Witness. *Consequently, let T be such a task, and let A_1 and A_2 be two such rows that share a mask. Suppose that A_1 realizes T at cost C_1, by a realization to which part (i) applies on every axis, while A_2 requires cost $\omega(\tilde{O}(C_1))$ at the tolerance of part (ii). Then, for all large n, some invariant satisfies $\mathcal{I}(A_2; C_1) < \mathcal{I}(T) \leq \mathcal{I}(A_1; C_1)$.*

Proof in Section C.

We call the inequality $\mathcal{I}(A; C) \geq \mathcal{I}(T)$ on a single axis the *gate* of that axis, and we say that the gate clears when the inequality holds. On the symmetry axis the gate reads $\mathrm{Sym}(A; C) \subseteq \mathrm{Sym}(T)$, because a larger equivariance group is a stronger constraint (Theorem D.3).

In terms of gates, part (i) says that every gate is necessary. An architecture that realizes a task of the class $\mathcal{T}_{\mathcal{I}}$ to error below $1/2$ at cost C clears the gate of $\mathcal{I}$ at the budget C. Part (i) carries three restrictions: the layered hypothesis on the kernel-rank axis at depth above one, exact realization on the piecewise-degree axis, and a single recurrence on the recurrent-state axis. Theorem D.13, Theorem D.14 and Theorem D.17 say why each restriction is there.

Part (ii) says that on the canonical tasks the five gates are also jointly sufficient. Consider a row whose mask lets the read position see every input position, and whose five gates all clear at a budget C. That row realizes the task at the tolerance part (ii) states and at cost $\tilde{O}(C)$, that is, within a polylogarithmic factor of C.

Part (iii) combines the two halves. Suppose that one row realizes a canonical task at cost C_1, by a realization that part (i) covers on every axis. Suppose also that a second row with the same mask needs, at the tolerance of part (ii), a cost more than a polylogarithmic factor above C_1. Then, for all large n, some gate clears for the first row and fails for the second, both at the budget C_1.

The first corollary reads the theorem on a single row: NoPE softmax, which is softmax attention with no positional embedding, under the bidirectional mask. Its symmetry group is the group S_n of all permutations of the positions other than the read position (Theorem 3.1).

Corollary 4.3 (What a position-free architecture realizes)**.** *Among the canonical tasks of Theorem 4.2(ii), NoPE softmax with a bidirectional mask realizes precisely those that are invariant under S_n. It realizes AR_N at width $O(\log N)$, and it realizes Avg_n and Count_n at the costs that Theorem 4.2(ii) gives. It fails Index_n at every cost. The two retrieval tasks AR_N and Index_n are thus separated by the Sym axis alone.*

Proof in Section C.

Theorem D.15 compares this corollary with the known characterization of position-free average-hard attention.

The remaining consequences read the theorem at a cost budget, and Section B states them. The first, which Table 2 tabulates, is the least cost of every canonical task on every row of Table 1 other than sliding-window softmax. The second is the width that single-layer linear attention needs on associative recall. The third is a strict hierarchy of the polynomial kernels in their degree.

AI USE STATEMENT

Claude (Anthropic) and Codex (OpenAI) helped develop and check proofs, search and verify relevant literature, and draft and edit the manuscript. The authors posed the research question, reviewed the AI-assisted work, and take responsibility for the paper and any errors.

Under the ICLR 2027 AI Policy for Authors, work on the theoretical framework, mathematical claims, and proofs requires disclosure; drafting, literature work, and reference editing are recommended for disclosure. Translation, data, experiments, and research code do not arise in this paper.

REPRODUCIBILITY STATEMENT

All results are theoretical; no experiments, datasets, or research code are involved. The architecture, task, and cost conventions are defined in the main text. The appendices collect the proof ingredients and full arguments, with pointers from the corresponding statements.

REFERENCES

Dimitris Achlioptas. Database-friendly random projections: Johnson–Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

Noga Alon. Perturbed identity matrices have high rank: Proof and applications. *Combinatorics, Probability and Computing*, 18(1-2):3–15, 2009.

Eric Alsmann, Lowejatan Noori, and Martin Lange. On the expressiveness of state space models via temporal logics. In *International Conference on Learning Representations (ICLR)*, 2026.

Noah Amsel, Gilad Yehudai, and Joan Bruna. Quality over quantity in attention layers: When adding more heads hurts. In *International Conference on Learning Representations (ICLR)*, 2025. Preprint as arXiv:2407.16153, "On the Benefits of Rank in Attention Layers".

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. In *International Conference on Learning Representations (ICLR)*, 2024a.

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall-throughput tradeoff. In *International Conference on Machine Learning (ICML)*, 2024b.

Satwik Bhattamishra, Arkil Patel, and Navin Goyal. On the computational power of transformers and its implications in sequence modeling. In *Proceedings of the 24th Conference on Computational Natural Language Learning (CoNLL)*, 2020.

Satwik Bhattamishra, Michael Hahn, Phil Blunsom, and Varun Kanade. Separations in the representational capabilities of transformers and recurrent architectures. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 36002–36045, 2024.

Srinadh Bhojanapalli, Chulhee Yun, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Low-rank bottleneck in multi-head attention models. In *International Conference on Machine Learning (ICML)*, 2020.

Richard P. Brent and H. T. Kung. A regular layout for parallel adders. *IEEE Transactions on Computers*, C-31(3):260–264, 1982.

Lijie Chen, Binghui Peng, and Hongxun Wu. Theoretical limitations of multi-layer transformer. In *Proceedings of the 66th IEEE Symposium on Foundations of Computer Science (FOCS)*, 2025. arXiv:2412.02975.

David Chiang, Peter Cholak, and Anand Pillay. Tighter bounds on the expressivity of transformer encoders. In *International Conference on Machine Learning (ICML)*, 2023.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with Performers. In *International Conference on Learning Representations (ICLR)*, 2021.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML)*, 2024.

Mete Demircigil, Judith Heusel, Matthias Löwe, Sven Upgang, and Franck Vermet. On a model of associative memory with huge storage capacity. *Journal of Statistical Physics*, 168(2):288–299, 2017.

Ayoub Ghriss and Sourav Chakraborty. Kernelized linear attention: Breaking the capacity wall with symmetric cones. *arXiv preprint arXiv:2607.17419*, 2026.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling (COLM)*, 2024.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations (ICLR)*, 2022.

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

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

W. Daniel Hillis and Guy L. Steele, Jr. Data parallel algorithms. *Communications of the ACM*, 29 (12):1170–1183, 1986.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Joseph JáJá. *An Introduction to Parallel Algorithms*. Addison-Wesley, 1992.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. In *International Conference on Machine Learning (ICML)*, 2024.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning (ICML)*, 2020.

Peter M. Kogge and Harold S. Stone. A parallel algorithm for the efficient solution of a general class of recurrence equations. *IEEE Transactions on Computers*, C-22(8):786–793, 1973.

Teuvo Kohonen and Matti Ruohonen. Representation of associated data by matrix operators. *IEEE Transactions on Computers*, C-22(7):701–702, 1973.

Dmitry Krotov and John J. Hopfield. Dense associative memory for pattern recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Richard E. Ladner and Michael J. Fischer. Parallel prefix computation. *Journal of the ACM*, 27(4): 831–838, 1980.

Jiaoda Li and Ryan Cotterell. Characterizing the expressivity of fixed-precision transformer language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Jiaoda Li and Ryan Cotterell. Characterizing the expressivity of local attention in transformers. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pp. 37485–37507, 2026.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In *International Conference on Learning Representations (ICLR)*, 2023.

Will Merrill and Ashish Sabharwal. A little depth goes a long way: The expressive power of log-depth transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

William Merrill and Ashish Sabharwal. The parallelism tradeoff: Limitations of log-precision transformers. *Transactions of the Association for Computational Linguistics*, 11:531–545, 2023a.

William Merrill and Ashish Sabharwal. A logic for expressing log-precision transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b.

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

William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models. In *International Conference on Machine Learning (ICML)*, pp. 35492–35506, 2024.

Beren Millidge, Tommaso Salvatori, Yuhang Song, Thomas Lukasiewicz, and Rafal Bogacz. Universal Hopfield networks: A general framework for single-shot associative memory models. In *International Conference on Machine Learning (ICML)*, 2022.

Nicola Muca Cirone, Antonio Orvieto, Benjamin Walker, Cristopher Salvi, and Terry Lyons. Theoretical foundations of deep selective state-space models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Franz Nowak, Ryan Cotterell, and Reda Boumasmoud. A compositional theory of causally masked transformers. *arXiv preprint arXiv:2607.26988*, 2026.

Binghui Peng, Srini Narayanan, and Christos Papadimitriou. On limitations of the transformer architecture. In *Conference on Language Modeling (COLM)*, 2024. arXiv:2402.08164.

Jorge Pérez, Javier Marinković, and Pablo Barceló. On the Turing completeness of modern neural network architectures. In *International Conference on Learning Representations (ICLR)*, 2019.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning (ICML)*, 2023.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. In *International Conference on Learning Representations (ICLR)*, 2021.

Michael Rizvi-Martel, Maude Lizaire, Clara Lacroce, and Guillaume Rabusseau. Simulating weighted automata over sequences and trees with transformers. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 2368–2376, 2024.

Marco Sälzer, Chris Köcher, Alexander Kozachinskiy, Georg Zetzsche, and Anthony Widjaja Lin. The counting power of transformers. In *International Conference on Learning Representations (ICLR)*, 2026.

Clayton Sanford, Daniel J. Hsu, and Matus J. Telgarsky. Representational strengths and limitations of transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Clayton Sanford, Daniel Hsu, and Matus Telgarsky. One-layer transformers fail to solve the induction heads task. *arXiv preprint arXiv:2408.14332*, 2024a.

Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Transformers, parallel computation, and logarithmic depth. In *International Conference on Machine Learning (ICML)*, 2024b.

Yash Sarrof, Yana Veitsman, and Michael Hahn. The expressive capacity of state space models: A formal language perspective. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning (ICML)*, 2021.

Lena Strobl. Average-hard attention transformers are constant-depth uniform threshold circuits. *arXiv preprint arXiv:2308.03212*, 2023.

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

Anej Svete, William Merrill, Ryan Cotterell, and Ashish Sabharwal. Revisiting padded transformer expressivity: Which architectural choices matter and which don't. *arXiv preprint arXiv:2605.30523*, 2026.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer dissection: An unified understanding for transformer's attention via the lens of kernel. In *Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Andy Yang, David Chiang, and Dana Angluin. Masked hard-attention transformers recognize exactly the star-free languages. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Andy Yang, Michaël Cadilhac, and David Chiang. Knee-deep in C-RASP: A transformer depth hierarchy. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, 2025. arXiv:2506.16055.

Gilad Yehudai, Noah Amsel, and Joan Bruna. Compositional reasoning with transformers, RNNs, and chain of thought. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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 (ICLR)*, 2020.

Nikola Zubić, Qian Li, Yuyi Wang, and Davide Scaramuzza. On the expressive power and limitations of multi-layer SSMs. *Transactions on Machine Learning Research*, 2026.

A The Machinery of the Proof

This appendix collects the statements on which the proof of the completeness theorem (Theorem 4.2) rests, in the order that proof uses them; their proofs are in Section C. Section A.1 gives the kernel-rank obstruction, Section A.2 the influence-set bound, and Section A.3 three further ingredients. The proof of part (ii) of the theorem reads the architectures row by row from Table 1, which lists the five invariants of each standard architecture of Section 3.

The statements below and their proofs speak of the residual stream at a position. It is the vector that a layer leaves at that position, and the next layer, or the readout, reads it there. After an attention layer with a residual connection, the residual stream at a position is the layer's input at that position plus the attention output there.

Table 1: Structural-invariant profiles of the standard architectures at width d, depth L and length n. The masks are bidirectional (all positions), causal (the current and earlier positions), and windowed (distance at most w). For polynomial attention of degree p, the kernel rank KRank is $\binom{d+p}{p}$; for hard and average-hard attention, ∞ is a convention because their scores depend on the visible key set. The piecewise degree PDeg counts the positions influencing the read output on each input region; ∞ denotes the cap n. The output range Bound is shown up to the polylogarithmic factor of the norm convention; ∞ means growth with length at fixed depth. The symmetry group Sym is S_n for NoPE softmax and trivial otherwise. The state dimension State is d_h for recurrent rows and ∞ for feedforward rows. Costs are comparable only between rows with a common mask.

Architecture	Mask	KRank	PDeg	Bound	Sym	State
Softmax	bidir.	∞	∞	$2^{\Theta(L)}$	$\{e\}$	∞
Hard	bidir.	∞	2^L	$2^{\Theta(L)}$	$\{e\}$	∞
Average-hard	bidir.	∞	∞	$2^{\Theta(L)}$	$\{e\}$	∞
Polynomial degree p	bidir.	$\binom{d+p}{p}$	∞	∞	$\{e\}$	∞
Sigmoid	bidir.	∞	∞	∞	$\{e\}$	∞
Sliding-window softmax	window w	∞	$Lw+1$	$2^{\Theta(L)}$	$\{e\}$	∞
NoPE softmax	bidir.	∞	∞	$2^{\Theta(L)}$	S_n	∞
Causal softmax	causal	∞	∞	$2^{\Theta(L)}$	$\{e\}$	∞
Linear SSM	causal	∞	∞	∞	$\{e\}$	d_h
Selective SSM	causal	∞	∞	∞	$\{e\}$	d_h

A.1 The Kernel-Rank Obstruction

This subsection holds the machinery behind the first axis, the kernel-rank obstruction and the constructions that match it. It first defines linear-retrieval tasks, the class of tasks on which the kernel rank is an obstruction. It then bounds the rank of the kernel matrix of a polynomial score functional (Theorem A.2). The kernel-rank obstruction (Theorem A.3) turns that bound into a lower bound on the width, and its second part gives a construction at logarithmic width for kernels such as softmax and sigmoid. Finally, Theorem A.4 shows that the lower bound for polynomial kernels is attained. The completeness theorem uses the argument of Theorem A.3(a) for its soundness on the KRank axis, and the constructions of Theorem A.3(b) and Theorem A.4 for its sufficiency on the retrieval tasks.

Definition A.1 (Linear-retrieval task)**.** Consider a task T whose inputs consist of keys $K = (k_1, \dots, k_n)$, values $v = (v_1, \dots, v_n)$ and a query q. A pair (K, q) is admissible if some input of T has keys K and query q. We then call K an admissible key set and q a query admissible for K. The task T is a linear-retrieval task if two conditions hold at every admissible pair (K, q). First, every value vector $v \in [-1, 1]^n$ gives an input (K, v, q) of T. Second, the target of T at each such input is a linear functional $\mathbf{c}(K, q)^\top v$ of the values, with a coefficient vector $\mathbf{c}(K, q) \in \mathbb{R}^n$ determined by the keys and the query. The rank r of the task is the largest number of linearly independent vectors that the family $\{\mathbf{c}(K, q) \mid q \text{ admissible for } K\}$ contains at a single admissible key set K. The task is of *retrieval type* if the rank is attained at a key set K whose family contains the standard basis vectors $e_1, \dots, e_r$ of $\mathbb{R}^n$ after a fixed reordering of the coordinates. We call the r coordinates that the reordering sends to $1, \dots, r$ the retrieval coordinates of T.

A linear-retrieval task asks for a linear combination of the values, chosen by the keys and the query. Its rank counts how many independent combinations it can ask for at one key set. A task of retrieval type asks for r of the values one at a time, since the coefficient vector e_i gives the target v_i. The rank is the task-side kernel rank $\mathrm{KRank}(T)$ of Section 3, and the class $\mathcal{T}_{\mathrm{KRank}}$ of Theorem 4.1 consists of linear-retrieval tasks. The associative-recall task AR_N and the indexing task Index_n are of retrieval type, with rank $r = N$ for the first and $r = n$ for the second. For both, at every key set the family of coefficient vectors is the full set of selection vectors, the standard basis vectors that each pick out one value.

The first ingredient bounds the rank of the kernel matrix of a polynomial score functional. The proof of Theorem A.3(a) applies it at dimension $d - 1$.

Lemma A.2 (Bilinear kernel factorization)**.** *Let ϕ and ϕ' be maps from $\mathbb{R}^d$ to the same Euclidean space, each of whose coordinates is a polynomial of degree at most p, and let $\sigma(q,k) = \langle \phi(q), \phi'(k) \rangle$ be the kernel that they define. Let $\tilde{q} \in \mathbb{R}^d$ range over a family of query vectors, indexed by the query q, and let $\tilde{k}_1, \ldots, \tilde{k}_n \in \mathbb{R}^d$ be key vectors. Let Z be a positive function of the query. Then the kernel matrix G with entries $G_{q,j} = \sigma(\tilde{q}, \tilde{k}_j)/Z(q)$ satisfies* $\operatorname{rank} G \le \binom{d+p}{p}$.

Proof in Section C.

The bound $\binom{d+p}{p}$ is the number of monomials of degree at most p in d variables. However many queries there are, the rows of the kernel matrix lie in a space of that dimension, and dividing each row by its normalizer does not change the rank.

The next lemma is the kernel-rank obstruction. Its part (a) turns the rank bound into a lower bound on the width of every polynomial model that realizes a linear-retrieval task, exactly or, for a task of retrieval type, to error below 1. Its part (b) gives, for kernels such as softmax and sigmoid, a model whose width is logarithmic in the rank.

Lemma A.3 (Kernel-rank obstruction, and the matching constructions)**.** *Let A be a single-layer attention architecture, normalized or unnormalized, with the split embedding of Section 2.3, in which each key-value position carries its value in a channel of its own that the query and key maps do not read. Its readout may be an arbitrary function of the residual stream at the query position. Its score functional is of one of two kinds. In the polynomial case it is a polynomial kernel $\sigma_A(q,k) = \langle \phi(q), \phi'(k) \rangle$, with feature maps ϕ and ϕ' of degree at most p. Polynomial-linear attention of degree p is the case in which $\phi' = \phi$. In the analytic case it is $\sigma_A(q,k) = f_A(\langle q, k \rangle)$ for a real-analytic function $f_A : \mathbb{R} \to \mathbb{R}$ that is not a polynomial and is positive where the functional is normalized. Let T be a linear-retrieval task of rank r.*

(a) In the polynomial case, let a model in the class of A realize T exactly, or to error $\epsilon < 1$ when T is of retrieval type. Then $\binom{d-1+p}{p} \ge r$, and hence the width of the model satisfies

$$d \ge \frac{p}{e} r^{1/p} - p + 1,$$

which is at least $\frac{p}{2e} r^{1/p} + 1$ whenever $r \ge (2e)^p$.

(b) In the analytic case, call f_A exponentially separating *if reals $s < t$ and a constant $c > 0$ exist with the following property. For every large β, the ratio $f_A(\beta s)/f_A(\beta t)$ is at most $e^{-c\beta}$. Suppose that f_A is positive, nondecreasing and exponentially separating. Then, for every $r \ge 2$ and every $\epsilon \in (0,1]$, there is a model in the class of A of width $\lceil 24 \log_2 r \rceil + 3$ with a linear readout. This model realizes AR_r to error ϵ. The embedded tokens of this model have norm $O(\sqrt{\log(r/\epsilon)})$. Both* exp *and* sigmoid *are exponentially separating, the first with every pair $s < t$ and the second with every pair $s < 0 < t$.*

Proof in Section C.

In part (a) the width enters through $\binom{d-1+p}{p}$ rather than $\binom{d+p}{p}$ because the query and key maps do not read the value channel, so the kernel factors through the polynomial feature space of the remaining $d-1$ coordinates. Part (a) therefore forces width at least $\frac{p}{2e} r^{1/p}$ on a polynomial kernel of degree p once $r \ge (2e)^p$, while part (b) needs only width logarithmic in r for an exponentially separating kernel. Theorem D.6 discusses this gap between the two regimes. Part (b) is an approximate realization, and Theorem D.5 explains why no single layer of softmax or sigmoid attention with the split embedding realizes AR_N exactly.

The next proposition shows that the lower bound of part (a) is attained: a polynomial model of degree p realizes AR_N exactly at the least width that part (a) allows.

Proposition A.4 (Matching polynomial construction)**.** *Fix $p \ge 1$ and N, and let $d_p(N)$ be the least d with $\binom{d+p}{p} \ge N$. There is a depth-1 unnormalized polynomial-linear model of degree p that realizes AR_N exactly at width $d_p(N) + 1$, and this width is optimal by Theorem A.3(a). There is also such a model at width $p\lceil N^{1/p} \rceil + 1$, with an explicit feature map. Both widths are $\Theta(pN^{1/p})$ when $N \ge (2e)^p$.*

Proof in Section C.

The width $d_p(N) + 1$ has a direct reading: $d_p(N)$ is the least number of variables in which the polynomials of degree at most p form a space of dimension at least N, and the extra coordinate is the value channel. With Theorem A.3(a), the proposition fixes the width of polynomial attention on AR_N, and the construction it gives is the realization that the sufficiency half of Theorem 4.2 uses for the polynomial row.

A.2 The Influence-Set Bound

This subsection holds the machinery behind the piecewise degree of Section 3. It first defines selectivity, which bounds how many positions each query of a layer attends to, together with the semi-analytic regions on which selectivity is read. The influence-set bound (Theorem A.6) then compounds selectivity over the layers into a bound on the piecewise degree, and hence into a lower bound on the depth of exact realizations.

Definition A.5 (Semi-analytic regions, and selectivity)**.** A basic set of $\mathbb{R}^m$ is a set of the form

$$\{x : f_1(x) = \cdots = f_a(x) = 0 \text{ and } g_1(x) > 0, \ldots, g_b(x) > 0\}$$

with $f_1, \ldots, f_a, g_1, \ldots, g_b$ real-analytic on $\mathbb{R}^m$. A semi-analytic region of $\mathbb{R}^m$ is a finite union of basic sets. A score functional is κ-selective if a cover of the following kind exists at every length n and for every choice of the parameters of a layer that uses the functional. The cover is a family of finitely many semi-analytic regions whose union is the space $(\mathbb{R}^d)^n$ of layer inputs, and it has two properties. First, on each region R and at every query position i, some set S_i of at most κ positions carries every nonzero attention weight. Second, on R each such weight is the restriction of a function real-analytic on $(\mathbb{R}^d)^n$ that depends only on the layer inputs at i and at the positions of S_i.

In words, a basic set is cut out by finitely many real-analytic equations and strict inequalities, and a semi-analytic region is a finite union of such sets. A κ-selective functional attends, on each region of a finite cover, to at most κ positions from each query, and within a region its weights vary analytically. Semi-analytic regions are also the pieces of the input domain on which the piecewise degree of Theorem 3.1 counts positions.

Hard attention, with its ties broken as Section 2.1 fixes, is 1-selective. Its regions are those on which every argmax is a fixed position. Average-hard attention is n-selective. At the inputs whose argmax sets have at most κ positions, it puts the weight of each query on at most κ positions. Top-κ attention, which keeps the κ largest scores of each query and gives every other position weight zero, is κ-selective with no condition on the argmax sets. Sliding-window attention of half-width w is $(2w{+}1)$-selective. Softmax, like every unmasked analytic functional with a nonvanishing normalizer, is n-selective. A layer of H heads that are κ-selective is $H\kappa$-selective, its active positions being the union of those of its heads.

The next proposition compounds selectivity over the layers. On each region, a layer makes the representation at a position i a function of the previous representation at i, carried by the residual connection, and of the representations at the at most κ positions that i attends to. So the set of input positions that the representation depends on grows by a factor of at most $\kappa + 1$ per layer. The proposition gives the general bound on the piecewise degree that Theorem 3.1 cites.

Proposition A.6 (Influence-set growth under bounded selectivity)**.** *Let A be a depth-L architecture whose score functional is κ-selective, and fix a model of A. Then the input domain of the model is covered by finitely many semi-analytic regions. On each of these regions the layer-L representation at every position is the restriction of a real-analytic function of the input. Moreover, on each region the output at every position is a function of at most $(\kappa + 1)^L$ input positions. Hence the piecewise degree satisfies $\mathrm{PDeg}(A, L) \le (\kappa + 1)^L$. If a task depends on all n positions at every input of some nonempty open set of inputs, then an exact realization of it by a model of A forces depth $L \ge \log n / \log(\kappa + 1)$. For hard attention, which is the case $\kappa = 1$, the bound is 2^L and the requirement is $L \ge \log_2 n$.*

Proof in Section C.

The depth requirement is what blocks hard attention from realizing Avg_n and Count_n exactly below depth $\log_2 n$. Under the bidirectional mask, part (c) of Theorem A.9 attains the least depth that this

requirement allows, which is $\lceil \log_2 n \rceil$, at every n. The argument is local and needs exact agreement on an open set, and Theorem D.14 explains why it gives nothing at positive error.

A.3 Three Further Ingredients

The first ingredient bounds the state dimension of a recurrence, and the proof of Theorem 4.2(i) uses it on the recurrent-state axis. The recurrence reads its input through the split embedding of Section 2.3, which embeds a key-value position as $u_j + v_j w$: the key part u_j depends on the key alone, and the value v_j sits along the value direction w.

Lemma A.7 (Linear-recurrence rank bound)**.** *Consider a model that embeds an input of length $N+1$ by the split embedding of Section 2.3 and reads it through a linear recurrence. Its state after position t is $h_t = R_t h_{t-1} + E_t \iota_t$, starting from $h_0 = 0$, where ι_t is the embedded token at position t. The transition matrix $R_t \in \mathbb{R}^{d_h \times d_h}$ and the input matrix $E_t \in \mathbb{R}^{d_h \times d}$ are fixed, or are fixed functions of the key channel of the token at position t. The readout may be any token-wise map of the final state h_{N+1} and the query token. If the model realizes AR_N to error $\epsilon < 1$, then its state dimension satisfies $d_h \geq N$. The same holds with N replaced by r for a streaming retrieval task of r values, which is a linear-retrieval task of retrieval type and rank r whose r retrieval coordinates precede the query.*

Proof in Section C.

The proof uses nothing about the recurrence beyond its state being affine in the values, so the bound covers every transition and input matrix that reads the key channel alone.

The second ingredient lifts lower bounds from depth one to greater depth, and the proof of Theorem 4.2(i) uses it on the kernel-rank axis at depth $L > 1$.

Lemma A.8 (Compositional inheritance under a layered realization)**.** *Let A realize a task T to error ϵ* layeredly*, meaning that some layer ℓ has two properties. First, the input to layer ℓ at every position is a fixed function of the token at that position. Second, the model's output is a function of the output of layer ℓ at the read position alone. Then layer ℓ is itself a depth-1 model that realizes T to the same error ϵ. Its embedding is the fixed function of the first property, and its readout is the map of the second property, from the output of layer ℓ at the read position to the model's output. This depth-1 model has the score functional, the width and the head count of A, and its readout is token-wise. Consequently, a lower bound on depth-1 realizations of T applies to A under two conditions. The bound must allow the readout to be an arbitrary function of the residual stream at the read position, and the extracted embedding must meet the bound's other hypotheses.*

Proof in Section C.

Theorem D.13 explains why the layered hypothesis is needed.

The third ingredient is a construction rather than a bound. It computes the sum of n values at the read position by recursive doubling. At the doubling layer of step k, each position $i > 2^k$ adds the partial sum held at position $i - 2^k$. At every position $i \geq 2^{k+1}$ this doubles the length of the window of values that the position has summed, from 2^k to 2^{k+1}. The proof of Theorem 4.2(ii) uses part (a) of the next lemma for hard attention on Avg_n and Count_n, and its part (b) for causal softmax on Count_n.

Lemma A.9 (Recursive doubling: summation at logarithmic depth)**.** *Let $n \geq 2$, and let $D = 2\lfloor \log_2 n \rfloor$, which bounds the depths of the models in parts (a) and (b) below.*

- *(a) There is a hard attention model of depth at most D and width 4, with a rotary positional embedding, that realizes Count_n exactly at the read position, under the bidirectional and under the causal mask alike. When n is a power of two, depth $\log_2 n$ and width 3 suffice.*
- *(b) For every $\epsilon \in (0, 1]$ there is a softmax attention model of depth at most D and width $2\lceil \log_2 n \rceil + 2$, with a rotary positional embedding, that realizes Count_n to error ϵ at the read position under either mask. Its embedded tokens have norm $O(\sqrt{\log n \cdot \log(n/\epsilon)})$. This norm is within the norm convention of Section 2.1 when $\log(1/\epsilon)$ is polylogarithmic in n.*

Table 2: The cost of Theorem 2.2 at which each architecture of Table 1 realizes each canonical task. Each entry is the cost $L(d^2 + d_h^2)$ of Theorem 2.2, for depth L, width d and state dimension d_h, of the realization that Theorem 4.2(ii) supplies. By Theorem B.1, each entry is minimal up to polylogarithmic factors. In the column heads, N is the number of key-value pairs of associative recall AR_N. Each of the other three tasks carries n values. In the polynomial row, p is the degree, and $d_p(N)$ is the least d with $\binom{d+p}{p} \geq N$, as in Theorem A.4. An entry 1 is a constant cost, and an entry ∞ means that the architecture does not realize the task at any cost. The last column is the cost of realizing Count_n by one model at every length. Costs are comparable only between rows that share a mask, bidirectional (bidir.) or causal. Theorem D.20 says which entries are exact and why the sliding-window row is absent.

Architecture	Mask	AR_N	Index_n	Avg_n	Count_n	Count uniform
Softmax	bidir.	$\log^2 N$	$\log^2 n$	1	$\log n$	∞
Hard	bidir.	$\log^2 N$	$\log^2 n$	$\log n$	$\log n$	∞
Average-hard	bidir.	$\log^2 N$	$\log^2 n$	1	$\log n$	∞
Polynomial degree p	bidir.	$d_p(N)^2$	$d_p(n)^2$	1	1	1
Sigmoid	bidir.	$\log^2 N$	$\log^2 n$	1	1	1
NoPE softmax	bidir.	$\log^2 N$	∞	1	$\log n$	∞
Causal softmax	causal	$\log^2 N$	$\log^2 n$	1	$\log^3 n$	∞
Linear SSM	causal	$N^2 \log^2 N$	$n^2 \log^2 n$	1	1	1
Selective SSM	causal	N^2	n^2	1	1	1

(c) Under the bidirectional mask there is a hard attention model of depth $\lceil \log_2 n \rceil$ and width 3, with a rotary positional embedding, that realizes Count_n exactly at the read position, whether or not n is a power of two. Under every mask, no hard attention model of depth below $\lceil \log_2 n \rceil$ realizes Count_n or Avg_n exactly.

In every part, a readout that divides the output by n gives Avg_n.

Proof in Section C.

B Consequences

This section draws three further consequences from Theorem 4.2. The first reads the theorem at a cost budget, and it says what a canonical task costs on a row: up to polylogarithmic factors, the cost is the least budget at which all five gates clear. Table 2 writes these costs out, and Theorem D.2 works out how each gate responds to the budget.

Corollary B.1 (Closed-form prediction)**.** *Let T be one of the canonical tasks of Theorem 4.2(ii), and let A be a row of Table 1 whose mask lets the read position see every input position. Fix an error $\epsilon < 1/2$ within the tolerance of Theorem 4.2(ii), and consider the least cost at which A realizes T to error ϵ by a realization to which Theorem 4.2(i) applies on every axis. Up to polylogarithmic factors that do not depend on ϵ, this cost is the least budget C at which $\mathcal{I}(A; C) \geq \mathcal{I}(T)$ holds for all five invariants.*

Proof in Section C.

Table 2 lists these costs, each minimal up to polylogarithmic factors, for every row and every canonical task.

The other two consequences concern the kernel-rank axis alone. The first fixes the degree at one, which is linear attention, and allows several heads.

Corollary B.2 (Linear attention on associative recall)**.** *Let A be single-layer linear attention, the case $p = 1$ of Theorem A.3, with H heads of per-head width d/H aggregated additively. Every model in the class of A that realizes AR_N to error $\epsilon < 1$ has width $d \geq \max\{N, H\}$. Exact realization is possible at every width that is a multiple of H and at least $N + 1$, and so at some width at most $N + H$. When $H = 1$, exact realization is possible already at width N, which is then*

optimal. So the width is $\Omega(N)$ at every head count, and $\Theta(N)$ when $H = O(N)$. This width is exponentially above the width $O(\log N)$ of Theorem A.3(b).

Proof in Section C.

The second compares two consecutive degrees of polynomial-linear attention on associative recall.

Corollary B.3 (Strict degree hierarchy)**.** *Let $p \geq 1$. Single-layer polynomial-linear attention of degree $p+1$ realizes AR_N exactly at width $(p+1)\lceil N^{1/(p+1)}\rceil + 1$. At degree p the same architecture needs width at least $(p/e)N^{1/p} - p$, even to realize AR_N to error $\epsilon < 1$. At fixed p, the ratio of the degree-p lower bound to the degree-$(p+1)$ width is $\Theta(N^{1/(p(p+1))})$. This ratio exceeds 1.4 once $N \geq (2e(p+1)/p)^{p(p+1)}$. So the single-layer polynomial architectures form a strictly increasing hierarchy in the degree, and every level of it lies above the width $O(\log N)$ of the exponentially separating kernels.*

Proof in Section C.

C PROOFS

Every proof omitted from the body appears here, in the order its statement appears above.

Proof of Theorem A.2. We show that every row of the unnormalized kernel matrix lies in the row space of one fixed matrix with $\binom{d+p}{p}$ rows, and then account for the normalizer.

The real polynomials in d variables of total degree at most p form a vector space. Index a monomial by its exponent vector $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}^d$, so that $x^\alpha = x_1^{\alpha_1} \cdots x_d^{\alpha_d}$ has total degree $|\alpha| = \alpha_1 + \cdots + \alpha_d$. The monomials x^α with $|\alpha| \leq p$ are a basis of this space, and there are $\binom{d+p}{p}$ of them, so the space has dimension $\binom{d+p}{p}$. Let the monomial map $\chi : \mathbb{R}^d \to \mathbb{R}^{\binom{d+p}{p}}$ list those monomials. Every coordinate of the feature map ϕ, and every coordinate of the feature map ϕ', lies in that space. Hence $\phi = M\chi$ and $\phi' = M'\chi$ for fixed coefficient matrices M and M', and the score factors as

$$\sigma(q, k) = \langle M\chi(q), M'\chi(k)\rangle = \chi(q)^\top M^\top M'\chi(k).$$

Now write $\chi(\tilde{K})$ for the $\binom{d+p}{p} \times n$ matrix whose columns are the monomial vectors $\chi(\tilde{k}_1), \ldots, \chi(\tilde{k}_n)$ of the keys. By the factorization, the row of the unnormalized kernel matrix at a query $\tilde{q}$ is $\chi(\tilde{q})^\top M^\top M'\chi(\tilde{K})$. This row lies in the row space of the matrix $M^\top M'\chi(\tilde{K})$, which has size $\binom{d+p}{p} \times n$ and does not depend on the query. So the rows of the unnormalized kernel matrix span a space of dimension at most $\binom{d+p}{p}$, however many queries there are. Finally, dividing row q by the normalizer $Z(q) > 0$ is left multiplication by an invertible diagonal matrix, and it does not change the rank. □

Proof of Theorem A.3. The proof has three steps. We first write the attention output at the query position as a baseline that does not depend on the values, plus the kernel matrix applied to the values. Part (a) then argues by contradiction from this decomposition, and part (b) is a construction.

The kernel decomposition. By the split hypothesis of Section 2.3, the embedding of a key-value position is $\iota(k_j, v_j) = u_j + v_j w$. Here the key part u_j is a fixed function of the key, and the value direction w is a fixed vector. The query and key maps annihilate the value direction: $W_Q w = W_K w = 0$. So the query vector $\tilde{q} = W_Q \iota(q)$ and every key vector $\tilde{k}_j = W_K u_j$ are functions of the keys and the query alone, not of the values. In particular, neither the scores nor the normalizer depends on the values v. Applying the linear map W_V and separating the value channel, we write

the attention output at the query position as

$$\begin{aligned} y(q,v) &= \frac{1}{Z(q)} \sum_j \sigma_A(\tilde{q}, \tilde{k}_j) W_V(u_j + v_j w) \\ &\quad + \frac{\sigma_A(\tilde{q}, W_K \iota(q))}{Z(q)} W_V \iota(q) \\ &= \alpha_q + (W_V w) \sum_j G_{q,j} v_j. \end{aligned}$$

Both sums run over the key positions. The kernel matrix G has entries $G_{q,j} = \sigma_A(\tilde{q}, \tilde{k}_j)/Z(q)$, one row for each query and one column for each key position. The baseline α_q collects the terms in the key parts u_j together with the query's own summand, so it does not depend on the values v. In words, the output is the baseline plus the fixed vector $W_V w$ times $\sum_j G_{q,j} v_j$, the q-th coordinate of the kernel matrix times the value vector. The unnormalized case is the case $Z(q) \equiv 1$. In the normalized case the normalizer $Z(q)$ is a positive scaling of row q, and it changes neither the rank of the kernel matrix G nor anything below. The residual stream at the query position is $\iota(q) + y(q, v)$, and the readout is a function of that vector alone.

Part (a). We argue by contradiction. If the binomial coefficient $\binom{d-1+p}{p}$ were smaller than the rank r, we would find two inputs with opposite values that the model cannot tell apart but whose targets differ. We first bound the rank of the kernel matrix, then build the two inputs, and last convert the binomial inequality into the stated width bound.

The value direction w is nonzero. Indeed, if $w = 0$, then the output does not depend on v, and no task of rank $r \geq 1$ is realized. Let $J \in \mathbb{R}^{d \times (d-1)}$ be the basis matrix whose columns are an orthonormal basis of the orthogonal complement $w^\perp$ of the value direction. Since the query and key maps annihilate w, we have $W_Q = W_Q J J^\top$ and $W_K = W_K J J^\top$. Write $z = J^\top \iota(q)$ and $z_j = J^\top u_j$, two vectors in $\mathbb{R}^{d-1}$. The score between the query and key position j is therefore $\langle \phi(W_Q J z), \phi'(W_K J z_j) \rangle$. The maps $\phi \circ W_Q J$ and $\phi' \circ W_K J$ have coordinates that are polynomials of degree at most p in $d-1$ variables. So Theorem A.2, applied in dimension $d-1$, bounds the rank of the kernel matrix G_K at every key set K:

$$\operatorname{rank} G_K \leq \binom{d-1+p}{p}.$$

Suppose now that $\binom{d-1+p}{p} < r$. Fix an admissible key set K at which the rank r is attained; for T of retrieval type, take the key set that Theorem A.1 names. Fix also admissible queries $q_1, \ldots, q_r$ whose coefficient vectors $\mathbf{c}(K, q_i)$ are linearly independent; for T of retrieval type, take those with $\mathbf{c}(K, q_i) = e_i$. Let the coefficient matrix be the $r \times n$ matrix with rows $\mathbf{c}(K, q_i)^\top$. It has rank r, so some set S of r of its columns is nonsingular. For T of retrieval type, take S to be the r reordered coordinates, on which the coefficient matrix is the identity.

The columns of the kernel matrix G_K indexed by S have rank at most $\operatorname{rank} G_K$, which is less than r by the rank bound and the supposition. So some nonzero vector v supported on S has $G_K v = 0$. Scale it so that $\max_j |v_j| = 1$, which puts it in $[-1, 1]^n$. By the kernel decomposition, $y(q, v) = \alpha_q = y(q, -v)$ for every admissible query q. So the inputs (K, v, q) and $(K, -v, q)$ produce the same residual stream at the query position, and hence the same output. Their targets, however, differ by $2\mathbf{c}(K, q)^\top v$. The columns S of the coefficient matrix are nonsingular, and v is nonzero and supported on S. So some query q_i has $\mathbf{c}(K, q_i)^\top v \neq 0$, and at that query the two targets differ while the two outputs coincide. That rules out exact realization. For T of retrieval type the difference is explicit: $\mathbf{c}(K, q_i)^\top v = v_i$. At an index i with $|v_i| = 1$ the two targets differ by 2, while the two outputs coincide. So one of the two errors is at least 1, which exceeds ϵ. In both cases the supposition fails, and hence $\binom{d-1+p}{p} \geq r$.

It remains to convert this binomial inequality into a bound on the width d. We use the inequality $\binom{d-1+p}{p} \leq (e(d-1+p)/p)^p$, which combines two bounds. First, $\binom{d-1+p}{p} \leq (d-1+p)^p/p!$, since each of the p factors in the numerator of the binomial coefficient is at most $d-1+p$. Second, $p! \geq (p/e)^p$, since $p^p/p!$ is one term of the series $e^p = \sum_{i \geq 0} p^i/i!$. Chaining the combined inequality with the lower bound $\binom{d-1+p}{p} \geq r$ just proved gives $r \leq (e(d-1+p)/p)^p$. Hence

$d-1+p \geq (p/e)r^{1/p}$, which is the stated bound. For the last clause of the statement, let $r \geq (2e)^p$. Then $\frac{p}{2e}r^{1/p} \geq p$, and therefore the right-hand side of the bound on d satisfies

$$\begin{aligned}\frac{p}{e}r^{1/p} - p + 1 &= \frac{p}{2e}r^{1/p} + \left(\frac{p}{2e}r^{1/p} - p\right) + 1 \\ &\geq \frac{p}{2e}r^{1/p} + 1.\end{aligned}$$

Part (b). The proof is a construction in three steps. We give the r keys sign vectors with small pairwise inner products. We then build a layer in which the matching key scores βt and every other position scores at most βs. Last, we choose the scale β so that the separating hypothesis makes the error at most ϵ.

We begin with the sign vectors. Set $d_0 = \lceil 24 \log_2 r \rceil$. Let $\psi : [r] \to \{-1, +1\}^{d_0}$ be a random sign map, which gives each key $k \in [r]$ a sign vector $\psi(k)$. All rd_0 coordinates of these sign vectors are independent and uniform. Each sign vector has $\langle \psi(k), \psi(k) \rangle = d_0$ exactly. Now take two distinct keys $k \neq k'$ in $[r]$. The products $\psi(k)_i \psi(k')_i$, one for each coordinate i, are independent uniform signs. So the inner product $\langle \psi(k), \psi(k') \rangle$ is a sum of d_0 independent uniform signs. Hoeffding's inequality (Hoeffding, 1963) bounds the probability that such a sum exceeds $d_0/2$ in absolute value by $2\exp(-(d_0/2)^2/(2d_0))$:

$$\Pr\left[|\langle \psi(k), \psi(k') \rangle| > d_0/2\right] \leq 2e^{-d_0/8} \leq 2r^{-3},$$

the last step because $d_0 \geq 24 \log_2 r \geq 24 \ln r$. There are $r(r-1)/2 < r^2/2$ pairs of distinct keys, so a union bound leaves failure probability below $(r^2/2) \cdot 2r^{-3} = 1/r$. Fix a sign map ψ for which every non-matching inner product, between the sign vectors of two distinct keys, is at most $d_0/2$ in absolute value. This sign map is a Johnson–Lindenstrauss-style embedding (Johnson & Lindenstrauss, 1984) of the r standard basis vectors using random signs as in Achlioptas (2003).

Next we build the layer. Without loss of generality $t > 0$: replacing t by a larger value only decreases the ratio $f_A(\beta s)/f_A(\beta t)$, since f_A is nondecreasing. Fix a scale $\beta > 0$, to be chosen at the end, and put $a = 2\beta(t-s)/d_0$. The embedding has four blocks: a sign block, then a constant coordinate, then a flag coordinate that marks a key, and last the value channel. Key-value position j, with key $k_j \in [r]$ and payload v_j, is embedded as $x_j = (\sqrt{a}\psi(k_j), \sqrt{\beta}, \sqrt{\beta}, v_j) \in \mathbb{R}^{d_0+3}$. The query position $r+1$, with query $q \in [r]$, is embedded as $x_{r+1} = (\sqrt{a}\psi(q), \sqrt{\beta}, 0, 0)$, with flag 0 and value 0. Let the key map W_K be the identity on the first $d_0 + 2$ coordinates and zero on the value channel. Let the query map W_Q be the identity on the sign block. On the constant coordinate, let W_Q copy it into itself with factor $3s - 2t$ and into the flag coordinate with factor $t - s$. Let W_Q annihilate the flag and value coordinates. The score between the query and key position j is then

$$a\langle \psi(q), \psi(k_j) \rangle + (3s - 2t)\beta + (t-s)\beta = a\langle \psi(q), \psi(k_j) \rangle + (2s - t)\beta.$$

The query's flag is 0, so the query's score with itself is $ad_0 + (3s - 2t)\beta$. By the choice of a we have $ad_0 = 2\beta(t-s)$. Let j^* be the matching position, the one whose key is $k_{j^*} = q$, which exists and is unique because the keys are a permutation of $[r]$. It scores $ad_0 + (2s-t)\beta = \beta t$. Every other key position j has $k_j \neq q$, and so it scores at most $\beta(t-s) + (2s-t)\beta = \beta s$, by the bound $d_0/2$ on non-matching inner products. The query itself scores $2\beta(t-s) + (3s-2t)\beta = \beta s$. Let W_V extract the value channel, and let the readout be the identity on it.

We now bound the error, first for a normalized functional. Write θ_j for the attention weight on position j, which is f_A of its score divided by the sum of f_A over the scores of all $r+1$ positions. The r positions other than the matching position j^*, namely the $r-1$ non-matching key positions and the query position, score at most βs. Since f_A is positive and nondecreasing, the weight on the matching position satisfies

$$\theta_{j^*} \geq \frac{f_A(\beta t)}{f_A(\beta t) + r f_A(\beta s)} \geq 1 - r\frac{f_A(\beta s)}{f_A(\beta t)},$$

the second step by $1/(1+x) \geq 1 - x$ for $x \geq 0$. The output is $\sum_j \theta_j v_j$, the sum over all $r+1$ positions, where the query's own value is 0. Since the weights sum to 1, the output differs from the

target v_{j^*} by

$$\left|\sum_{j\neq j^*}\theta_j(v_j - v_{j^*})\right| \leq 2(1-\theta_{j^*}) \leq 2r\frac{f_A(\beta s)}{f_A(\beta t)},$$

the first step because each $|v_j - v_{j^*}|$ is at most 2 and the weights θ_j with $j \neq j^*$ are nonnegative and sum to $1 - \theta_{j^*}$. In the unnormalized case, let W_V scale the value channel by $1/f_A(\beta t)$, which is at most $1/f_A(0)$ because $t > 0$. Then the matching position contributes exactly v_{j^*} and the query position contributes 0. Each of the other $r - 1$ key positions contributes its value times a factor in $(0, f_A(\beta s)/f_A(\beta t)]$. The difference is then at most $(r-1)f_A(\beta s)/f_A(\beta t) \leq r f_A(\beta s)/f_A(\beta t)$.

Last, we choose the scale β. By the separating hypothesis there is a threshold $\beta_0 > 0$ such that the ratio $f_A(\beta s)/f_A(\beta t)$ is at most $e^{-c\beta}$ for every $\beta \geq \beta_0$. So every $\beta \geq \max\{\beta_0, (1/c)\log(2r/\epsilon)\}$ makes both differences at most ϵ. The threshold β_0 and the constant c depend on f_A alone. Since $r \geq 2$ and $\epsilon \leq 1$, the logarithm $\log(r/\epsilon)$ is at least $\log 2$. So $\beta_0 \leq (\beta_0/\log 2)\log(r/\epsilon)$, and $\log(2r/\epsilon) \leq 2\log(r/\epsilon)$. Hence this sufficient choice of scale, $\max\{\beta_0, (1/c)\log(2r/\epsilon)\}$, is $\Theta(\log(r/\epsilon))$. For $f_A = \exp$ and every $s < t$ the ratio is $e^{-(t-s)\beta}$, so exp is exponentially separating with $c = t - s$. For $f_A = \mathrm{sigmoid}$ and every $s < 0 < t$ the ratio is $(1 + e^{-\beta t})/(1 + e^{-\beta s})$, whose numerator is at most 2 and whose denominator exceeds $e^{|s|\beta}$. So the ratio is at most $2e^{-|s|\beta}$. This bound is at most $e^{-|s|\beta/2}$ once $\beta \geq (2\ln 2)/|s|$, so sigmoid is exponentially separating with $c = |s|/2$. Two special cases give an explicit scale. For $f_A = \exp$ with $s = 0$ and $t = 1$, the ratio is $e^{-\beta}$. For $f_A = \mathrm{sigmoid}$ with $s = -1$ and $t = 1$, the ratio is at most $2e^{-\beta}$. So the scale $\beta = \log(4r/\epsilon)$ serves both functionals, since it makes $2re^{-\beta} = \epsilon/2$ and $2r \cdot 2e^{-\beta} = \epsilon$. This scale is positive because $r \geq 2$ and $\epsilon \leq 1$.

It remains to read off the size of the model. The width is $d_0 + 3$. The embedded tokens have squared norm $ad_0 + 2\beta + 1 = 2\beta(t - s + 1) + 1 = O(\log(r/\epsilon))$. This is within the norm convention of Section 2.1 whenever $1/\epsilon$ is at most quasi-polynomial in n. The entries of the query, key and value maps are bounded by constants depending on s, t and $f_A(0)$ alone. □

Proof of Theorem A.4. The proof gives the two constructions in turn. The first has an explicit feature map and width $p\lceil N^{1/p}\rceil + 1$. The second reaches width $d_p(N)+1$ through an interpolation argument, and we then show that this width is optimal and, when $N \geq (2e)^p$, is $\Theta(pN^{1/p})$.

The explicit construction. Put $m = \lceil N^{1/p}\rceil$. Since $m^p \geq N$, there is an injection $\gamma : [N] \to [m]^p$. The injection gives each symbol $k \in [N]$ a string of p digits in $[m]$, which we write $\gamma(k) = (\gamma_1(k), \ldots, \gamma_p(k))$. Embed a key-value position with key k and payload v as the concatenation of the p one-hot vectors $e_{\gamma_1(k)}, \ldots, e_{\gamma_p(k)} \in \mathbb{R}^m$, followed by v. Embed the query as the same concatenation for q, followed by 0. The width is $pm + 1$. Let $W_Q = W_K$ be the projection onto the pm positional coordinates. Define the feature map $\phi : \mathbb{R}^{pm} \to \mathbb{R}^{m^p}$ as follows. It sends a vector u, written in blocks $u^{(1)}, \ldots, u^{(p)}$, to the vector of products

$$\phi(u) = \left(u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(p)}_{i_p}\right)_{i\in[m]^p}.$$

Every coordinate of this feature map is a monomial of degree exactly p. Its inner products factor over the blocks:

$$\langle \phi(u), \phi(u')\rangle = \sum_{i\in[m]^p}\prod_{t=1}^{p} u^{(t)}_{i_t} u'^{(t)}_{i_t} = \prod_{t=1}^{p}\left\langle u^{(t)}, u'^{(t)}\right\rangle.$$

So on one-hot blocks the score between the query q and a key-value position with key k is $\prod_{t=1}^{p}\mathbf{1}[\gamma_t(q) = \gamma_t(k)] = \mathbf{1}[q = k]$, the last step because γ is injective. This score therefore selects the position whose key is the query. Take $Z \equiv 1$, and let W_V be the projection onto the value coordinate. At the query position, the layer then outputs exactly the payload of the position whose key is q, since the query's own summand contributes its value coordinate 0. The readout is the identity. The width is $pm + 1 = p\lceil N^{1/p}\rceil + 1$.

The construction at width $d_p(N) + 1$. Write $d' = d_p(N)$. By the definition of $d_p(N)$, the real polynomials of degree at most p in d' variables form a space of dimension $\binom{d'+p}{p} \geq N$. Pick N

linearly independent ones, $P_1, \ldots, P_N$. We first find interpolation points: points $a_1, \ldots, a_N \in \mathbb{R}^{d'}$ for which the matrix $B = [P_i(a_k)]_{i,k}$ is nonsingular. They exist by induction on N. In the base case $N = 1$, the polynomial P_1 is nonzero by independence, so some point a_1 has $P_1(a_1) \neq 0$. For the step, suppose the points already chosen make the leading $(N-1) \times (N-1)$ minor nonsingular. The determinant of B, as a function of the last point a_N, is a combination of $P_1, \ldots, P_N$ in which P_N has the nonzero minor as coefficient. Hence this determinant is a nonzero polynomial in a_N, which is nonzero somewhere. A nonzero polynomial vanishes on no nonempty open set. So each point a_k may be chosen in the open cube $(0, 1/(N+1))^{d'}$, which gives $\|a_k\|^2 < d'/(N+1)^2 < 1$, as $d' = d_p(N) \leq N - 1$.

From the interpolation points we build a feature map. Let $(L_1, \ldots, L_N) = B^{-1}(P_1, \ldots, P_N)$. Then $L_i(a_k) = \mathbf{1}[i = k]$, so each L_i is 1 at the point a_i and 0 at the others. The map $\phi = (L_1, \ldots, L_N)$ is a feature map of degree at most p, and

$$\langle \phi(a_q), \phi(a_k) \rangle = \sum_i \mathbf{1}[i = q]\mathbf{1}[i = k] = \mathbf{1}[q = k].$$

Embed a key-value position with key k and payload v as (a_k, v), and embed the query q as $(a_q, 0)$. Let $W_Q = W_K$ project onto the first d' coordinates, let W_V project onto the last, and take $Z \equiv 1$. The output at the query position is then the payload of the position whose key is q, since the query's own summand contributes 0. The width is $d' + 1$, and the embedded tokens have norm at most 2.

Optimality and the size of $d_p(N)$. The width is optimal. Indeed, Theorem A.3(a) gives $\binom{d-1+p}{p} \geq N$ for every exact realization, and by the definition of $d_p(N)$ this says $d \geq d_p(N) + 1$. Now let $N \geq (2e)^p$. Then Theorem A.3(a), applied at width $d_p(N) + 1$, gives $d_p(N) \geq \frac{p}{2e} N^{1/p}$. In the other direction, with $m = \lceil N^{1/p} \rceil$ as in the explicit construction, we have

$$\binom{pm+p}{p} = \prod_{i=0}^{p-1} \frac{p(m+1) - i}{p - i} \geq (m+1)^p \geq N,$$

since each factor is at least $m + 1$. For the factor with index i, clearing the denominator turns this bound into the inequality $im \geq 0$. By the definition of $d_p(N)$, the bound $\binom{pm+p}{p} \geq N$ gives $d_p(N) \leq pm$. Last, $N^{1/p} \leq m \leq N^{1/p} + 1 \leq 2N^{1/p}$, as $N^{1/p} \geq 1$. So both widths, $d_p(N) + 1$ and $pm + 1$, lie between $\frac{p}{2e} N^{1/p}$ and $2pN^{1/p} + 1 \leq 3pN^{1/p}$. Hence both widths are $\Theta(pN^{1/p})$.

Both constructions realize Index_n as well. There the key of position j is j itself, supplied by the positional part of the embedding as Theorem 2.4(ii) allows. □

Proof of Theorem A.6. The proof has two parts. The first is an induction on the depth, which builds the cover and bounds the set of input positions that each representation depends on. The second derives the depth requirement from that bound.

We regard an input, for each value of its discrete coordinates, as a point of $\mathbb{R}^m$, where m is the number of its real coordinates, so that the input domain is a subset of $\mathbb{R}^m$. The cover built below covers all of $\mathbb{R}^m$, and hence the input domain. Nothing below depends on the discrete values.

The cover and the influence sets. We show by induction on the layer index ℓ that the input domain is covered by finitely many semi-analytic regions with the following property. On each region, the layer-ℓ representation at every position i is the restriction of a function that is real-analytic on the whole input space. This function depends on the input only through a set $I_\ell(i)$ of at most $(\kappa + 1)^\ell$ input positions, the influence set of position i at layer ℓ.

At $\ell = 0$ the cover is the single region $\mathbb{R}^m$. The embedding is real-analytic in the real content by Theorem 2.4(ii), and it reads position i alone, so $I_0(i) = \{i\}$.

For the inductive step, fix a region R of the cover at layer $\ell - 1$. Write F for the real-analytic map that gives the layer-$(\ell - 1)$ representation on R. By selectivity (Theorem A.5), the space of layer inputs is covered by finitely many semi-analytic regions R'. Each intersection $R \cap F^{-1}(R')$ is again semi-analytic. Indeed, composing the defining functions of R' with F, and conjoining the resulting conditions with those of R, gives a finite union of sets of the defining form. Over all R and R', these intersections cover the domain. On one such intersection, the active set at position i is a fixed set S_i

of at most κ positions, the positions that carry every nonzero attention weight. Each attention weight there is a real-analytic function of the layer inputs at i and at the positions of S_i. With the residual connection, the layer-ℓ representation at i is therefore a real-analytic function of the layer-$(\ell-1)$ representations at i and at the positions of S_i. Composed with F, it is a real-analytic function of the input, and its influence set satisfies

$$I_\ell(i) \subseteq I_{\ell-1}(i) \cup \bigcup_{j \in S_i} I_{\ell-1}(j), \qquad |I_\ell(i)| \le (\kappa+1) \max_j |I_{\ell-1}(j)| \le (\kappa+1)^\ell.$$

This completes the induction.

The token-wise readout makes the output at position i a function of the layer-L representation there, and hence of the positions of the influence set $I_L(i)$. When the readout is a feedforward network, this function need not be analytic. Reading the cover at layer L against Theorem 3.1 gives $\mathrm{PDeg}(A, L) \le (\kappa+1)^L$.

The depth requirement. Let U be a nonempty open subset of $\mathbb{R}^m$, contained in the input domain, such that at every input of U the task depends on all n positions. We find a nonempty open set inside one region of the cover, and then argue by contradiction. Write each region of the cover at layer L as a finite union of basic sets

$$\{f_1 = \cdots = f_a = 0 \text{ and } g_1 > 0, \ldots, g_b > 0\}.$$

Consider a basic set in which some f_i is not identically zero. It lies in the zero set of f_i, which is closed and, by the identity theorem on the connected space $\mathbb{R}^m$, has empty interior. Every other basic set is open. Suppose no open basic set met U. Then U would lie in a finite union of closed sets with empty interior, which has empty interior, contradicting $U \neq \emptyset$. So some open basic set B meets U. The intersection $U_0 = B \cap U$ is then a nonempty open set inside one region. On U_0 the output is a function of a set I of at most $(\kappa+1)^L$ positions.

Suppose $|I| < n$, and pick an input $x \in U_0$ and a position $j \notin I$. The task depends on j at x, and U_0 is a neighborhood of x. So some input $x' \in U_0$ differs from x at position j alone and carries a different target. The model, however, depends only on the positions of I on U_0, so it takes the same value at x and x'. An exact realization cannot do so, and hence $(\kappa+1)^L \ge n$, which is the stated depth requirement. □

Proof of Theorem A.7. The proof is the kernel argument of Theorem A.3(a), run on the state instead of the residual stream. The state after the keys is affine in the values, and a kernel vector of its linear part gives two inputs, with values v and $-v$, that the model cannot tell apart.

As in Section 2.3, split the embedding of a key-value position as $\iota(k_j, v_j) = u_j + v_j w$, with key part u_j and value direction w. Fix the keys $K = (k_1, \ldots, k_N)$. The transition maps R_t and the input maps E_t read the key channel, so they do not depend on the values v. Unrolling the recurrence over the N key positions gives the state after the keys:

$$h_N = \sum_{j=1}^{N} R_N \cdots R_{j+1} E_j (u_j + v_j w) = \text{baseline} + \Gamma v,$$
$$\Gamma = [R_N \cdots R_{j+1} E_j w]_{j=1}^{N} \in \mathbb{R}^{d_h \times N}.$$

The product $R_N \cdots R_{j+1}$ is empty at $j = N$. The baseline collects the terms in the key parts u_j, so it does not depend on v. Column j of the matrix Γ is the direction in which the value v_j moves the state. The final state $h_{N+1} = R_{N+1} h_N + E_{N+1} \iota(q)$ and the output are functions of h_N and the query token.

Suppose $d_h < N$. Then the $d_h \times N$ matrix Γ has a nonzero kernel vector v. Scale it so that $\max_j |v_j| = 1$, pick a position i with $|v_i| = 1$, and let the query be its key, $q = k_i$. The inputs (K, v, q) and $(K, -v, q)$ then produce the same state h_N, and hence the same output. Their targets v_i and $-v_i$, however, differ by 2. So one of the two errors is at least 1, which exceeds ϵ. Hence $d_h \ge N$.

For a streaming retrieval task of r values, let t_0 be the position of the first query token, and fix a key set at which the task is of retrieval type. Restrict to inputs whose values vanish outside the r

retrieval coordinates, all of which precede t_0. The argument runs on the state just before the first query token, because from t_0 on the transitions may read the query token. Unrolling the recurrence up to position $t_0 - 1$, as above, writes the state h_{t_0-1} as a baseline plus $\Gamma' v$. Here Γ' is the $d_h \times r$ matrix whose column for a retrieval coordinate j is $R_{t_0-1} \cdots R_{j+1} E_j w$.

If $d_h < r$, then the matrix Γ' has a nonzero kernel vector v, which we scale so that $\max_j |v_j| = 1$. The inputs with values v and $-v$ then have the same state h_{t_0-1} and the same tokens from t_0 on, hence the same output. Pick a retrieval coordinate i with $|v_i| = 1$, and the query whose coefficient vector is e_i. At that query the targets of the two inputs are 2 apart. So one of the two errors is at least 1, and hence $d_h \geq r$.

The argument uses nothing about the recurrence beyond the state being affine in the values. So it covers every transition and input map that reads the key channel alone. □

Proof of Theorem A.8. The proof extracts layer ℓ as a model of its own. By hypothesis, the input to layer ℓ is, position by position, a fixed function ι' of the token. So the extracted embedding ι', together with layer ℓ, is a depth-1 model with that layer's score functional, width and head count. Its output at the read position is the vector that the hypothesis names. Let g be the map from that vector to the model's output. Composing with g reproduces the model's output on every input. So the depth-1 model with readout g realizes T to the same error. Consider now a lower bound that holds for every function of the residual stream at the read position. It holds in particular for the readout g, provided the lower bound's other hypotheses, such as a split embedding, hold for the extracted embedding ι'. □

Proof of Theorem A.9. The proof of all three parts is a construction by recursive doubling. An accumulator at each position doubles the length of the window it sums at every doubling layer. Collect layers add the windows that the binary expansion of n selects into a collector at the read position. Part (b) replaces each hard layer by a softmax layer and bounds the accumulated error. Part (c) runs the doubling around a cycle of length n, and halves the values that the read position would otherwise count twice.

Part (a). We define the layers, prove by induction that the accumulators hold window sums, and then check that the collected windows tile $[n]$.

Let $k^* = \lfloor \log_2 n \rfloor$, and write n in binary as $n = \sum_{k=0}^{k^*} b_k 2^k$, so that the top bit is $b_{k^*} = 1$. For each bit index k, let the offset $o_k = \sum_{k'>k} b_{k'} 2^{k'}$ be the part of n above bit k. Embed position j as $(\cos(2\pi j/2n), \sin(2\pi j/2n), v_j, 0)$. The first two coordinates form a positional block, a point on a cycle of length $2n$. The third coordinate is an accumulator a, initialized to the value v_j. The fourth is a collector s, initialized to 0.

We next define a layer *at offset* $\Delta \geq 0$. Its query map W_Q rotates the positional block by the angle $-2\pi\Delta/2n$, its key map W_K fixes the positional block, and both annihilate a and s. The score between query i and key j is then $\cos(\pi(i-\Delta-j)/n)$. Suppose $i - \Delta \geq 1$. Every visible $j \neq i - \Delta$ has $0 < |i-\Delta-j| \leq n$, and hence a score strictly below 1, while $j = i - \Delta$ scores 1. So the unique argmax over the visible positions is $j = i - \Delta$, under either mask. The embedding is absolute and sinusoidal, and the rotation sits in W_Q. The scores are those that a rotary embedding at the same frequency gives, and this is the sense in which the statement calls the embedding rotary.

The model runs the following loop for $k = 0, 1, \ldots, k^* - 1$ in turn. If $b_k = 1$, it first applies a *collect* layer at offset o_k, whose value map W_V adds the attended position's accumulator a into the collector s. It then applies a *doubling* layer at offset 2^k, whose value map adds the attended position's a into a. The loop uses $k^* + \sum_{k<k^*} b_k \leq 2k^* = D$ layers.

Write $\mathrm{win}(i, m) = \sum_{j=i-m+1}^{i} v_j$ for the window sum of length m ending at i, defined when $i \geq m$. We claim, by induction on k, that when the loop reaches step $k \leq k^*$, every position $i \geq 2^k$ holds the window sum $a_i = \mathrm{win}(i, 2^k)$. At $k = 0$ this is the initialization. For the step, the doubling layer at offset 2^k updates the accumulator at every $i \geq 2^{k+1}$ as

$$a_i \leftarrow \mathrm{win}(i, 2^k) + \mathrm{win}(i - 2^k, 2^k) = \mathrm{win}(i, 2^{k+1}),$$

both windows being defined there. Positions with $i - \Delta < 1$ attend wherever their argmax falls, ties broken by the fixed rule of Section 2.1. The values they hold are never read by anything that the read

position depends on. Collect layers leave the accumulator a untouched, and doubling layers leave the collector s untouched.

It remains to follow the read position n. Consider the collect layer for a bit $b_k = 1$ with $k < k^*$. There position n attends to $n - o_k = \sum_{k' \le k} b_{k'} 2^{k'} \ge 2^k$. By the claim, that position holds the window sum $\mathrm{win}(n - o_k, 2^k)$, and the collect layer adds it into s_n. After the loop the accumulator at n is $a_n = \mathrm{win}(n, 2^{k^*})$, the window of the top bit, for which $o_{k^*} = 0$. The windows $\{n - o_k - 2^k + 1, \dots, n - o_k\}$ over the set bits k tile $[n]$. Indeed, the window of the top bit ends at n, and each next window ends one position before the previous one begins. Their lengths 2^k over the set bits k sum to n, so the last of them begins at position 1. So at the end $a_n + s_n = \sum_{j=1}^{n} v_j$, which the readout at position n returns. Every attended position is at most the attending one. So the schedule runs under the causal mask as it does under the bidirectional one. When $n = 2^{k^*}$, no bit below k^* is set, and the run is the $\log_2 n$ doubling layers. The collector s is then never written, and dropping it leaves width 3.

Part (b). We replace the positional block so that the target position wins the softmax by a margin, run the schedule of part (a), and bound by induction on the layers how far the softmax run drifts from the exact run.

Replace the positional block by one on $F = \lceil \log_2 n \rceil$ frequencies, with a scale λ fixed below. Position j is embedded as $\sqrt{\lambda}(\cos(2\pi j/2^f), \sin(2\pi j/2^f))_{f=1}^{F}$, followed by the accumulator a and the collector s. A layer at offset Δ now has its query map W_Q rotate the f-th pair by $-2\pi\Delta/2^f$. Its key map W_K fixes the positional block, and both maps annihilate a and s. The score between query i and key j is then

$$\lambda \sum_{f=1}^{F} \cos\left(2\pi(i - \Delta - j)/2^f\right).$$

At the target $j = i - \Delta$ every cosine is 1, and the score is λF. For $j \ne i - \Delta$, write $\delta = i - \Delta - j$, so that $0 < |\delta| < n \le 2^F$. Let 2^ν be the largest power of two dividing δ, so that $\nu < F$. At $f = \nu + 1$ the angle $2\pi\delta/2^{\nu+1}$ is an odd multiple of π, so the cosine is -1, and the score is at most $\lambda(F - 2)$. The target's margin is therefore at least 2λ. Write θ for the softmax weight on the target position. By the margin, $\theta \ge 1/(1 + ne^{-2\lambda}) \ge 1 - ne^{-2\lambda}$. At each position, the layer adds to the accumulator a convex combination of the attended coordinate over the visible positions.

Run the schedule of part (a) with these layers, and compare with the exact run. In the softmax run each layer at most doubles the largest accumulator magnitude, since it adds a convex combination of existing values. So every accumulator coordinate has magnitude at most $B = 2^D \le n^2$. Call a coordinate valid if the exact run reads it. The valid coordinates are a_i at positions $i \ge 2^k$ when the loop reaches step k, and s_n throughout. After the t-th layer, we judge validity at the loop step that the run is then in. The doubling layer of step k ends that step, so after it the run is in step $k + 1$. Let ξ_t bound the deviation of every valid coordinate from its exact value after t layers, so that $\xi_0 = 0$.

We bound ξ_{t+1} in terms of ξ_t. A valid coordinate at the next layer is its own previous value plus the convex combination that the layer adds. The coordinate that the combination reads at the target position is valid. So the deviation of the valid coordinate is at most the sum of three terms. The first is ξ_t, for its own value. The second is ξ_t, for the target's value. The third is $(1 - \theta)$ times the largest gap between two visible values, and that gap is at most $2B$. Since $1 - \theta \le ne^{-2\lambda}$ by the margin and $2B \le 2n^2$, the third term is at most $2n^3 e^{-2\lambda}$. Hence

$$\xi_{t+1} \le 2\xi_t + 2n^3 e^{-2\lambda}, \qquad \text{and so} \qquad \xi_t \le (2^t - 1) \cdot 2n^3 e^{-2\lambda},$$

the second by induction on t, since $2(2^t - 1) + 1 = 2^{t+1} - 1$. The schedule has at most D layers, so after its last layer every valid coordinate deviates by at most $2^D \cdot 2n^3 e^{-2\lambda} \le 2n^5 e^{-2\lambda}$, as $2^D \le n^2$. The readout $a_n + s_n$ therefore deviates by at most twice that, which is at most $4n^5 e^{-2\lambda}$. Taking $\lambda = \frac{1}{2}\log(4n^5/\epsilon)$ makes that deviation at most ϵ. The width is $2F + 2$. The embedded tokens have squared norm $\lambda F + 1 = O(\log n \cdot \log(n/\epsilon))$, which is polylogarithmic in n when $\log(1/\epsilon)$ is.

Part (c). We define the embedding and the layers, prove by induction what each accumulator holds, and then count how often the read position sees each value.

Let $L = \lceil \log_2 n \rceil$. By the choice of L, the length satisfies $n \le 2^L < 2n$. Call the positions $2n - 2^L + 1, \dots, n$ the repeated positions. There are $2^L - n$ of them, and there are none when n is a

power of two. Embed position j, carrying the value v_j, as the vector $(\cos(2\pi j/n), \sin(2\pi j/n), c_j v_j)$ of $\mathbb{R}^3$. The weight c_j is $1/2$ at a repeated position and 1 at every other position. The first two coordinates form the positional block, and the third coordinate is the accumulator.

The model has L layers. At layer ℓ, the query matrix W_Q rotates the positional block by the angle $-2\pi 2^{\ell-1}/n$, the key matrix W_K leaves it fixed, and both annihilate the accumulator. The score between query i and key j is then $\cos(2\pi(i - 2^{\ell-1} - j)/n)$. The score is 1 when $j \equiv i - 2^{\ell-1} \pmod n$ and strictly below 1 otherwise, and exactly one $j \in [n]$ has that residue, so the argmax is unique. The value matrix W_V copies the accumulator of the attended position, so the residual connection adds it to the accumulator at i and leaves the positional block unchanged. As before, the embedding is absolute and sinusoidal and the rotation sits in W_Q, which is the sense in which the statement calls it rotary.

Stepping back m places around the cycle from a position i lands on one position of $[n]$, and we write $j(i, m)$ for it. In symbols, $j(i, m)$ is the position congruent to $i - m$ modulo n. Write $a_i^{(\ell)}$ for the accumulator at position i after layer ℓ. We show by induction on ℓ that

$$a_i^{(\ell)} = \sum_{m=0}^{2^\ell - 1} c_{j(i,m)} v_{j(i,m)}.$$

At $\ell = 0$ the sum has the single term $c_i v_i$, which is the embedded accumulator. At layer $\ell \geq 1$, position i adds the accumulator of the position $j(i, 2^{\ell-1})$. By the induction hypothesis, that accumulator is the sum of $c_j v_j$ over the positions $j(i, 2^{\ell-1} + m')$ with $0 \leq m' < 2^{\ell-1}$. Added to the sum over $m < 2^{\ell-1}$ that position i already holds, it gives the sum over $m < 2^\ell$, which completes the induction. Unlike the windows of part (a), these windows wrap around the cycle and may visit a position more than once.

Now read the accumulator at the read position $i = n$ after the last layer, $\ell = L$. The indices $m = 0, \ldots, n-1$ give the positions $n, n-1, \ldots, 1$, each once. There are $2^L - n < n$ remaining indices $m = n, \ldots, 2^L - 1$, and since $n \leq m < 2n$ each gives the position $j(n, m) = 2n - m$. These run over $2n - 2^L + 1, \ldots, n$, which are exactly the repeated positions, each once more. So each repeated position is counted twice with weight $1/2$, every other position is counted once with weight 1, and $a_n^{(L)} = \sum_{j=1}^n v_j$. The readout returns the accumulator. When n is a power of two, there are no repeated positions and every weight is 1. The width is 3, and the embedding is real-analytic in the value. The windows wrap around the cycle, so intermediate positions attend to later ones, and the construction needs the bidirectional mask.

Both tasks, Count_n and Avg_n, depend on every position at every input of the open set $(0, 1)^n$. Indeed, each value v_j enters the target with coefficient 1 or $1/n$. So Theorem A.6 with $\kappa = 1$ gives $2^{L'} \geq n$ for every hard attention model that realizes Count_n or Avg_n exactly, whatever its depth L' and its mask. As L' is an integer, this says $L' \geq \lceil \log_2 n \rceil$, so the depth $\lceil \log_2 n \rceil$ of part (c) is the least possible, at every n. □

Proof of Theorem 4.2. The proof takes the three parts in order. Part (i) is proved one axis at a time, each by exhibiting two inputs that the architecture cannot separate or an output range that it cannot reach. On each axis we bound the invariant at the resource setting of the realizing model. That setting has cost at most C. So it is one of the settings over which Theorem 3.2 takes the extreme value $\mathcal{I}(A; C)$, and the bound passes to $\mathcal{I}(A; C)$. Part (ii) computes the invariant profiles of the four canonical tasks and then, row by row, gives a realization wherever the five gates are met. Part (iii) combines the two parts.

(i), the KRank *axis.* At $L = 1$, suppose the kernel of A factors through a feature space of dimension $m = \mathrm{KRank}(A) < \infty$. When $\mathrm{KRank}(A) = \infty$ there is nothing to prove. The kernel matrix at every key set then has rank at most m. With this rank bound, the argument of Theorem A.3(a) goes through: it finds two value vectors v and $-v$ with the same residual stream at the query position. For exact realization, that argument gives $m \geq \mathrm{KRank}(T)$. At every $\epsilon > 0$ it gives the same bound when T is of retrieval type. At $\epsilon > 0$ every member of $\mathcal{T}_{\mathrm{KRank}}$ is of retrieval type (Theorem 4.1), so the bound holds on the whole class.

With $H > 1$ heads of a polynomial functional of degree p, the same argument runs on the matrix stacking the H kernel matrices. Since $W_K^{(h)} w = 0$, each key of head h is a linear function of the

key part $u_j \in \mathbb{R}^d$ of the embedding. So each row of the stacked matrix is, as a function of the key position, a polynomial of degree at most p in the key part u_j. Hence the stacked matrix has rank at most $\binom{d+p}{p} = \mathrm{KRank}(A)$.

At $L > 1$ we reduce to depth 1. By Theorem A.8, a layered realization contains a depth-1 model of the same width and head count whose readout is a function of the residual stream at the read position. The embedding of this depth-1 model is split, by the hypothesis of part (i). The argument above uses nothing about the readout, so the bound transfers.

(i), the PDeg *axis.* By the definition of $\mathrm{PDeg}(A, L)$, the realizing model admits a finite cover of its input domain by semi-analytic regions with the following property: on each region, its output at the read position is a function of at most $\mathrm{PDeg}(A, L)$ positions. Let U be the interior of the set of inputs at which T depends on at least $\mathrm{PDeg}(T)$ positions. This interior is nonempty by the definition of $\mathrm{PDeg}(T)$. As in the proof of Theorem A.6, some open basic set of the cover meets U in a nonempty open set U_0 that lies in one region. On U_0 the model is a function of a set I of at most $\mathrm{PDeg}(A, L)$ positions. Suppose $|I| < \mathrm{PDeg}(T)$. Then at every input x of U_0 the target depends on some position $j \notin I$. So some input in U_0 differs from x at j alone and carries a different target. The model output at that input is the same as at x, which an exact realization cannot allow. Hence $\mathrm{PDeg}(A, L) \geq \mathrm{PDeg}(T)$. Theorem D.14 says why the argument stops at exact realization.

(i), the Bound *axis.* The realizing model is within the norm convention of Section 2.1. So its output magnitude is at most $\mathrm{Bound}(A, L)$, by definition. The task T, on the other hand, attains magnitude arbitrarily close to $\mathrm{Bound}(T)$. Realization to error ϵ therefore forces $\mathrm{Bound}(A, L) \geq \mathrm{Bound}(T) - \epsilon > \mathrm{Bound}(T) - 1/2$. This is the inequality on this axis, as Theorem 3.2 reads it.

For a normalized row, induction on the layers bounds the largest token norm at depth L by $(1 + Hc_V)^L$ times the largest embedded norm. The readout adds at most a constant factor and a polylogarithmic offset. So $\mathrm{Bound}(A, L) \leq c(1 + Hc_V)^L P(n) + P(n)$, where c is the readout's constant. A length-uniform realization, by one model at every length, fixes the depth L. So a normalized row realizes no member of $\mathcal{T}_{\mathrm{Bound}}$ length-uniformly; only an unnormalized row, with $\mathrm{Bound}(A, 1) \geq n$, can.

(i), the Sym *axis.* Let $\Pi = \mathrm{Sym}(A)$, a group of permutations of the positions other than the read position ρ. We show that the output at the read position is Π-invariant, and then argue by contradiction. By the definition of $\mathrm{Sym}(A)$, the model, embedding and layers together, is Π-equivariant, and the readout is token-wise. So for every permutation $\pi \in \Pi$ the output satisfies

$$\mathcal{M}(\pi x)_\rho = \mathcal{M}(x)_{\pi^{-1}(\rho)} = \mathcal{M}(x)_\rho.$$

That is, the output at the read position is Π-invariant. Suppose some permutation $\pi \in \Pi$ lies outside $\mathrm{Sym}(T)$. The definition of $\mathcal{T}_{\mathrm{Sym}}$ then gives $\sup_x |T(x) - T(\pi x)| \geq 1 > 2\epsilon$. So some input x has $|T(x) - T(\pi x)| > 2\epsilon$. The model gives one and the same output value at x and at πx, and that value cannot be within ϵ of both targets. Hence $\mathrm{Sym}(A) \subseteq \mathrm{Sym}(T)$, which is the inequality in the orientation of Theorem D.3.

(i), the State *axis.* For a feedforward architecture $\mathrm{State}(A) = \infty$, and there is nothing to prove. By Theorem 2.4(iv), a recurrent architecture is a single recurrent layer, and for it this axis is Theorem A.7. The kernel argument of that lemma shows that a recurrence of state dimension d_h realizing a streaming retrieval task of r values to error $\epsilon < 1$ has $d_h \geq r$. That argument uses two properties, both part of the definition of $\mathcal{T}_{\mathrm{State}}$: the split embedding, and the order of the retrieval coordinates before the first query token. The lemma is stated for tasks of retrieval type, and at $\epsilon > 0$ every member of $\mathcal{T}_{\mathrm{State}}$ is of retrieval type.

It remains to treat exact realization of a task that is not of retrieval type. There the argument of Theorem A.3(a) runs with the kernel matrix G_K replaced by the matrix Γ of the proof of Theorem A.7. Here Γ has a column for each position before the first query token. If $d_h < r$, the argument gives a nonzero v with $\Gamma v = 0$, supported on a set of r columns on which the coefficient matrix is nonsingular, as in the proof of Theorem A.3(a). Set every value after the first query token to 0. The inputs with values v and $-v$ then have the same state before the first query token and the same tokens after it. Hence they have the same output, while some query separates their targets.

(ii). We first compute the profiles of the four canonical tasks, then read the gates row by row, and last give the realization that meets each gate. The task profiles, with one row for each task and one

column for each axis, are

	KRank	PDeg	Bound	Sym	State
AR_N	N	1	1	S_n	N
Index_n	n	1	1	$\{e\}$	n
Avg_n	1	n	1	S_n	1
Count_n	1	n	n	S_n	1

.

In the Sym column, the symbol S_n stands, as in Theorem 3.1, for the group of all permutations of the positions other than the read position. For AR_N those positions are the N key-value positions, and for Index_n they are the n value positions. For Avg_n and Count_n they are the positions $1, \ldots, n-1$.

The entries come from the definitions as follows. For AR_N and Index_n, the coefficient family is the full set of selection vectors, the target depends on one position at every input, and the streaming rank equals the rank. These two tasks differ only on the symmetry axis. The task AR_N carries its key as token content, and so it is invariant under every permutation of the positions other than the read position. By contrast, Index_n addresses by position, and every permutation other than the identity changes its target, as the proof of Theorem 4.3 shows. For Avg_n and Count_n, the coefficient family is one vector, the target depends on all n positions at every input, and the streaming rank is 1. Strictly, Count_n is not a linear-retrieval task, since its values range over $[0, 1]$ rather than $[-1, 1]$. Its target is still the single linear functional $\sum_j v_j$, whose one coefficient vector has rank 1. On both the KRank and State axes we give Count_n that rank.

Now take the rows in turn, starting with softmax, sigmoid, hard, average-hard and NoPE softmax. For these rows the KRank, PDeg and State entries are ∞, except for $\mathrm{PDeg} = 2^L$ at hard attention. The three tasks AR_N, Index_n and Avg_n have output range 1. On them the Bound gate asks only $\mathrm{Bound}(A; C) > 1/2$, and every row meets it at depth 1. Indeed, each row realizes AR_N at depth 1 to an error below $1/2$, as below, and so outputs more than $1/2$ at an input whose requested payload is 1. So on Index_n the only gate with content is Sym, which excludes NoPE and admits the rest. On AR_N no gate has content for these rows, and every one of them realizes it, NoPE included. On Avg_n the only gate with content is PDeg, which forces $2^L \geq n$ at hard attention and is free elsewhere.

The corresponding realizations are as follows. For softmax and sigmoid on AR_N and Index_n, and for NoPE softmax on AR_N, the realization is Theorem A.3(b). Its width is $O(\log N)$ on AR_N and $O(\log n)$ on Index_n. On AR_N that construction keys on token content rather than on position, and so it survives the removal of the positional component. For hard and average-hard, the same embedding with the argmax in place of the kernel is exact, because the matching key is the unique maximizer. On Index_n the key of position j is j itself, supplied by the positional part of the embedding, and the scores are those of the proof of Theorem A.3(b) verbatim.

On Avg_n, uniform scores give the realization for softmax, average-hard and NoPE softmax, at width $O(1)$ and depth 1. With the query map zero, every position gets weight $1/n$. The value map copies v_j into a coordinate that the embedding leaves at 0. Sigmoid attention has normalizer 1, so with the query map zero every score is $\mathrm{sigmoid}(0) = 1/2$. A value map that copies v_j with the factor $2/n$ then outputs exactly the mean $\bar{v}$. This value map has operator norm $2/n$, which is within the norm convention. For hard attention on Avg_n the realization is Theorem A.9(a), at depth $O(\log n)$ and width 4.

On Count_n two gates have content. The Bound gate asks a normalized row for depth $\Omega(\log(n/P(n)))$. For hard attention, the PDeg gate asks $2^L \geq n$. Both gates are met at cost $O(\log n)$. The normalized rows realize Count_n at depth $O(\log n)$: softmax, average-hard and NoPE softmax by the doubling construction below, and hard attention by Theorem A.9(a). The unnormalized rows, polynomial-linear and sigmoid, have $\mathrm{Bound}(A, 1) \geq n$. They realize Count_n exactly at depth 1 and width $O(1)$ by giving every position the same score. For polynomial-linear attention, the query and key maps copy a constant coordinate 1 of the embedding, and the feature map takes the p-th power of that coordinate. So every score is 1, and a value map that copies v_j outputs $\sum_j v_j$. With the factor $1/n$ in the value map, the same model realizes Avg_n. For sigmoid attention every score is $\mathrm{sigmoid}(0) = 1/2$ as above, and the value map carries the factor $2 = 1/\mathrm{sigmoid}(0)$, the bound $1/f_A(0)$ that the unnormalized case of Theorem A.3(b) puts on the value map, evaluated at $f_A = \mathrm{sigmoid}$.

For polynomial-linear attention of degree p, the KRank gate has content on AR_N and Index_n. It asks $\binom{d+p}{p} \geq N$ on AR_N and $\binom{d+p}{p} \geq n$ on Index_n. The matching realization is Theorem A.4, at width $d_p(N)+1$ on AR_N and $d_p(n)+1$ on Index_n.

Causal softmax is read at the last position, which the causal mask lets see every input position. There the retrieval constructions and the uniform-score construction apply unchanged. On Count_n the realization is Theorem A.9(b), at depth $O(\log n)$ and width $O(\log n)$, to every error $\epsilon \geq e^{-\mathrm{polylog}(n)}$.

Next come the two state-space rows. Each is a single recurrence by Theorem 2.4(iv), and each is read against the causal softmax row. On AR_N their State gate asks $d_h \geq N$, and on Index_n it asks $d_h \geq n$. On Avg_n and Count_n it asks $d_h \geq 1$, and one accumulator meets this last gate. Take the recurrence with $R_t = 1$ and with E_t reading the value coordinate. After position t its state holds $\sum_{j \leq t} v_j$. The readout returns h_n on Count_n and h_n/n on Avg_n, and both realizations are exact.

The selective row meets the retrieval gate by storing each payload in the coordinate of its key. Take the key part u_j orthogonal to the value direction w, and carry the key as k_j/N in u_j. At the key positions set $R_t = I$ and $E_t = e_{k_t} w^\top / \|w\|^2$, which are functions of the key channel. The state after the keys is then $\sum_j v_j e_{k_j}$. At the query set $E_{N+1} = 0$ and $R_{N+1} = e_1 e_q^\top$, a function of the query token. This moves the payload of key q into the first coordinate, which is read out linearly. The state dimension is N, and the width is $O(1)$.

The linear row has fixed maps. For it, a shift register holds the last $N+1$ tokens and writes them into the residual stream. The register holds each token as the binary expansion of its key minus one, in $\lceil \log_2 N \rceil$ coordinates, together with its payload. The residual stream therefore has width $d = \Theta(N \log N)$, and the readout has width $O(N \log N) = O(d)$, as Theorem 2.4(iii) asks. For each slot, the readout computes the indicator $b = \mathrm{relu}(1 - \sum_\ell |x_\ell - y_\ell|)$ that the slot's key bits x match the query's bits y, using $|z| = \mathrm{relu}(z) + \mathrm{relu}(-z)$. It then returns

$$\sum_{\text{slots}} \left[\mathrm{relu}(v + 2b - 2) - \mathrm{relu}(-v + 2b - 2) \right],$$

which is the payload of the slot whose key matches, the query's own slot contributing 0. On the vectors the model produces, this output is at most their largest coordinate, as the norm convention asks. The cost is $O(N^2 \log^2 N)$. For Index_n both state-space rows use the same constructions, with the key of a position being the position itself.

The matching construction for Count_n under the bidirectional mask is the doubling one, which we now give. Keep the value in one coordinate and an accumulator in a second, and initialize the accumulator to v_i. Let every layer attend uniformly. Then, by induction on the layer index ℓ, the accumulator at position i after layer ℓ is $a_i^{(\ell)} = v_i + (2^\ell - 1)\bar{v}$, where $\bar{v}$ is the mean of the values. The inductive step is

$$\begin{aligned} a_i^{(\ell)} &= a_i^{(\ell-1)} + \frac{1}{n} \sum_j a_j^{(\ell-1)} \\ &= v_i + (2^{\ell-1} - 1)\bar{v} + \bar{v} + (2^{\ell-1} - 1)\bar{v}. \end{aligned}$$

At depth $L = \lceil \log_2(n+1) \rceil$ the coefficient $2^L - 1$ is at least n. The token-wise readout sees both coordinates, and it returns $\frac{n}{2^L - 1}(a_i^{(L)} - v_i) = \sum_j v_j$.

The realizations above are exact except in these cells: the retrieval cells of the softmax, sigmoid and causal softmax rows, the AR_N cell of NoPE softmax, and the Count_n cell of causal softmax. In those cells they realize the task to every error $\epsilon \geq e^{-\mathrm{polylog}(n)}$. This is how the tolerance clause of part (ii) is to be read.

Part (ii) asks, at every budget C at which the gates clear, for a realization at cost $\tilde{O}(C)$. Every budget is at least 1, so a realization at polylogarithmic cost meets this demand at every budget. Every cell but seven has a realization of that kind. In one of the seven, the Index_n cell of NoPE softmax, the Sym gate fails at every budget, so part (ii) asks nothing there. The other six are the retrieval cells of the polynomial row and of the two state-space rows, and there the gate bounds the budget from below. On the polynomial row, $\mathrm{KRank}(A; C) = \binom{\lfloor \sqrt{C} \rfloor + p}{p} \geq N$ forces $\sqrt{C} \geq d_p(N)$, against the cost $(d_p(N)+1)^2 \leq 4 d_p(N)^2$ of Theorem A.4. On the state-space rows, $\mathrm{State}(A; C) \geq N$ forces

$d_h \geq N$ and so $C \geq N^2$, against the costs $O(N^2)$ and $O(N^2 \log^2 N)$ above. On Index_n the same holds with n in place of N. With part (i), therefore, in every cell the five gates clear if and only if the task is realizable at the stated cost, for the realizations to which part (i) applies on every axis. The proof of part (iii) below makes that phrase precise.

(iii). We check the inequality for the cheaper row A_1 axis by axis, and then apply part (ii) to the other row A_2. The row A_1 realizes T at cost C_1 by a realization to which part (i) applies on every axis. We read the phrase "a realization to which part (i) applies on every axis" as follows. It means a realization to which part (i) applies on every axis where the row's invariant can fall below the task's. Concretely, the realization is exact when the row's piecewise degree at the realization's depth is below n. It is layered, with a split extracted layer, when the row has finite kernel rank and depth above 1. Recall that a recurrent row is a single recurrence, by Theorem 2.4(iv).

With this reading, $\mathcal{I}(A_1; C_1) \geq \mathcal{I}(T)$ holds on every axis, as follows. On the Bound axis, the argument of part (i) applies to every task. Where T lies in the class of the KRank, Sym or State axis, the inequality on that axis is part (i). Where it does not, $\mathcal{I}(T)$ is 1, S_n and 1 on these three axes respectively, which every row meets. On the PDeg axis, part (i) applies to an exact realization, while a row with $\mathrm{PDeg} = n$ meets $\mathrm{PDeg}(T) \leq n$.

It remains to apply part (ii) to the row A_2. The claim that A_2 requires cost $\omega(\tilde{O}(C_1))$ is a lower bound, and by the convention of Section 2.2 it holds for realization to every error below $1/2$, exact realization included. Let $g(n)C_1$ be the cost at which part (ii) realizes T on A_2 whenever all five gates of A_2 clear at C_1. The factor g is polylogarithmic. By hypothesis, there is an n_0 such that for every $n \geq n_0$ the row A_2 does not realize T at cost $g(n)C_1$ at the tolerance of part (ii). For every such n, part (ii), contraposed, gives an invariant with $\mathcal{I}(A_2; C_1) < \mathcal{I}(T)$. □

Proof of Theorem 4.3. The proof has two halves: the failure on Index_n, which comes from the Sym axis, and the realizations of the other three tasks, which come from part (ii) of Theorem 4.2.

NoPE softmax with a bidirectional mask has $\mathrm{Sym}(A) = S_n$. Its embedding reads the content of a token and not its position. A bidirectional softmax layer gives position i the vector x_i plus the average of the vectors $W_V x_j$ over all positions j, with weights proportional to $\exp\langle q_i, k_j\rangle$. Permuting the input positions therefore permutes the outputs in the same way, and the readout is token-wise.

For Index_n, with $n \geq 2$, the key is the position. Let π be a permutation of the value positions that is not the identity. It acts on an input x so that position i of the permuted input πx carries the value $v_{\pi^{-1}(i)}$. Pick a query i^* with $\pi^{-1}(i^*) \neq i^*$. Consider the input x with $v_{i^*} = 1$, with $v_{\pi^{-1}(i^*)} = -1$, and with every other value 0. Its target is 1, while the target at the permuted input πx is -1. So every permutation other than the identity changes the target by 2 at some input. This gives $\mathrm{Sym}(\mathrm{Index}_n) = \{e\}$, and it places Index_n in the class $\mathcal{T}_{\mathrm{Sym}}$ of Theorem 4.1, as part (i) requires. Part (i) of Theorem 4.2, on the Sym axis, forces $\mathrm{Sym}(A) \subseteq \mathrm{Sym}(T)$, and this inclusion fails. The argument of part (i) bounds no resource, so no budget repairs the failure. The failure needs only $\mathrm{Sym}(\mathrm{Index}_n) \neq S_n$, which a single transposition already gives.

For the other three tasks the target is S_n-invariant, so the Sym gate clears. The remaining four gates clear as in part (ii), which supplies the realizations. For AR_N the realization is the construction of Theorem A.3(b), which keys on token content and needs no positional component. For Avg_n the realization is by uniform scores, at width $O(1)$ and depth 1. For Count_n the realization is the doubling construction at depth $\lceil \log_2(n+1) \rceil$, which attends uniformly and is therefore equivariant. □

Proof of Theorem B.1. We bound the least realizing cost from below by part (i) of Theorem 4.2 and from above by part (ii). Write C^* for the least cost C with $\mathcal{I}(A; C) \geq \mathcal{I}(T)$ for all five invariants. When some budget clears all five gates, the least such cost exists, since each gate is monotone in C and the costs of Theorem 2.2 are integers. When no budget clears all five gates, as for NoPE softmax on Index_n, the least cost is $C^* = \infty$. Now take a realization of T by A, at cost C and error ϵ, to which part (i) applies on every axis, as in the statement. The proof of Theorem 4.2(iii) makes an axis-by-axis check for the row A_1, and that check applies to every realization to which part (i) applies on every axis. So this realization has $\mathcal{I}(A; C) \geq \mathcal{I}(T)$ for all five invariants. Hence its cost C is at least C^*.

Conversely, part (ii) realizes T on A at cost $\tilde{O}(C^*)$ to error ϵ, since ϵ lies within its tolerance and the cost of each of its realizations does not depend on the error. That realization is one to which part (i) applies on every axis. The hard row is the only row of part (ii) whose piecewise degree falls below n, and its realizations are exact. The polynomial row is the only one of finite kernel rank, and its realizations have depth 1. Each state-space realization is a single recurrence. The two bounds agree to within the polylogarithmic factor. □

Proof of Theorem B.2. The lower bound $d \geq N$ is the kernel argument of Theorem A.3(a), run on the kernel matrices of all H heads stacked together. The bound $d \geq H$ holds because the per-head width is a positive integer. The upper bounds are two constructions, one for a single head and one for H heads.

The lower bound. Fix a key set K. The value direction w is nonzero, since if $w = 0$ the output does not depend on the values. Let $J \in \mathbb{R}^{d\times(d-1)}$ be the basis matrix whose columns are an orthonormal basis of the orthogonal complement $w^{\perp}$. Every key map $W_K^{(h)}$ annihilates w, so $W_K^{(h)} = W_K^{(h)} JJ^{\top}$. At $p = 1$, consider the score of head h between the query and key position j, divided by that head's normalizer. It is an affine function of $J^{\top}u_j \in \mathbb{R}^{d-1}$, whose coefficients depend on the head, the query and the key set K. Now stack the H kernel matrices $G^{(h)}$ at K into one matrix, with rows indexed by a head and a query and columns indexed by the key positions. Every row of this stacked matrix lies in the span of the $d-1$ coordinate rows of $J^{\top}u_j$ and the all-ones row. So the stacked matrix has rank at most d.

We next sum the kernel decompositions of the proof of Theorem A.3 over the heads. The residual stream at the query position is then a fixed vector plus $\sum_h (O_h W_V^{(h)} w)(G^{(h)}v)_q$. Restricted to value vectors v in the common kernel of the kernel matrices $G^{(h)}$, this sum does not depend on v. Suppose $d < N$. Then the columns of the stacked matrix at the N key positions have rank less than N. So some nonzero v supported on the key positions lies in that common kernel. Scale it so that $\max_j |v_j| = 1$, and say $|v_i| = 1$. Let q be the key of position i. The inputs (K, v, q) and $(K, -v, q)$ then have equal outputs and targets 2 apart, contradicting $\epsilon < 1$. Hence $d \geq N$. Also $d \geq H$, because the per-head width d/H is a positive integer.

The constructions. At $H = 1$, Theorem A.4 at $p = 1$ realizes AR_N exactly at width $d_1(N)+1 = N$. Indeed, $\binom{d+1}{1} = d+1$, which gives $d_1(N) = N-1$. For general H, let d be any multiple of H that is at least $N+1$. The least such multiple is at most $N+H$. Then $d/H \geq \lceil N/H \rceil$, since d/H is an integer larger than N/H. Partition $[N]$ into H blocks $B_1, \ldots, B_H$ of size at most $\lceil N/H \rceil$. Embed a key-value position with key k and payload v as the one-hot vector $e_k \in \mathbb{R}^N$, followed by v and zeros. Embed the query q as the one-hot vector e_q, followed by zeros. Let head h project queries and keys onto the coordinates of its block B_h, which fit in its per-head width d/H. Its score between the query and a key-value position with key k is then $\mathbf{1}[q = k]\mathbf{1}[k \in B_h]$. Let every head read the value coordinate and let O_h place it in a common output coordinate. Exactly one head has q in its block, and the query's own summand contributes its value coordinate 0. So the additive aggregate is the payload of the position whose key is q. □

Proof of Theorem B.3. We bound the ratio of the two widths from above and from below, show that both bounds equal $\frac{p}{e(p+1)} N^{1/(p(p+1))}$ up to factors that tend to 1, and then make the strictness explicit.

The width upper bound $(p+1)\lceil N^{1/(p+1)} \rceil + 1$ at degree $p+1$ is Theorem A.4 at degree $p+1$. The width lower bound $(p/e)N^{1/p} - p$ at degree p follows from Theorem A.3(a) at degree p. Write $R_p(N)$ for the ratio of the lower bound to the upper bound:

$$R_p(N) = \frac{(p/e)N^{1/p} - p}{(p+1)\lceil N^{1/(p+1)} \rceil + 1}.$$

For the upper estimate, drop the $-p$ from the numerator, and the $+1$ and the ceiling from the denominator. Using $1/p - 1/(p+1) = 1/(p(p+1))$, this gives

$$R_p(N) \leq \frac{(p/e)N^{1/p}}{(p+1)N^{1/(p+1)}} = \frac{p}{e(p+1)} N^{1/(p(p+1))}.$$

For the lower estimate, let $N \geq e^p$, so that $1 - eN^{-1/p} \geq 0$. Factor $(p/e)N^{1/p}$ out of the numerator. In the denominator, first use $\lceil x \rceil \leq x + 1$, and then factor out $(p+1)N^{1/(p+1)}$. This gives

$$R_p(N) \geq \frac{(p/e)N^{1/p}\left(1 - eN^{-1/p}\right)}{(p+1)N^{1/(p+1)}\left(1 + \frac{p+2}{p+1}N^{-1/(p+1)}\right)} = \frac{p}{e(p+1)}N^{1/(p(p+1))} \cdot \frac{1 - eN^{-1/p}}{1 + \frac{p+2}{p+1}N^{-1/(p+1)}}.$$

The last fraction is the correcting factor, and $\frac{p}{e(p+1)}N^{1/(p(p+1))}$ is the leading factor. At fixed p both corrections in the correcting factor tend to 0, so the correcting factor tends to 1, and the two estimates agree:

$$R_p(N) = \frac{p}{e(p+1)}\left(1 + o(1)\right)N^{1/(p(p+1))} = \Theta\left(N^{1/(p(p+1))}\right).$$

Here the constant $p/(e(p+1))$ lies in $[1/(2e), 1/e)$ for every $p \geq 1$.

The hierarchy is strict at every level beyond an explicit length. Write $\tau = N^{1/(p(p+1))}$, and suppose $\tau \geq 2e(p+1)/p$. Then $N \geq (2e)^{p(p+1)} \geq e^p$, so the lower estimate applies. The first correction is small: $eN^{-1/p} = e\tau^{-(p+1)} \leq e(2e)^{-2} = 1/(4e) < 0.1$, as $4e > 10$. So is the second: $\frac{p+2}{p+1}N^{-1/(p+1)} = \frac{p+2}{p+1}\tau^{-p} \leq \frac{3}{2}(2e)^{-1} = 3/(4e) < 0.28$, as $4e > 10.72$. So the correcting factor exceeds $0.9/1.28 > 0.7$. Meanwhile the leading factor $\frac{p}{e(p+1)}\tau$ is at least 2. Hence $R_p(N) > 1.4$ for every $N \geq (2e(p+1)/p)^{p(p+1)}$, a threshold that depends on p alone. For the comparison with the analytic regime, Theorem A.3(b) gives width $O(\log N)$, while $N^{1/p} = \omega(\log N)$ at every fixed p. □

D Scope, Attribution and Discussion

This appendix collects the remarks on the statements of the main text. They say what the hypotheses of a statement are for, what the nearest published result is, and why the obvious strengthening of a statement fails. Each remark opens by naming the statement it concerns and the question it answers, so that a reader who arrives from a cross-reference can read it on its own.

Remark D.1 (Why the Bound axis needs the norm convention). This remark concerns the norm convention of Section 2.1, and asks why the output-range axis Bound of Theorem 3.1 is read under it. The convention bounds the output range of a normalized model as follows. A normalized layer adds to each token at most H output-projected convex combinations of value vectors $O_h W_V x_j$, where H is the head count. So the output range of a normalized model at depth L is at most a constant times $(1 + Hc_V^2)^L$ times the norm of the embedded tokens, plus the offset of the readout. Here c_V bounds both the value matrices and the head output maps, and the factor $(1 + Hc_V^2)^L$ is $2^{\Theta(L)}$ when the head count is bounded. The convention keeps the norm of the embedded tokens at most $P(n)$, a polylogarithmic function of the length n, and it keeps the offset of the readout at most $P(n)$ as well.

Without the convention, a softmax layer of depth 1 with value matrix $W_V = nI$ would put the sum $\sum_j v_j$ into the residual stream, at cost $O(1)$. A readout with offset n would output the constant n, also at cost $O(1)$.

Even under the convention, the per-length Count_n column of Table 2 costs every row at most $O(\log^3 n)$. Up to the polylogarithmic factors of Theorem B.1, that column is therefore flat, and the Bound axis separates rows only under length-uniform realization, by one model at every length. Length-uniform realization needs the convention as well, on the embedding. With fixed parameters and embedded tokens of polylogarithmic norm, the output of a normalized row is $O(\mathrm{polylog}(n))$ at every fixed depth. So a normalized row realizes length-uniformly no task whose magnitude grows faster than that bound, and Count_n is among those tasks. A normalized row may still realize a task of polylogarithmic magnitude, for instance the value $\log_2 n$ read off the read position's own embedding.

The invariants are compared at a common cost, and the next remark works through what that comparison means on each axis.

Remark D.2 (Reading an invariant at a budget). This remark concerns Theorem 3.2, the profile $\mathcal{I}(A; C)$ of an architecture A at a cost budget C, and asks how each of the five invariants responds to the budget. Every gate of Theorem 4.2 is evaluated at this profile, so we instantiate it once here. Fix a budget C. By the cost formula of Eq. (2), a feedforward architecture of depth L may then take any width $d \le \sqrt{C/L}$. A recurrent architecture divides C/L between the squared width d^2 and the squared state dimension d_h^2. The five coordinates of Theorem 3.1 respond to that freedom in four different ways.

The kernel rank moves with the width alone. Polynomial-linear attention of degree p has $\mathrm{KRank}(A; C) = \binom{\lfloor\sqrt{C}\rfloor+p}{p}$, the supremum being attained at depth 1 and at the largest width that the budget then allows. Softmax, by contrast, has $\mathrm{KRank}(A; C) = \infty$ at every budget C.

The piecewise degree and the output range move with the depth alone. Write L_C for the greatest depth that the budget allows. Hard attention then has $\mathrm{PDeg}(A; C) = 2^{L_C}$. Every row whose layer outputs a convex combination of its inputs, or a single one of them, has $\mathrm{Bound}(A; C) = 2^{\Theta(L_C)}$.

The state dimension moves with the recurrent block: $\mathrm{State}(A; C) = \Theta(\sqrt{C})$ for a recurrent row, and $\mathrm{State}(A; C) = \infty$ for a feedforward one.

The symmetry group does not move at all: $\mathrm{Sym}(A; C) = \mathrm{Sym}(A)$ for every budget C. The reason is that equivariance is a property of the embedding and the mask, and neither of those is a resource that the budget allocates. This last reading is why the Sym gate is a binary separation rather than a scaling. The gate is a yes-or-no condition, and an architecture that fails it at one budget fails it at every budget.

The symmetry axis is also the one axis on which a larger value means less capacity, and the next remark fixes how the inequalities of the completeness theorem are read on it.

Remark D.3 (Orientation of the axes). This remark concerns the inequality $\mathcal{I}(A; C) \ge \mathcal{I}(T)$ in the statement of Theorem 4.2, and asks what it means on each of the five axes. Four of the five axes are oriented so that a larger value means more capacity: a larger kernel rank, a larger influence set, a larger output range and a larger state dimension. The symmetry axis runs the other way, because a larger equivariance group is a stronger constraint. So $\mathrm{Sym}(A) = S_n$, equivariance under every permutation, is the least capable setting, and $\mathrm{Sym}(A) = \{e\}$, equivariance under the identity alone, is the most capable. On the symmetry axis, an inequality $\mathcal{I}(A_2; C_1) < \mathcal{I}(T) \le \mathcal{I}(A_1; C_1)$ is read as $\mathrm{Sym}(A_2) \not\supseteq \mathrm{Sym}(T) \supseteq \mathrm{Sym}(A_1)$. In words, the losing architecture A_2 is equivariant under some permutation under which the task is not invariant, and every permutation under which the winning architecture A_1 is equivariant leaves the task invariant. On the rows of Table 1, where every Sym entry is $\{e\}$ or S_n, the two readings, as an inequality and as an inclusion, agree. We keep the single uniform inequality in the statements rather than splitting them by axis, and this remark is what licenses that choice.

The next six remarks concern the kernel-rank lemma, Theorem A.3, and the polynomial construction that matches it, Theorem A.4; the first says what is taken from the literature.

Remark D.4 (What the literature supplies here, and what it does not). This remark concerns Theorem A.3(a) and the dimension count behind it, Theorem A.2, and asks which parts are taken from the literature. The count itself, that the polynomials of degree at most p in d variables span a space of dimension $\binom{d+p}{p}$, is classical and is not ours. Its matrix form, that applying a polynomial of degree p entrywise to a matrix of rank d gives a matrix of rank at most $\binom{d+p}{p}$, appears as Lemma 2.3 of Alon (2009). The lower bound that the count yields is a separate matter. Sanford, Hsu, and Telgarsky (2023) prove an embedding–precision lower bound for sparse averaging and a head–embedding–precision tradeoff for triple detection by communication complexity. The precision count of Theorem D.6 is in their style rather than ours. They do not prove a rank bound for polynomial score functionals. We have not found in the literature the consequence for associative recall AR_N at degree p, which is part (a) of Theorem A.3 together with Theorem A.4.

The lower bound of part (a) and the construction of part (b) hold for different notions of realization, and the next remark says which and why.

Remark D.5 (Exact against approximate realization in the analytic regime). This remark concerns the two parts of Theorem A.3 and the error convention of Section 2.2, and asks which bounds hold for exact realization and which only for approximate realization. Part (a) of Theorem A.3 is a lower bound. It holds for exact realization of every linear-retrieval task, and at every error $\epsilon < 1$ for the tasks of retrieval type. Part (b) is not available at $\epsilon = 0$, and the obstruction is not technical, as the following argument shows.

Consider a single layer, normalized or not, with the split embedding and a score function $f_A > 0$, and let $N \geq 2$. Fix the query q, and let j^* be the key position whose key is q. Fix another key position $j \neq j^*$. Write $s = \sum_i G_{q,i} v_i$, where G is the kernel matrix of the proof of Theorem A.3 and v is the value vector. The residual stream at the query is an affine function of the scalar s, and every entry $G_{q,i}$ is positive. Put $\delta = \min(1, G_{q,j}/G_{q,j^*})$. The value vectors δe_{j^*} and $(\delta G_{q,j^*}/G_{q,j}) e_j$ then lie in the cube $[-1,1]^N$ of value vectors and give the same scalar s, hence the same output. Their targets, however, are δ and 0. So no single layer of softmax or sigmoid attention with the split embedding realizes AR_N exactly at any width, whatever its readout.

Every upper bound in this paper for a retrieval task in the super-polynomial analytic regime is accordingly ϵ-approximate, at the width of part (b). The softmax realizations of Avg_n are exact, by contrast, and so is that of Count_n under the bidirectional mask. Every separation drawn against one of these approximate upper bounds is drawn at its tolerance ϵ. On the other side of such a separation stands the lower bound of part (a), which holds at every $\epsilon < 1$ for the retrieval tasks. That range of errors covers the tolerance at which the upper bound is achieved, and this is what makes the two bounds comparable. A lower bound that held only for exact realization would say nothing against an approximate upper bound.

The two parts of Theorem A.3 also differ in how the width grows with the rank, and the next remark compares the two growth rates.

Remark D.6 (Polynomial-analytic dichotomy). This remark concerns the two parts of Theorem A.3, and compares the widths they give for a task of rank r. At every fixed degree p, the width that Theorem A.3(a) forces grows as a power of r. For an exponentially separating analytic kernel, the width is $O(\log r)$ by part (b). It is $\Theta(\log r)$ at every bounded precision, by the following count, which holds for every score functional.

Suppose that the embedded query vectors carry b bits per coordinate. Here b bits per coordinate means that each coordinate of an embedded query vector takes at most 2^b distinct values as the query varies. Then an embedded query vector in $\mathbb{R}^d$ takes at most 2^{bd} values, and the model is a function of the embedded tokens. So at most 2^{bd} queries can receive pairwise distinct embedded vectors, and the argument below finds r queries that must.

Fix a key set K at which the rank of a linear-retrieval task T is attained, and r queries whose coefficient vectors at K are linearly independent. These r queries have pairwise distinct target functionals, so an exact realization gives them pairwise distinct embedded vectors. For T of retrieval type and $\epsilon < 1/2$, take instead the queries q_i with coefficient vectors $\mathbf{c}(K, q_i) = e_i$, whose targets are the values v_i. Take two distinct indices i and i', and a value vector with $v_i = 1$ and $v_{i'} = -1$. On that vector the targets of the queries q_i and $q_{i'}$ differ by 2, which exceeds 2ϵ. So again the r embedded query vectors are pairwise distinct.

In either case $2^{bd} \geq r$, so $d \geq \log_2 r / b$. The query vectors of part (b) take two values in each coordinate, so part (b) runs at $b = 1$, and the width $\Theta(\log r)$ is attained at every precision. This count is why $\mathrm{KRank} = \infty$ does not mean cost-free.

The dichotomy is a statement at fixed p. Letting the degree grow closes the gap: Theorem A.4 at $p = \lceil \log_2 r \rceil$ has $\lceil r^{1/p} \rceil \leq 2$ and realizes AR_r exactly at width at most $2\lceil \log_2 r \rceil + 1$. Analytic kernels that are neither polynomial nor exponentially separating are covered by the count alone.

The kernel-rank lemma also carries three hypotheses that the tasks themselves do not need, and the next remark says what each is for.

Remark D.7 (What the split embedding gives). This remark concerns three hypotheses of Theorem A.3: the single key set, the split embedding, and the retrieval-type qualification on the tolerance of part (a). It asks what each is for. All three are there because the rank argument needs them, and not because the tasks do.

First, the rank of a task is read at a single key set. The reason is that the r coefficient vectors that the argument compares have to be rows of one kernel matrix, and a family spread over several key sets does not supply that. The two retrieval tasks AR_N and Index_n attain their rank at every key set.

Second, the split embedding keeps the scores independent of the values. This independence makes the output affine in the value vector v, and it makes the kernel matrix a fixed object to bound. Every construction in this paper is split. So is every retrieval model in the literature we know, since a key that reads its own value is a key that changes when the value does.

Without the split, part (a) survives in the unnormalized case with a linear readout. Write $F_j(v_j)$ for the summand that position j contributes to the output after the readout, as a function of its own value v_j. The key of position j is now $W_K(u_j + v_j w)$, where u_j is the key part of the embedding and w the value direction. For each fixed v_j, the summand $F_j(v_j)$ is a polynomial of degree at most p in the embedded query. Define the matrix Δ, with a row for each query q and a column for each position j, by $\Delta_{q,j} = \frac{1}{2}[F_j(1) - F_j(-1)]$. By the argument of Theorem A.2, the matrix Δ has rank at most $\binom{d+p}{p}$.

Now compare the outputs at a sign vector v and at $-v$, over all sign vectors v. The query's own summand cancels, and the comparison gives $\sum_j |\Delta_{q,j} - \mathbf{c}(K,q)_j| \le \epsilon$ for every query q. For a task of retrieval type at $\epsilon < 1$, this inequality makes the $r \times r$ submatrix of Δ on the retrieval queries and coordinates strictly diagonally dominant, hence of rank r. For exact realization $\epsilon = 0$, so the inequality makes every row of Δ equal to the coefficient vector of its query, and Δ is the coefficient matrix itself. In both cases $r \le \binom{d+p}{p}$, which is the bound of part (a) at one more dimension. In the normalized case the normalizer depends on the values, the summands no longer separate, and we leave that case open.

Third, the retrieval-type qualification on the tolerance of part (a) is not removable. Fix some $r \ge 3$, and consider the family of coefficient vectors $\mathbf{c}_1 = e_1$ and $\mathbf{c}_i = e_1 + \eta e_i$ for $2 \le i \le r$, which has rank r. A split linear attention of width 2 whose kernel is the indicator of the first key realizes this family to error η. At width 2, part (a) would ask $\binom{2}{1} = 2 \ge r$. For $\eta < 1$, part (a) therefore fails on this family as a statement about every task of rank r. The completeness theorem of Section 4 applies part (a) at positive error only to tasks of retrieval type: the class $\mathcal{T}_{\mathrm{KRank}}$ requires it (Theorem 4.1), and AR_N and Index_n are of that type. So nothing downstream is narrowed.

The polynomial row of Table 1 chooses its feature map freely, and the next remark shows that the width saving of Theorem A.4 depends on that freedom.

Remark D.8 (The feature map is part of the model). This remark concerns the polynomial row of Table 1 and its construction in Theorem A.4, and asks where the width saved by the degree comes from. The polynomial row chooses its feature map together with its other parameters, and the width that the degree saves comes from that choice. Fix instead the homogeneous kernel $\sigma(q,k) = \langle q,k\rangle^p$, taken unnormalized or, at even p, normalized. Then every single-layer model with the split embedding that realizes AR_N exactly has width $d \ge N+1$, at every degree p.

For the proof, fix a query q, and let j^* be the position whose key is q. The residual stream at the query is an affine function of the scalar $s = \sum_j G_{q,j} v_j$, where G is the kernel matrix. The entry G_{q,j^*} is nonzero, or the output would not depend on the payload asked for. Suppose that $G_{q,j} \ne 0$ for another key position j, and put $\delta = \min(1, |G_{q,j}/G_{q,j^*}|)$. Then the value vectors δe_{j^*} and $(\delta G_{q,j^*}/G_{q,j}) e_j$ lie in the cube $[-1,1]^N$ of value vectors and give the same s, hence the same output. Their targets, however, are δ and 0. On the keys, therefore, the kernel matrix G is diagonal and nonsingular up to reordering. Up to a positive row scaling, the kernel matrix is $G = M^{\circ p}$, the entrywise p-th power of the matrix M with entries $M_{q,j} = \langle \tilde{q}, \tilde{k}_j\rangle$. The off-diagonal zeros and the nonzero diagonal of G pass to M, so M is a nonsingular $N \times N$ matrix. The keys $\tilde{k}_j = W_K u_j$ lie in the image under W_K of the hyperplane $w^\perp$ orthogonal to the value direction w. That hyperplane has dimension $d-1$, as w is nonzero, so the rank of M is at most $d-1$. Therefore $N \le d-1$, that is, $d \ge N+1$.

By contrast, the row with a free feature map realizes AR_N exactly at width $d_p(N)+1$, by Theorem A.4. At that width the unnormalized homogeneous architecture has $\mathrm{KRank} = \binom{d+p-1}{p} \ge N$, and it has the same entries as the free row on the other four axes. For $p \ge 2$ the width $d_p(N)$ is $o(N)$, so the two architectures are separated, and none of the five invariants witnesses the separation.

At positive error the picture splits by degree. A realization to error $\epsilon < 1/2$ still has equal outputs on inputs with equal s. It follows that every row of the kernel matrix satisfies $\sum_{j \neq j^*} |G_{q,j}| \leq 2\epsilon |G_{q,j^*}|$. Otherwise some $t \in (2\epsilon, 1]$ has $t|G_{q,j^*}|$ below that sum. Then the value vector te_{j^*} and a value vector vanishing at j^* give the same s. Yet their targets, t and 0, are more than 2ϵ apart. The entry G_{q,j^*} is nonzero here too. Otherwise the inequality makes row q of G zero, and the output does not depend on the values. Then the value vectors e_{j^*} and $-e_{j^*}$ give the same output, although their targets are 2 apart.

At degree $p = 2$, row q of G is a positive multiple of the squares of the entries of row q of M. Divide each row q of M by its entry M_{q,j^*}, which is nonzero. This rescaling keeps the rank of M at most $d - 1$. It turns the row inequality of the previous paragraph into $\sum_{j \neq j^*} M_{q,j}^2 \leq 2\epsilon$. Reorder the columns so that the matching key of each query stands on the diagonal, and call the result M'. Its diagonal entries are 1, so its trace is $\operatorname{tr}(M') = N$, and its squared Frobenius norm is $\|M'\|_F^2 \leq (1+2\epsilon)N$. Let the columns of a matrix Y be an orthonormal basis of the column space of M'. There are $\operatorname{rank} M' \leq d - 1$ of them. Then $M' = YY^\top M'$, and $\|YY^\top\|_F^2 = \operatorname{tr}(Y^\top Y) = \operatorname{rank} M'$. By the Cauchy–Schwarz inequality for the entrywise inner product of two matrices, and by the bounds on the two Frobenius norms above,

$$\begin{aligned} N = \operatorname{tr}(YY^\top M') &= \sum_{q,j} (YY^\top)_{q,j} M'_{j,q} \\ &\leq \|YY^\top\|_F \|M'\|_F \leq \sqrt{(d-1)(1+2\epsilon)N}. \end{aligned}$$

Hence $d - 1 \geq N/(1+2\epsilon)$, which exceeds $N/2$ at every $\epsilon < 1/2$, so the separation persists at every such ϵ.

At degree $p \geq 3$ the width $N + 1$ is no longer forced for the unnormalized kernel. Since $G = M^{\circ p}$ up to a positive row scaling, the rescaling of the case $p = 2$ turns the row inequality into $\sum_{j \neq j^*} |M'_{q,j}|^p \leq 2\epsilon$. Over the $N - 1$ off-diagonal entries of row q, the power-mean inequality then gives $\sum_{j \neq j^*} (M'_{q,j})^2 \leq (N-1)^{1-2/p} (2\epsilon)^{2/p}$. Summing over the rows, $\|M'\|_F^2 \leq N\left(1 + (2\epsilon)^{2/p}(N-1)^{1-2/p}\right)$, and the trace of M' is still N. So the Cauchy–Schwarz step above gives

$$d - 1 \geq \frac{N}{1 + (2\epsilon)^{2/p}(N-1)^{1-2/p}} \geq \frac{1}{2} \min\left\{ N, \left(\frac{N}{2\epsilon}\right)^{2/p} \right\},$$

which is at least $\frac{1}{2} N^{2/p}$ at every $\epsilon < 1/2$, against the free row's width $d_p(N) + 1 = O(N^{1/p})$. So the separation from the free row persists at every $p \geq 2$. Take the random sign vectors ψ of the proof of Theorem A.3(b), in dimension $d = \lceil 2\ln(2N^2)(N/\epsilon)^{2/p} \rceil$. By Hoeffding's inequality and a union bound over fewer than N^2 pairs, with positive probability every non-matching normalized inner product is at most $\mu = (\epsilon/N)^{1/p}$ in absolute value. With such a choice of ψ, the queries and keys $\psi(k)/\sqrt{d}$, together with a value channel, realize AR_N to error $(N-1)\mu^p < \epsilon$ at width $d + 1 = o(N)$, which matches that lower bound up to a logarithmic factor.

The rank tools behind this remark, for matrices close to the identity and for entrywise powers of a matrix of low rank, appear in Alon (2009). His bounds are near matching for balanced sign codes up to logarithmic factors; the attention-kernel comparison here is our adaptation. The statements of this remark are for attention kernels, and they are proved here directly.

The contrast between polynomial and exponential kernels in Theorem D.6 has a precedent in the theory of associative memories, and the next remark says what it is and what is new here.

Remark D.9 (What the associative-memory literature already has). This remark concerns the polynomial-analytic dichotomy of Theorem D.6 and the kernel-rank lemma, Theorem A.3, and asks what the associative-memory literature already has. The dichotomy is not the first appearance of this phenomenon, and it would be wrong to present it as one. In dense associative memory, Krotov and Hopfield (2016) derived polynomial capacity at a fixed bit-error threshold, and Demircigil et al. (2017) proved exponential capacity for the generalized Hopfield dynamics. A degree-n polynomial interaction over N neurons has the N^{n-1} capacity scaling in the former calculation, while an exponential interaction stores $e^{\Theta(N)}$ random patterns in the latter theorem. Demircigil et al. also prove polynomial capacity for a related tensor-interaction update, distinct from the original polynomial dynamics. In these two results, as in those papers, the letter n is the degree of the interaction

and the letter N is the number of neurons. Elsewhere in this paper the same letters denote the sequence length and the number of keys.

Millidge et al. (2022) state the pair as a single principle about the separation function. The exponential half suggests the $O(\log r)$ width of Theorem A.3(b); that attention-model bound follows from our construction, with the matching lower bound at bounded precision from Theorem D.6. The polynomial half, inverted, is a width that grows as a root of the capacity, as in Theorem A.3(a) and Theorem B.3: the index $n-1$ of a degree-n energy is the degree of the kernel that its update applies. Since the modern Hopfield update rule is the attention mechanism (Ramsauer et al., 2021), the correspondence is not an analogy.

Three differences separate the statements here from those results, and these differences are what this paper contributes on this point. First, those results are capacity statements about randomly drawn patterns under an energy dynamics, and they hold with high probability. The polynomial count holds only when a small fraction of retrieval errors is tolerated. Theorem A.3(a), by contrast, is a worst-case bound that holds for every model in the class and every admissible instance. Second, those results are achievability results: they count how many patterns a given mechanism does store. Part (a) is a lower bound on the width that every realization must have, and no capacity count supplies such a bound. Third, the mechanism differs. The bound here comes from the dimension $\binom{d-1+p}{p}$ of the polynomial feature space of the key coordinates, through which the kernel factors by Theorem A.2, rather than from a statistical count of retrievable patterns. That is why the bound is insensitive to the distribution of the keys, and why Theorem A.4 matches it exactly.

The next remark turns from the kernel-rank axis to the influence-set bound behind the PDeg axis, and compares it with the nearest published argument.

Remark D.10 (Relation to Hahn's impossibility results). This remark concerns the influence-set bound, Theorem A.6, and compares it with the restriction argument of Hahn (2020). The influence-set argument is the region-wise counterpart of Hahn's argument. The two are close but not the same, so we say which is which. Hahn proves that for every hard-attention transformer and every constant fraction $C \in (0,1)$ there is a restriction, leaving at least Cn inputs free, under which the computed function depends on a number of inputs independent of n. From this Hahn concludes that hard attention models neither PARITY nor $\mathrm{2DYCK}$. The per-layer growth factor in that argument is a constant depending on the head count, the fan-in and the fraction C, rather than 2. The argument also fixes inputs, rather than passing to a region on which the argmax pattern is constant. Theorem A.6 is the statement that the cost analysis here needs, and it is proved in Section C rather than imported. The conclusion it draws for Avg_n is a depth bound rather than an impossibility.

The State axis rests on the linear-recurrence rank bound, and the next remark places that bound against the published ones.

Remark D.11 (The published state bounds, and what the rank bound adds). This remark concerns the linear-recurrence rank bound, Theorem A.7, by which a single recurrence realizing AR_N to error $\epsilon < 1$ needs state dimension $d_h \geq N$, and asks how it relates to the published bounds. Two published bounds are close to Theorem A.7, and neither is the same statement. Arora et al. (2024b) prove, by a communication-complexity reduction, that every causally dependent recurrent model needs $\Omega(N)$ *bits* of state to solve multi-query associative recall. Their result is stronger in the model class, since it covers selective recurrences and gated convolutions alike. It is weaker in the currency, since bits become dimensions only after a precision convention. Sanford, Hsu, and Telgarsky (2023) prove the matching $\Omega(N)$-bit bound for recurrent networks on sparse averaging. Theorem A.7 is the counterpart of these two bounds for a single recurrence, in the currency that the State axis uses: a count of dimensions, with no precision convention and no hidden constant. The kernel argument of its proof is the reason that State is an invariant of the architecture rather than a bit budget.

The next remark returns to the kernel-rank axis and extends the rank count of Theorem A.3(a) to a layer with several heads.

Remark D.12 (Multi-head rank count). This remark concerns Theorem A.3(a) and Theorem B.2 for a layer with several heads, and asks how the head count enters the rank bound. Consider a depth-1 polynomial-linear model with H heads, each of degree at most p and of width d/H. The model has the split embedding, aggregates its heads additively, and has a token-wise readout. Summing the kernel decompositions of the heads shows that the residual stream at the query position is independent of the value vector v on the common null space of the H kernel matrices. So the argument of

Theorem A.3(a) runs on the matrix that stacks them, and realizing AR_N to error $\epsilon < 1$ forces

$$N \le \min\left\{ H\binom{d/H+p}{p}, \binom{d-1+p}{p} \right\}.$$

The first bound is rank sub-additivity combined with Theorem A.2 at the per-head dimension d/H, and it gives $d \ge (p/e)H^{1-1/p}N^{1/p} - Hp$. The second bound is Theorem A.2 at dimension $d-1$, since every key of every head is a linear image of the projection of the key part of the embedding onto $w^{\perp}$. Neither bound dominates the other. At $p = 1$ the second gives the bound $d \ge N$ of Theorem B.2, against the bound $d \ge N - H$ from the first. At $p \ge 2$, with $H \ge 2$ and d/H large, the first is the smaller. Either way, the head count changes the rate at which KRank binds, and it introduces no axis of its own.

On the kernel-rank axis, part (i) of Theorem 4.2 reaches depth above one only through compositional inheritance, and the next remark explains the hypothesis that this step carries.

Remark D.13 (Why the hypothesis is there). This remark concerns the layered hypothesis of Theorem A.8, and asks why the lemma needs it. Without that hypothesis nothing in the proof of the lemma transfers, and there are two separate reasons. First, the layers below layer ℓ may re-embed the input. A first layer can move each value into the channel of a different key, or spread a key over several positions. After that, the layer that retrieves reads no split embedding of the input, and the kernel decomposition of Theorem A.3 has nothing to bound. Second, the layers above layer ℓ may read positions other than the read position, so that no single layer's output determines the target.

So the lemma is a statement about realizations in which one layer performs the retrieval from a fixed embedding. In such a realization the layers below that layer act position by position, and the layers above it read the read position alone. The realization is then a depth-1 model, padded, and that is what Theorem 4.2 assumes when it lifts the KRank axis past depth 1. We do not know whether the KRank lower bound holds at depth greater than one without that hypothesis, that is, whether depth can substitute for kernel rank on a linear-retrieval task. Amsel, Yehudai, and Bruna (2025) show, for a different low-rank attention model and an approximate nearest-neighbor task, that depth helps at short contexts; they conjecture a long-context separation. Their result does not settle the kernel-rank question here.

Part (i) of Theorem 4.2 also restricts its PDeg axis to exact realization, and the next remark says why.

Remark D.14 (The PDeg axis at positive error). This remark concerns the PDeg axis of Theorem 4.2(i), and asks why that axis is stated for exact realization only. At error $\epsilon > 0$ the argument behind the PDeg axis gives nothing. The reason is that the argument is local. It finds an open set on which the model is a function of few positions, and it asks the target to depend on more positions there. Exact agreement on an open set transfers that dependence to the model. Agreement to within ϵ does not, since every continuous target is within ϵ of a constant on a small enough open set, and nothing bounds how small the regions of a selective functional are. For Avg_n more is true. At $\epsilon \ge 1/n$, no change of one coordinate moves the target by more than 2ϵ, so every argument at positive error has to perturb many coordinates at once.

So the PDeg axis of part (i) is a statement about exact realization. The depth $\log_2 n$ that hard attention pays on Avg_n and Count_n in Table 2 is the cost of exact realization. Under the bidirectional mask, part (c) of Theorem A.9 attains the least depth $\lceil \log_2 n \rceil$ that the influence-set bound of Theorem A.6 allows, at every n. We do not know whether hard attention realizes Avg_n to a fixed error $\epsilon > 0$ at depth $o(\log n)$.

The next remark concerns the position-free row of Table 1, and relates Theorem 4.3 to an exact characterization in the literature.

Remark D.15 (The function-class face of the corollary). This remark concerns Theorem 4.3, which says what NoPE softmax, the position-free row, realizes. It asks how the corollary relates to a function-class characterization of position-free transformers in the literature. Sälzer et al. (2026) characterize the average-hard-attention transformers with neither positional encodings nor masking exactly, as the semialgebraic counting properties. In words, what such an architecture decides about a word over a finite alphabet is a property of its letter counts. Their exact characterization is for average-hard attention. For softmax they prove that every semialgebraic counting property is expressible, which is an inclusion rather than a characterization.

Theorem 4.3 is the cost-side statement for the position-free row, on the real-valued canonical tasks. By part (i) of Theorem 4.2, a position-free row realizes no task of $\mathcal{T}_{\mathrm{Sym}}$, the task class of the symmetry axis, since each such task is changed by some permutation that the row cannot distinguish. The task Index_n is such a task (Theorem 4.3), and no budget repairs the failure. The three tasks that pass are realized at the costs of part (ii) of Theorem 4.2, among them AR_N at width $O(\log N)$. Neither their characterization nor Theorem 4.3 subsumes the other, for the reason Section 1.3 gives. For average-hard attention theirs fixes which properties are computable, and for softmax it shows that every semialgebraic counting property is expressible; this one fixes what the three S_n-invariant canonical tasks cost.

Part (ii) of Theorem 4.2 proves sufficiency on the four canonical tasks only, and the conjecture below states the extension of part (iii) beyond them.

Conjecture D.16 (Unrestricted completeness)**.** *Read with realization to a fixed error $\epsilon \in (0, 1/2)$, part (iii) of Theorem 4.2 holds for every task and every pair of rows of Table 1 of matching depth and common mask, not only for the canonical tasks.*

The next remark separates what the completeness theorem proves from what the conjecture asks, and explains the scope of each part of the theorem.

Remark D.17 (What is proved and what is conjectured). This remark concerns the completeness theorem, Theorem 4.2, and Theorem D.16, and asks which parts are proved and why each restriction is there. Part (i) of the theorem is the substantive half: each of the five invariants is a genuine obstruction, for every task in its class and every standard architecture, subject to three restrictions. The three restrictions are the caveat on the KRank axis at depth greater than one, the restriction of the PDeg axis to exact realization, and the restriction of the recurrent rows to a single recurrent layer. The last restriction is needed. A stack of recurrent layers holds its state jointly, and the kernel argument of Theorem A.7 then gives only $Ld_h \geq r$. Consider N selective layers of state dimension 1, each keeping the payload of one key and gated at the query by whether the query is that key. Together they realize AR_N exactly at cost $O(N)$, against the cost $\Theta(N^2)$ of one recurrence.

Part (ii) is what turns soundness into completeness, and it is a finite verification rather than a general theorem. It says that on the canonical tasks the five gates are not merely necessary but jointly sufficient, so nothing outside the tuple of five invariants is doing any work there. Theorem D.16 is the statement one would like. The obstacle to proving it is not a missing lemma but the quantifier: sufficiency has to be established task by task, and there is no enumeration of tasks. The conjecture is stated over the rows of Table 1, and not over every standard architecture, because over every standard architecture it is false. Theorem D.8 exhibits two polynomial kernels of the same degree that at some budget agree on four invariants and both have kernel rank at least N. At that budget and depth 1, one of them realizes AR_N exactly and the other, the homogeneous kernel $\langle q, k\rangle^p$, does not. The homogeneous kernel does not realize AR_N at that budget and depth 1 even to an error below $1/2$, by the bound of Theorem D.8 at positive error, at every degree $p \geq 2$ and every large enough N.

The conjecture is read at a fixed positive error because, for exact realization, it is false even over the rows of Table 1. It is false there as soon as the readouts are linear or piecewise-linear, as feedforward networks with ReLU activations are. The counterexample is a task that we call the weighted average, WAvg_n. Position j of an input carries a value $v_j \in [-1, 1]$ and a log-weight $s_j \in [-1, 0]$. The target at the read position n is the average of the values under the weights e^{s_j}, namely $\sum_{j=1}^{n} e^{s_j} v_j / \sum_{j=1}^{n} e^{s_j}$.

Softmax attention realizes WAvg_n exactly at depth 1 and width 4. Embed position j as $(v_j, s_j, 1, 0)$. Let the query read the constant coordinate and the key read s_j, so that the score of position j is s_j and its softmax weight is $e^{s_j} / \sum_{i=1}^{n} e^{s_i}$. Under the bidirectional mask the read position attends to every position, itself included. Let the value copy v_j into the fourth coordinate, which the embedding leaves at 0, and let the readout return that coordinate. The output at the read position is then the target. The embedding is affine, so it is real-analytic on all of $\mathbb{R}^2$, as Theorem 2.4(ii) asks. The readout is linear, and the embedded tokens have norm at most $\sqrt{3}$.

Average-hard attention with a linear or piecewise-linear readout does not realize WAvg_n exactly at any depth or width. The proof is by contradiction, in three steps. It first finds an open set of inputs on which every argmax set is constant. It then shows that on that set the output is a sum of functions of one token each. Finally, it compares a mixed second derivative of that sum with the same derivative

of the target. Suppose, then, that an average-hard model with a linear or piecewise-linear readout realizes WAvg_n exactly, for some $n \geq 2$.

First, there is a nonempty open set U of inputs on which every argmax set, at every layer, head and position, is constant. To find it, start from the interior of the input domain, take the layers in order and, within a layer, each head-position pair in turn, and shrink the open set once for each pair. On the current open set the earlier argmax sets are fixed, so the scores of the current layer are continuous in the input. Near any input the argmax set can therefore only lose elements, since a position scoring strictly below the maximum still does so nearby. So at an input where the size of the argmax set is least over the current open set, the argmax set is constant on a neighborhood. That neighborhood becomes the next open set.

Second, fix a layer, a head of that layer and a position, and let W_V be the value matrix of the head. On U the argmax set of the head at that position is fixed. At that position, the head outputs the average of W_V applied to the representations at the positions in that argmax set. The layer sums the outputs of its heads, and adds the sum to the representation when there is a residual connection; either way the update is linear in the representations. So on U the residual stream at the read position is $\sum_j M_j \iota_j$. Here ι_j is the embedded token at position j, a real-analytic function of the input token at position j alone, and the M_j are fixed matrices.

A piecewise-linear readout is affine on each of finitely many closed polyhedra that cover its domain. The preimages of these polyhedra, under the map from the input to the residual stream at the read position, cover U by finitely many relatively closed sets. A finite union of closed sets with empty interior has empty interior, so the interior of one of these sets is nonempty. Call that interior V. On V the output is an affine function of $\sum_j M_j \iota_j$. It is therefore a sum $\sum_j g_j$ of real-analytic functions g_j, each of the token at position j alone.

Third, the value v_1 sits at position 1 and the log-weight s_2 at position 2, so no single g_j depends on both. The mixed second derivative of the output in these two variables is therefore 0 on V. For the target, the same derivative is

$$-\frac{e^{s_1+s_2}}{\left(\sum_{j=1}^{n} e^{s_j}\right)^2},$$

which is never 0. This contradicts exact realization.

In Table 1 the softmax row and the average-hard row have equal invariants at every budget. Yet average-hard attention does not realize WAvg_n exactly at any cost, while softmax realizes it at constant cost. Let $T = \mathrm{WAvg}_n$, let A_1 be the softmax row and let A_2 be the average-hard row. The conjecture extends part (iii) to every task. Applied to T and read for exact realization, part (iii) would then require an invariant with $\mathcal{I}(A_2; C_1) < \mathcal{I}(T) \leq \mathcal{I}(A_1; C_1)$. Equal invariants rule that out. The obstruction lies in the readout class: to realize WAvg_n exactly, the readout would have to divide. On the majority task below, the readout would likewise have to threshold. The argument obstructs exact realization only, since a mixed derivative constrains nothing at a positive error. The conjecture is therefore stated at a fixed positive error.

Two exclusions in part (ii) are deliberate. The majority task, whose target is 1 when more than half of n binary values are 1 and 0 otherwise, is left out because its realization turns on whether the readout class of Theorem 2.4(iii) contains a threshold. That is a question about the readout rather than about attention. Positionally masked rows are left out because a window fixes the influence set by position rather than by content. So a task can fail on a sliding window while its PDeg gate clears. For instance, take the task whose target, read at position n, is the first value v_1. That task depends on one position, so its piecewise-degree gate clears at depth 1. Yet a window of half-width w carries v_1 to position n only at depth at least $(n-1)/w$. That failure is a property of the mask, which every comparison here holds fixed, and not a defect of PDeg.

Part (iii) of Theorem 4.2 names an invariant for each separation it covers, and the next remark says which invariant that is on each canonical task.

Remark D.18 (Which invariant part (iii) names). This remark concerns part (iii) of Theorem 4.2 and Table 2, and asks which of the five invariants part (iii) names on the canonical tasks. Part (iii) names three of them. Let T, A_1, A_2 and C_1 be as in part (iii), with its hypotheses in force. Then the following hold.

(a) The task T is AR_N or Index_n.

(b) The invariant that part (iii) names depends on the row A_2 alone. When A_2 is a polynomial row, of any degree, the invariant is KRank. When A_2 is NoPE softmax, the invariant is Sym, and then T is Index_n. When A_2 is either state-space row, the invariant is State. No other row can be A_2, and no other invariant satisfies $\mathcal{I}(A_2; C_1) < \mathcal{I}(T)$.

(c) Part (ii) remains true when its hypothesis asks only that the gates of KRank, Sym and State clear.

In particular, part (iii) names neither PDeg nor Bound on any canonical task.

Part (a) is proved by contradiction. Suppose that T is Avg_n or Count_n. By the proof of Theorem 4.2(ii), every row of the kind that part (ii) allows realizes T, at the tolerance of part (ii), at cost $O(\log^3 n)$. The largest of these costs is that of Theorem A.9(b) for causal softmax on Count_n, at depth at most $2\log_2 n$ and width $2\lceil \log_2 n \rceil + 2$. Every other cell of the two columns costs $O(\log n)$. Every cost is at least 1, since $L \geq 1$ and $d \geq 1$ in Eq. (2), so $C_1 \geq 1$. Hence A_2 realizes T at cost $O(\log^3 n) \leq C_1 \cdot O(\log^3 n)$, which is $\tilde{O}(C_1)$. This contradicts the hypothesis of part (iii) that A_2 requires cost $\omega(\tilde{O}(C_1))$. So T is neither Avg_n nor Count_n.

For (b), let T be AR_N or Index_n. The proof checks the five gates one at a time. The gates of PDeg and Bound clear for every row, and each of the other three can fail only for the rows that (b) lists. Let A be any row of the kind that part (ii) allows, and let C be any budget at which A has a resource setting.

Every row has a resource setting of cost at most 2, at depth 1, width 1 and, on a recurrent row, state dimension at most 1. And $C_1 \geq 2$: a model of cost 1 has depth 1 and width 1, where the split embedding makes the query and key maps zero, since they vanish on the value direction, which spans $\mathbb{R}$. At fixed keys its output is then a function of the query and of $\sum_j a_j v_j$, for fixed reals a_j. Take the position j_0 with $|a_{j_0}|$ least and another position j_1. The value vectors e_{j_0} and $(a_{j_0}/a_{j_1})e_{j_1}$, or 0 when $a_{j_1} = 0$, give equal outputs. Yet the query that asks for the payload at j_0 has targets 1 and 0 on them, which no realization to error below $1/2$ allows. So C_1 is a budget at which A_2 has a resource setting.

The task profiles in the proof of Theorem 4.2(ii) give $\mathrm{PDeg}(T) = 1$ and $\mathrm{Bound}(T) = 1$. A resource setting of cost at most C keeps its cost at most C when its depth is lowered to 1, since the cost $L(d^2 + d_h^2)$ of Eq. (2) is increasing in L. So A has a setting of depth 1 at the budget C.

The PDeg gate clears. By Table 1, the piecewise degree $\mathrm{PDeg}(A, L)$ at depth L is 2^L for hard attention, and for every other row it is the length, which is at least 2. So $\mathrm{PDeg}(A, L) \geq 2$ at every depth $L \geq 1$. The Bound gate also clears. It asks $\mathrm{Bound}(A; C) > \mathrm{Bound}(T) - 1/2 = 1/2$. There is an in-budget witness: at width 1 and depth 1, embed every token as 1 and take the value map and readout to be the identity. Every normalized attention row then outputs 1 on the all-one input; a polynomial row with constant feature 1 outputs at least 1, and a sigmoid row outputs at least 1 because the input has at least two positions. For either recurrent row, take $R_t = 0$ and $E_t = 1$, and read the state $h_t = 1$. Each witness costs at most $2 \leq C$, so $\mathrm{Bound}(A; C) \geq 1 > 1/2$.

The other three gates clear for most rows. Every row other than a polynomial one clears the KRank gate, since its kernel rank in Table 1 is ∞. Every feedforward row clears the State gate, since its state dimension is ∞. On the symmetry axis the gate reads $\mathrm{Sym}(A) \subseteq \mathrm{Sym}(T)$ at every budget, by Theorem D.2. On AR_N it clears for every row, since $\mathrm{Sym}(\mathrm{AR}_N)$ is S_n, the group of all permutations of the positions other than the read position, and S_n contains every group of such permutations. On Index_n, where $\mathrm{Sym}(\mathrm{Index}_n) = \{e\}$, it clears for every row with $\mathrm{Sym}(A) = \{e\}$, which is every row except NoPE softmax. It fails for NoPE softmax, whose group S_n is not contained in $\{e\}$ when $n \geq 2$.

So the gates that can fail on T are, row by row, the following. A polynomial row has $\mathrm{Sym}(A) = \{e\}$ and $\mathrm{State}(A) = \infty$, so only its KRank gate can fail. NoPE softmax has $\mathrm{KRank}(A) = \mathrm{State}(A) = \infty$, so only its Sym gate can fail, and only on Index_n. A state-space row has $\mathrm{KRank}(A) = \infty$ and $\mathrm{Sym}(A) = \{e\}$, so only its State gate can fail. Softmax, hard, average-hard, sigmoid and causal softmax attention have no gate that can fail on T.

Part (iii) supplies, for the losing row A_2, an invariant with $\mathcal{I}(A_2; C_1) < \mathcal{I}(T)$. Under the reading of Theorem 3.2 and Theorem D.3, that inequality says that the gate of this invariant fails for A_2 at the budget C_1. Hence A_2 is a polynomial row, NoPE softmax on Index_n, or a state-space row. The named invariant is then the one that (b) lists for that row, the only gate of A_2 that can fail.

For (c), let T be a canonical task and let A be a row of the kind that part (ii) allows. Let C be a budget at which the gates of KRank, Sym and State clear. Suppose first that T is Avg_n or Count_n. Then A realizes T at cost $O(\log^3 n)$, as in the proof of (a), and this is $\tilde{O}(C)$ because $C \geq 1$. Suppose instead that T is AR_N or Index_n. Then the gates of PDeg and Bound also clear at C, by the proof of (b). So all five gates clear, and part (ii) gives a realization at cost $\tilde{O}(C)$.

The two invariants that part (iii) does not name are obstructions all the same, by part (i). On the per-length columns of Table 2 they act below the resolution of $\tilde{O}(\cdot)$. The PDeg axis forces hard attention to depth $\lceil \log_2 n \rceil$ for an exact realization of Avg_n or Count_n, by Theorem A.6, and Theorem A.9(c) attains that depth. Softmax realizes Avg_n at depth 1, so the PDeg axis separates the two rows in depth, by a factor of order $\log n$ in cost. The Bound axis separates only under length-uniform realization, as the last column of Table 2 and Theorem D.1 record.

The separations that part (iii) does name are the following, with the polynomial row at a fixed degree p, and with n in place of N on Index_n. On both retrieval tasks a polynomial row costs $d_p(N)^2$, where $d_p(N)$ is the least d with $\binom{d+p}{p} \geq N$, as in Theorem A.4. It is separated from every bidirectional row that realizes the task at cost $O(\log^2 N)$, and from every polynomial row of higher degree. Theorem B.2 is the case $p = 1$, and Theorem B.3 the case of consecutive degrees. NoPE softmax is separated on Index_n from every other bidirectional row, as in Theorem 4.3. The two state-space rows, at cost N^2 up to a factor $\log^2 N$, are separated from causal softmax, at cost $\log^2 N$, on both retrieval tasks. The hypothesis of part (iii) that A_2 requires the larger cost is established for NoPE softmax at every depth, by the Sym axis, and for the state-space rows by Theorem A.7. For a polynomial row it is established for realizations of depth 1, by Theorem A.3(a). It is also established for layered realizations whose extracted layer reads a split embedding, by Theorem A.8, and Theorem D.13 says what is open beyond them.

The next remark says where the five invariants of Theorem 3.1 come from.

Remark D.19 (Where the five invariants come from). This remark concerns the five invariants of Theorem 3.1, and asks where they come from. They arise from structural choices in an attention architecture. The score functional contributes two of them: KRank through its analytic feature space, and PDeg through its selectivity. The normalization contributes Bound. The positional structure of the embedding, together with the mask, contributes Sym. The stacking discipline contributes State. These are distinct measurements, although an architectural choice can affect more than one of them. Theorem 4.2(ii) shows that the five gates are jointly sufficient on the canonical tasks under its stated conditions. The head count is the one architectural parameter that contributes no invariant. By Theorem D.12, it changes the rate at which KRank binds and introduces no axis of its own. That is what one expects from the cost formula of Eq. (2), in which the head count H does not appear.

The last two remarks concern the consequences of Section B: the cost table and the case of degree one.

Remark D.20 (What each entry of the cost table is). This remark concerns Table 2, and says what each entry of the table is and which entries are exact. Every entry is the cost of Theorem 2.2 at the width, depth and state dimension that the proof of Theorem 4.2(ii) supplies for that entry. Three kinds of entry are ϵ-approximate: the retrieval entries of the softmax, sigmoid and causal softmax rows, the AR_N entry of NoPE softmax, and the Count_n entry of causal softmax. Their cost is independent of ϵ for every $\epsilon \geq e^{-\mathrm{polylog}(n)}$. Every other entry is for exact realization. The lower bounds behind the hard row on Avg_n and Count_n hold for exact realization, by Theorem 4.2(i). Those behind the polynomial row on the retrieval tasks at depth above one hold for layered realizations, also by Theorem 4.2(i). The sliding-window row is absent from the table because Theorem D.17 excludes positionally masked rows from the sufficiency half.

The last remark compares Theorem B.2, the case of degree one, with the literature.

Remark D.21 (The literature at degree one). This remark concerns Theorem B.2, the case of degree one, and asks what the literature supplies there. At this degree the literature supplies what Theorem D.4 records at every degree: the precision-sensitive attention-size bounds of Sanford, Hsu, and

Telgarsky (2023), by communication complexity, for sparse averaging and triple detection. Those bounds are a different statement, about a different pair of tasks.

For one head, the capacity limit has a long history. Kohonen and Ruohonen (1973) gave the construction direction: a linear associative memory recalls its pairs perfectly when the cues are linearly independent. Schlag, Irie, and Schmidhuber (2021) argued, in linear attention itself, that storing more associations than the key dimension causes a retrieval error. Theorem B.2 makes that argument a worst-case theorem at every error below 1, and extends it to H heads.

Ghriss and Chakraborty (2026) frame associative recall with positive linear features as a packing problem. At zero interference, d orthogonal positive-linear keys are possible, and their Welch bound keeps capacity $O(d)$ when interference tolerance is below a fixed multiple of $1/\sqrt{d}$. Their degree-two rank-one positive-semidefinite features admit exponentially many pairwise nearly orthogonal keys at fixed positive tolerance once d is large enough. Those are counts of keys that can be kept pairwise nearly orthogonal. Theorem B.2 and Theorem A.3(a) instead bound the width of every realization on worst-case values, where the interference of all the other keys adds up. So the exponential count at degree two and the width $\Omega(N^{1/2})$ at $p = 2$ measure different things, and they do not conflict.

E THE TECHNIQUES IN FULL

The proof of the completeness theorem rests on four technical ingredients. They are the kernel factorization (Theorem A.2), the influence-set bound (Theorem A.6), the linear-recurrence rank bound (Theorem A.7), and compositional inheritance (Theorem A.8). All four are elementary, and all four are proved in this paper rather than imported. What the literature supplies is not these statements but the separations they explain. For the first three ingredients, a remark says what the nearest published result is (Theorems D.4, D.10 and D.11). The fourth ingredient, compositional inheritance, carries a hypothesis, and Theorem D.13 says why the argument stops at that hypothesis. The contribution of the paper is the assembly. It shows that these ingredients, applied to the five invariants in the right combination, exhaust the structural axes on the tasks where every cost is known. It does so within the fixed-embedding regime, in which the embedding of a token depends only on its content and its position.

T1, bilinear kernel factorization, for kernel-rank. Consider a polynomial-linear attention layer of degree p and width d. Its output before the readout is a bilinear form, whose kernel matrix has, in row q and column j, the normalized score between the query q and the key at position j. This kernel matrix factors through the polynomial feature space, which has dimension at most $\binom{d+p}{p}$. Realizing a linear-retrieval target of rank r therefore forces $\binom{d+p}{p} \geq r$. When $r \geq (2e)^p$, this in turn forces a width of $\Omega(p \cdot r^{1/p})$.

The factorization is the standard polynomial-kernel feature map, stated and proved as Theorem A.2. Its consequence for AR_N, and the matching construction of Theorem A.4, are ours. For exponentially separating kernels such as softmax and sigmoid (Theorem A.3(b)), a Johnson–Lindenstrauss-style random-sign construction (Johnson & Lindenstrauss, 1984) gives an $O(\log r)$ upper bound. It sends each of the r keys to a random sign vector in $O(\log r)$ dimensions. Some choice of the signs gives every two distinct keys an inner product of absolute value at most half that of a key with itself. The lower bound for these kernels is a capacity count based on the bits of precision per coordinate (Theorem D.6). That count is in the style of the precision-sensitive size bounds of Sanford, Hsu, and Telgarsky (2023), and Theorem D.4 says what they prove instead.

T2, influence-set doubling, for piecewise-degree. The argmax of hard attention is constant on each of finitely many cells of the input space. On each cell, the output at a query position depends on a fixed set of input positions, which we call its influence set. The size of the influence set at most doubles with each layer, so at depth L the influence set has at most 2^L positions. A task that depends on all n positions throughout some open set therefore requires depth $\log_2 n$ for exact realization. We prove this bound as Theorem A.6, and we prove it not for the argmax alone but for every score functional with boundedly many active weights. Such a functional is κ-selective in the sense of Theorem A.5: on each of finitely many regions it puts every nonzero weight on at

most κ positions. The influence-set bound is the region-wise counterpart of the restriction argument by which Hahn (2020) showed that hard attention models neither parity nor $2\mathrm{DYCK}$, the Dyck language with two types of brackets. Theorem D.10 separates the two arguments.

T3, recurrence rank, for recurrent-state. In a streaming retrieval task, the values before the query must pass through the hidden state before the query arrives. Under the split embedding, the state is a linear function of those values, with a coefficient matrix of rank at most the state dimension d_h. The task's coefficient matrix therefore cannot have rank above d_h. This is Theorem A.7, and Theorem D.11 distinguishes this dimension bound from the bit bound of Arora et al. (2024b).

T4, layered realization, for compositional inheritance. For a multi-layer architecture, T4 lifts the single-layer rank bound of T1 to the composed model. It applies to a realization with the following property: *one layer reads a fixed embedding of the input, and its output at the read position determines the target*. Such a realization contains that layer as a depth-1 realizer with a token-wise readout. Every single-layer bound that holds for an arbitrary token-wise readout then transfers to the multi-layer model, and the kernel argument of T1 is a bound of that kind. This transfer is Theorem A.8, and the italicized hypothesis is where the argument stops. Without that hypothesis, the layers below can re-embed the input and the layers above can read other positions. Theorem D.13 says what is then open.

Multi-head extension of T1. Consider attention with H heads, each of width d/H. The kernel argument of T1 then runs on the matrix that stacks the H kernel matrices of the heads. That stacked matrix has rank at most $\binom{d-1+p}{p}$. The rank of a stack of matrices is at most the sum of their ranks, and this sub-additivity also gives the stacked matrix rank at most $H\binom{d/H+p}{p}$. These two bounds are Theorem D.12, and at $p = 1$ the count is Theorem B.2. The multi-head case introduces no new invariant: it is T1 applied to each head, with the heads combined linearly. The head count does not enter the cost proxy of Eq. (2).

Combining the techniques. The proof of the theorem first establishes soundness on each axis in turn, that is, that each invariant is a genuine obstruction on its own class of tasks. The Bound axis rests on a comparison of output ranges, and the Sym axis on a comparison of equivariance. For the PDeg axis the proof uses the selectivity bound of T2, for the State axis the recurrence rank bound, and for the KRank axis T1 together with T3. The proof then verifies, task by task on the canonical list, that the five gates are jointly sufficient. For an architecture A at a cost budget C and a task T, the gate of an invariant $\mathcal{I}$ is the condition $\mathcal{I}(A; C) \geq \mathcal{I}(T)$. That verification is what licenses reading a separation off the profiles. Each invariant reads a single architectural component, but two invariants can read the same one. The score functional is read by two of the five invariants (Theorem D.19), while head count enters the kernel-rank bounds through the stacked head kernels. Under the theorem's mask and realization assumptions, the five gates account for the cost gaps it covers on the canonical tasks.

F Extended Related Work

This appendix places the paper against the field in full. Section 1.3 gives the short form.

The architecture and the variants the taxonomy ranges over. The rows of our taxonomy are sequence-mixing architectures studied in the literature. Softmax attention is the mechanism of Vaswani et al. (2017). Polynomial-linear attention abstracts the kernel view of attention set out by Tsai et al. (2019), together with the linear-time variants that view produced. Feature-map variants include the formulation of Katharopoulos et al. (2020), the random-feature approximation of Choromanski et al. (2021), and the fast-weight reading of Schlag, Irie, and Schmidhuber (2021). Linformer (Wang et al., 2020) instead projects keys and values along the sequence dimension to approximate softmax attention. A finite-dimensional feature map caps the rank of the attention matrix, and that cap is exactly what KRank measures. The recurrent rows are the structured state-space line, from S4 (Gu et al., 2022) through Mamba (Gu & Dao, 2024) and the state-space duality of Dao and Gu (2024). The convolutional variants of Poli et al. (2023) sit alongside that line. These

models face a tradeoff between recall and throughput, which Arora et al. (2024a; 2024b) mapped empirically and theoretically.

Structural obstructions, assembled here. Our five invariants assemble obstructions that were previously studied in isolation. Hahn (2020) proved the first impossibility results for hard attention, on parity and on 2DYCK, the Dyck language with two types of brackets. His proof is a restriction argument, and that argument is the ancestor of the influence-set reasoning behind PDeg. Sanford, Hsu, and Telgarsky (2023) proved an embedding–precision lower bound for sparse averaging and a head–embedding–precision tradeoff for triple detection by communication complexity. On sparse averaging they also proved $\Omega(N)$-bit lower bounds on the state of recurrent networks. They extended the program to depth separations (Sanford et al., 2024a;b). Their bounds are capacity bounds based on precision, and the analytic branch of KRank uses a bound of that style.

The remaining sources concern the state bound, the mask and the multi-head case. Arora et al. (2024a) introduced the multi-query associative-recall benchmark. In a companion paper (Arora et al., 2024b) they proved an $\Omega(N)$-bit state bound for causally dependent recurrent models solving multi-query associative recall with bounded state entries. Bhattamishra, Patel, and Goyal (2020) showed that positional masking alone, with no positional encoding, already makes a transformer Turing-complete. It shows that a mask can supply positional structure even without a positional embedding; our comparisons therefore use a common mask. Multi-head extensions of the rank-counting argument relate to the low-rank bottleneck of Bhojanapalli et al. (2020) and to the rank tradeoffs of Amsel, Yehudai, and Bruna (2025).

Our contribution is not any single new lower bound but a meta-theoretic reduction. It shows each of these obstructions to be an axis of a common structure rather than an isolated fact. On the four canonical tasks under the theorem's hypotheses, any cost separation beyond polylogarithmic factors factors through one of the five axes.

Associative memory, and the capacity of a separation function. The dense-associative-memory literature reached the shape of our KRank dichotomy first, in a different model. It is the nearest prior art for Theorems B.3 and D.6. In this paragraph and the next, n is the degree of an interaction and N the number of neurons, as in the papers cited, and not the sequence lengths of the rest of this paper. For a polynomial interaction $F(x) = x^n$ over N neurons, Krotov and Hopfield (2016) derived $K_{\max} \approx \alpha_n N^{n-1}$ at a fixed single-neuron retrieval-error threshold. Demircigil et al. (2017) distinguish two dynamics: their polynomial-capacity theorem concerns a related tensor-interaction update, while their exponential-capacity theorem concerns the generalized Hopfield update with $F(x) = \exp(x)$. For the latter they prove storage of $M = \exp(\alpha N) + 1$ independently uniform random patterns in $\{-1, 1\}^N$ for sufficiently small α, under their retrieval-error conditions.

Millidge et al. (2022) state the pair as one principle. In that principle, higher polynomial degrees raise the capacity as N^{n-1}, while exponential separation functions such as the softmax give exponential capacity. Ramsauer et al. (2021) supply the bridge to the subject of this paper, by showing that the update rule of a continuous-state modern Hopfield network is the attention mechanism. Read in our direction, the exponential half has the same logarithmic scaling as the $O(\log r)$ width of Theorem A.3(b), which is $\Theta(\log r)$ at bounded precision by the count of Theorem D.6. The polynomial half is a width that grows as a root of the capacity, the scaling that Theorem B.3 records. Theorem D.9 says what remains ours: the lower-bound half in the attention model, rather than the phenomenon.

Universality, and why it is a different question. For fixed sequence length, transformers approximate continuous sequence-to-sequence maps on compact domains under the positional-encoding assumptions of Yun et al. (2020). Idealized hard-attention encoder–decoder models with unbounded autoregressive computation are Turing-complete (Pérez et al., 2019); positional masking also yields Turing completeness without positional encodings in the model of Bhattamishra et al. (2020). These results concern different models and resource regimes from the finite, fixed-depth comparisons here.

Circuit complexity of transformers. A parallel literature places transformers within circuit classes. Hao, Angluin, and Frank (2022) placed unique-hard-attention and generalized unique-hard-attention transformers in non-uniform AC^0. Merrill and Sabharwal (2023a) sharpened the log-precision case to logspace-uniform TC^0, through the parallelism tradeoff. Saturated attention aver-

ages over the argmax set, and Merrill, Sabharwal, and Smith (2022) placed it in TC^0. Strobl (2023) tightened that placement to uniform TC^0. Average-hard attention has broader language-recognition behavior than unique hard attention, while Table 1 records a separate distinction on the PDeg axis. The survey of Strobl et al. (2024) catalogs this landscape. This perspective is complementary to ours. Circuit bounds provide a separate, coarser constraint on language recognition; our invariants track the resources used in particular tasks.

Exact logical and algebraic characterizations. A distinct program studies transformer expressivity through formal logic and algebra, sometimes obtaining exact characterizations. Chiang, Cholak, and Pillay (2023) identified a first-order logic with counting quantifiers that brackets the class from both sides. Their logic is an upper bound for fixed-precision encoders and a lower bound for general ones. Merrill and Sabharwal (2023b) translated transformers into first-order logic with majority quantifiers. Yang, Chiang, and Angluin (2024) characterized unique-hard-attention transformers with strict one-sided masking and no positional embeddings as recognizing exactly the star-free languages. Li and Cotterell (2025) characterized fixed-precision soft-attention transformers with strict future masking and no positional encodings. Such transformers recognize exactly the fragment of linear temporal logic with a single temporal operator, the past operator. In a sequel, Li and Cotterell (2026) prove that adding local attention introduces a second temporal operator and strictly enlarges the class of recognizable regular languages. Global and local attention are therefore expressively complementary. That result is the function-class face of the reach paragraph of Section G, where the window enters the cost accounting through PDeg and through reach, the set of positions the output can depend on, rather than through a new coordinate.

For causal, position-free transformers with constant precision and fixed left-to-right evaluation, Nowak, Cotterell, and Boumasmoud (2026) give semigroup upper bounds for width-one sliding-window, sharp soft, and ordinary soft attention. Their bounds are tight under an explicit free-wiring assumption. Yang, Cadilhac, and Chiang (2025) give a depth hierarchy inside the C-RASP fragment. That hierarchy is the function-class face of the depth axis, which this paper leaves outside its theorem. Sälzer et al. (2026) characterize the average-hard-attention transformers with neither positional encodings nor masking exactly, as the semialgebraic counting properties. They also show that softmax transformers express every such property. This characterization is the function-class counterpart of the Sym axis. A position-free bidirectional architecture is permutation-equivariant; for permutation-invariant language recognition, its decision depends on the token multiset.

Some of these are completeness results of a different kind from ours. The logical and algebraic programs characterize the function class that an architecture computes. They ask which languages or functions are expressible at all. Our theorem characterizes the axes of resource-cost separation. It asks which structural property determines the cost of Theorem 2.2 when one architecture is more efficient than another. The two questions are orthogonal: a characterization fixes what is computable, whereas the theorem fixes which invariant governs the cost of computing it.

Svete, Merrill, Cotterell, and Sabharwal (2026) ask the function-class form of our question directly: which architectural choices matter for padded transformers. For polynomially padded, L-uniform fixed-point models, they find that numeric precision and depth matter, while the choice between softmax and average-hard attention and width beyond logarithmic do not. These are circuit-class results under their padding and uniformity assumptions. Table 2 gives softmax and average-hard attention identical rows. The width separations that the invariants witness, such as width N against width $O(\log N)$ on AR_N, compare the cost of a task both architectures can realize. Neither program subsumes the other, and a reader should not read the completeness theorem as a function-class characterization.

Composition, depth, and communication complexity. Peng, Narayanan, and Papadimitriou (2024) prove communication-complexity lower bounds for one transformer layer on a two-function composition task. Chen, Peng, and Wu (2025) prove unconditional separations between depth L and depth $L+1$ on L-step composition tasks. Zubić et al. (2026) carry the composition benchmark to multi-layer state-space models. For explicit-table inputs under their blockwise encoding, consider an L-layer model that composes $L+3$ functions with error at most $1/3$. Its state dimension d and per-scalar precision p must satisfy $d^2p = \Omega(N/L^3)$. In that bound N is the size of the domain of each composed function. Here d and p are their state dimension and precision, not the width and degree of this paper. On regular-language recognition and graph connectivity,

Merrill and Sabharwal (2025) show that logarithmic depth is more efficient than scaling width or adding a logarithmic number of intermediate tokens under their uniformity assumptions. Yehudai, Amsel, and Bruna (2025) bound the cost of compositional reasoning for transformers, recurrent models and chain of thought together. The depth phenomenon is orthogonal to the five width axes. At fixed five-invariant profiles, depth L and depth $L + 1$ can still separate; the theorem does not claim completeness for depth separations.

State tracking and state-space models. The State invariant captures the bandwidth bottleneck of recurrent architectures. Liu et al. (2023) construct logarithmic-depth transformers that simulate finite automata exactly and report that standard training finds shortcuts on their tested automata. Their shortcuts have depth $O(\log T)$ on inputs of length T, where a recurrence takes one step per position. Depth is not one of the five axes, as the previous paragraph says.

Bhattamishra, Hahn, Blunsom, and Kanade (2024) separate transformers from recurrent models on index lookup. One transformer layer of width $O(\log N)$ performs it, while a recurrent model with p bits of precision needs a hidden state of width at least N/p. Here p is their precision, not the degree of this paper. Merrill, Petty, and Sabharwal (2024) place log-precision generalized linear state-space models with diagonal or simultaneously diagonalizable transitions and TC^0-computable input-dependent maps in TC^0. On the selective side, Cirone et al. (2024) relate the hidden state of an input-controlled recurrence to the signature of the input path. Sarrof, Veitsman, and Hahn (2024) give the formal-language picture. Alsmann, Noori, and Lange (2026) refine that picture by gating mechanism, through fragments of linear temporal logic. Jelassi et al. (2024) separate transformers from state-space models on copying. What separates a selective recurrence from a linear one at equal state dimension is not the State axis, and this paper does not treat that separation. The theorem compares the two state-space rows of Table 1 against the causal softmax row, where State is what binds.

Sources for the constructions. Two of our upper bounds are standard constructions rather than new ones. The schedule of Theorem A.9 is recursive doubling. It is the adder schedule of Kogge and Stone (1973) and the scan of Hillis and Steele (1986), in the textbook form of JáJá (1992). The analytic upper bound of Theorem A.3(b) uses a Johnson–Lindenstrauss-style random-sign construction (Johnson & Lindenstrauss, 1984; Achlioptas, 2003), with a Hoeffding tail bound (Hoeffding, 1963).

In attention, the doubling schedule is the construction of Liu et al. (2023). Their Theorem 1 evaluates the prefix compositions of any semiautomaton on inputs of length T by a binary tree of depth $\lceil \log_2 T \rceil$, with the attention selecting the pairs and a feedforward network composing them. The construction is written for T a power of two, and it reaches other lengths by padding. Rizvi-Martel et al. (2024) simulate weighted automata the same way, with hard attention and bilinear layers, at depth $\log_2 T$ for T a power of two and by padding otherwise. Theorem A.9 is the summation case of the construction of Liu et al., carried out by attention alone and without padding positions, under either mask. Its part (c) reaches depth $\lceil \log_2 n \rceil$ at every length n. It runs the doubling around a cycle of length n and halves the values that the read position would otherwise count twice. The linear-size prefix circuits of Ladner and Fischer (1980) and of Brent and Kung (1982) save the $n \log n$ additions that recursive doubling spends. Attention has no use for that saving, since every position attends in parallel. For positive-definite kernels, the feature-space view underlying this factorization is the reproducing-kernel correspondence of Aronszajn (1950).

G Worked Readings

This appendix collects the worked example, the row-by-row reading of Table 1, and the expository paragraphs that the page limit moved out of the body.

A worked micro-example. To make Eq. (1) concrete, consider softmax attention at width $d = 2$ on an input of three tokens, $X = (x_1, x_2, x_3)$. Take all three parameter matrices to be the identity, $W_Q = W_K = W_V = I$, so that the query, the key and the value at each position are the token itself. At query position 1 the layer first computes the scores $s_{1j} = \langle x_1, x_j \rangle$, one for each $j \in \{1, 2, 3\}$. It then normalizes the scores to the weights $a_{1j} = e^{s_{1j}}/(e^{s_{11}} + e^{s_{12}} + e^{s_{13}})$. Finally it outputs

$\mathrm{Att}(X)_1 = a_{11}x_1 + a_{12}x_2 + a_{13}x_3$, a convex combination of the value vectors. Now take $x_1 = x_2 = (\lambda, 0)$ and $x_3 = (0, \lambda)$, for a real scale λ. Then $s_{11} = s_{12} = \lambda^2$ and $s_{13} = 0$. The weights are therefore $a_{11} = a_{12} = e^{\lambda^2}/(2e^{\lambda^2} + 1)$ and $a_{13} = 1/(2e^{\lambda^2} + 1)$. As λ grows, the weight on position 3 decays like $e^{-\lambda^2}$, while positions 1 and 2 share the rest. The difference between the output and $(\lambda, 0)$ therefore tends to 0, and the output moves away from x_3. The vector $(\lambda, 0)$ is the common value of the two positions whose keys align with the query. The layer thus selects by agreement of inner products, and the scale of the representations sets how sharp the selection is. This selective readout is the mechanism that all five invariants constrain, each in a different way.

The head count. The head count H does not appear in the cost of Eq. (2), and this is a consequence of the convention rather than an omission. An H-headed layer of total width d carries query, key and value matrices $W_Q^{(h)}, W_K^{(h)}, W_V^{(h)} \in \mathbb{R}^{(d/H)\times d}$ and an output map $O_h \in \mathbb{R}^{d\times(d/H)}$ for each head h. Each matrix has d^2/H entries, so the H sets hold $4d^2$ matrix entries however H is chosen. How many heads there are decides how the width is partitioned, not how much of it there is. This is why Theorem D.12 and Theorem D.19 find that multi-head structure contributes no invariant of its own and instead changes the rate at which the width axis binds. Eq. (2) has three architectural parameters: the width d, the depth L and the state dimension d_h. Together with the partition parameter H, they are the levers an architecture can pull. The invariants of Section 3 are the structural quantities that determine how each lever must scale with a task.

The two retrieval tasks. In AR_N the key is carried as token content: each key is bundled with its value into one token. This encoding differs from the multi-query formulation of Arora et al. (2024a). In their formulation a query token repeats an earlier token, and the target is the token immediately following that earlier occurrence. The two formulations are not interchangeable. The encoding here makes the target a function of the token multiset together with the query token, whereas theirs reads an adjacency and so depends on the order of the sequence. Of the retrieval tasks in this paper, the one that addresses by position is Index_n, and it is the one whose target moves when the positions are permuted.

The standard class. The standard architecture class of Theorem 2.4 is deliberately broad on the score functional and deliberately narrow on the embedding. The breadth on the score functional lets a single statement cover softmax, hard, polynomial, sigmoid and sliding-window attention at once. Condition (i) asks only that the score functional be real-analytic or κ-selective, the dichotomy that the invariants KRank and PDeg are built to read. Rotary and additive relative-position schemes fall under condition (ii), since they are fixed functions of position. Gated and gated-linear readouts fall under condition (iii), as token-wise maps of bounded complexity. The narrowness on the embedding in condition (ii) is essential and not a technicality. An embedding allowed to read the whole sequence can compute the task itself and hand the answer to a single attention layer of width $O(1)$. The structural class of that layer then decides nothing.

All score functionals in Eq. (1) satisfy Theorem 2.4, and so do linear and selective SSMs under the recurrent discipline. Multi-head variants fit by allowing H parallel heads, each of width d/H, whose outputs are aggregated additively. Condition (v), the mask, is recorded because two architectures that differ only in their mask are not comparable by the invariants of this paper. The theorem is stated for architectures sharing a mask. That restriction cannot be dropped. In our taxonomy, the composition tasks of Chen, Peng, and Wu (2025) are ones on which a bidirectional encoder beats a causal decoder while holding no more of any of the five invariants.

Taxonomy table. Table 1 lists the five-invariant profiles of the standard architectures. Reading across a row gives the structural signature of an architecture, and reading down a column shows which architectures sit at which value of an invariant. Every entry is a value rather than a verdict: which axis binds is decided by the task, not by the row. Hard attention has $\mathrm{KRank} = \infty$, since the argmax matrix can have full rank, and its piecewise structure shows up in PDeg. The state bottleneck of a state-space model shows up in State, and it leaves the kernel rank of the model unconstrained. The sliding-window entry in the PDeg column is the receptive-field bound of Theorem A.6. It is the one place where the mask, rather than the score functional, supplies the finite value.

Two entries deserve a word, because the obvious guess for each is wrong. Average-hard attention has $\mathrm{PDeg} = \infty$, not 2^L. It is hard attention that is 1-selective, whereas the argmax set an average-hard layer averages over can be the whole of $[n]$, and it is the whole of $[n]$ when all scores are equal. Sigmoid attention has $\mathrm{Bound} = \infty$, that is, $\mathrm{Bound}(A, 1) \geq n$, because it is unnormalized. Being unnormalized is exactly what lets it count at depth 1, uniformly in the length, and a normalized sigmoid would sit with softmax. The two state-space rows have $\mathrm{Bound} = \infty$ even under a power-bounded transition matrix R, one whose powers R^m have norm bounded uniformly in m. The reason is that the identity $R = I$ is power-bounded, while with it the state $h_t = \sum_{j \leq t} Ex_j$, where E is the input map, grows with the length.

Binding axis on associative recall. This paragraph applies the table to the running example AR_N. That task is a linear-retrieval task of rank N, with $\mathrm{KRank}(\mathrm{AR}_N) = N$, $\mathrm{Sym}(\mathrm{AR}_N) = S_n$ and $\mathrm{State}(\mathrm{AR}_N) = N$. For each architecture, the table predicts the binding invariant and the asymptotic cost by identifying the most-constrained axis in its row.

- Softmax and sigmoid have an analytic super-polynomial kernel, so the binding axis is KRank in its infinite-rank regime. The width is $O(\log N)$ by Theorem A.3(b), and $\Theta(\log N)$ at bounded precision by Theorem D.6.
- Hard and average-hard attention are not constrained on AR_N at all. The target reads a single value coordinate, so $\mathrm{PDeg}(\mathrm{AR}_N) = 1$, which a depth-1 layer already meets, and the argmax selects the matching key directly. Both realize AR_N at depth 1 and width $O(\log N)$. The task that separates hard attention from softmax and from average-hard attention is Avg_n, not AR_N. On Avg_n, average-hard attention costs what softmax costs (Table 2).
- Polynomial attention of degree p has a kernel of finite rank, so the binding axis is KRank, and when $N \geq (2e)^p$ the width is $\Theta(p \cdot N^{1/p})$, by Theorem A.3(a) for the lower bound and Theorem A.4 for the upper bound.
- Sliding-window softmax of half-width w fails AR_N whenever $Lw < N$, and the reason is reach, the set of positions on which the output can depend (defined in the next paragraph), rather than PDeg. The target reads one coordinate, which the PDeg gate already grants, but that coordinate can lie anywhere in $[N]$, while the read position, the last one, sees only the Lw positions before it.
- NoPE softmax clears the Sym gate on AR_N, since $\mathrm{Sym}(\mathrm{AR}_N) = S_n$. The key travels with the value as token content, so the target is a function of the token multiset and the query token, and a permutation-equivariant architecture can compute it. NoPE softmax realizes AR_N at width $O(\log N)$, alongside the other analytic rows. The task on which Sym binds is Index_n, where the key is the position. There $\mathrm{Sym}(\mathrm{Index}_n) = \{e\}$, while NoPE softmax has $\mathrm{Sym} = S_n$, which is not contained in $\{e\}$, so the task is unrealizable at any cost.
- Linear and selective SSMs have binding axis State. Their state dimension is at least N, by Theorem A.7. It is at most N for the selective row and $O(N \log N)$ for the linear row, by the constructions in the proof of Theorem 4.2(ii).

Reading the binding axis off the table therefore sorts the architectures of Table 1 into three cost regimes on AR_N. The first regime is logarithmic width, for the analytic kernels, NoPE softmax included, and for the two hard variants. The second is width $\Theta(pN^{1/p})$, for polynomial attention. The third is state $\Theta(N)$ up to a logarithmic factor, for the recurrences. The windowed row is the exception, and it is the subject of the next paragraph.

Reach, and why masked rows are compared only to each other. A mask fixes not only how many positions the read position may depend on but also which ones. Write $\mathrm{Reach}(A, L) \subseteq [n]$ for the set of input positions on which the depth-L output at the read position can depend. For a bidirectional mask this set is all of $[n]$, and so it is for a causal mask read at the last position. For a sliding window it is the set of $Lw + 1$ positions ending at the read position, which is the last one. A task whose target can depend on a position outside $\mathrm{Reach}(A, L)$ is unrealizable, however large $\mathrm{PDeg}(A, L)$ is. On a sliding window, AR_N is exactly that case. The size gate, the gate on PDeg, asks for one coordinate and is met by every row, while the coordinate in question ranges over all of $[N]$. Reach depends only on the mask and the depth. On every row the theorems compare, the mask

lets the read position see every input position, so $\mathrm{Reach}(A, L)$ is all of $[n]$ at every depth L and never appears in those theorems. Reach is also the reason those comparisons are made at a common mask.

The sliding-window entry. For sliding-window attention the generic bound $(2w+2)^L$ is far from tight, because the active sets are contiguous and overlap. At depth L the influence set is an interval of length at most $2Lw + 1$ at an interior position, and of length $Lw + 1$ at the last position, where the read position is. That last length is the receptive-field bound, and it is the value Table 1 records.

Reading the cost table. Each of the five structural axes is visible in Table 2, and each in a different shape. Among the bidirectional rows, the kernel rank is the one gradient in the AR_N column: every row other than the polynomial one sits at $\log^2 N$, and the polynomial row at $d_p(N)^2$, which is $\Theta(p^2 N^{2/p})$ once $N \geq (2e)^p$. That gradient is Theorem B.3 read across the degrees, and Theorem B.2 read at $p = 1$. The piecewise degree is the single finite gap in the Avg_n column: the $\log n$ of hard attention against the 1 of every other row. The output range is invisible within the per-length Count_n column, which is flat up to polylogarithmic factors, and appears only in the length-uniform column. The rows that move to ∞ there are exactly those with $\mathrm{Bound}(A, L) = 2^{\Theta(L)}$, and this is what it means for the axis to bind only under length-uniform realization. The symmetry axis appears once, as the ∞ in the NoPE row of the Index_n column, and Theorem 4.3 is why it appears there and not one column to the left. The state dimension is the gap between the two state-space rows and the causal softmax row above them, n^2 against $\log^2 n$, and it opens on the retrieval tasks alone.

Of these five gaps, part (iii) of Theorem 4.2 names only three, those of KRank, Sym and State. The other two, those of PDeg and Bound, are polylogarithmic in the per-length columns, below the resolution of part (iii) (Theorem D.18). The length-uniform column, where the gap of Bound is infinite, lies outside part (iii), which is a statement about realization at each length.