SOFTMAX TRANSFORMERS ARE DOUBLY EXPONENTIALLY SUCCINCT

Anonymous authors
Paper under double-blind review

ABSTRACT

We prove that fixed-precision softmax transformers can be doubly exponentially more succinct than finite automata, resolving an open question. Over a fixed alphabet, we construct constant-depth transformers with causal masking, one head per layer, no positional encodings, width and precision $O(n)$, and description length $O(n^3)$. The smallest equivalent NFAs and DFAs each have $2^{\Theta(2^n)}$ states, although every accepted word has length $\Theta(n2^n)$. The state lower bound comes from the 2^{2^n} Boolean tables encoded by prefixes. The separation also holds for fixed-precision average-hard attention. The same language requires description size $2^{\Theta(n)}$ in C-RASP$_+$, a counting-based formalism expressively equivalent to $(1, 1)$-precision transformers. This resolves a second open question by ruling out a polynomial-size translation into C-RASP$_+$. A direct finite-state simulation yields a doubly exponential DFA upper bound and singly exponential bounds on shortest accepted and distinguishing words. Finally, we show that nonemptiness is NEXP-complete under the same architectural restrictions.

1 INTRODUCTION

How compactly can a transformer represent a regular language? Succinctness compares the sizes of descriptions that agree on all input lengths. We study softmax transformers whose precision is fixed independently of input length and included in the description length.

Bergsträßer et al. (2026) separate unique-hard-attention transformers from finite automata using doubly exponentially long shortest accepted words. They leave the fixed-precision softmax and average-hard cases open. Yang et al. (2026) show that $(1, 1)$-precision transformers have tight singly exponential length complexity. Thus shortest accepted-word length alone cannot establish a doubly exponential state lower bound. Our direct finite-state simulation gives a doubly exponential DFA upper bound and singly exponential bounds on shortest accepted and distinguishing words under downward rounding and round-to-nearest (Proposition 9, Corollary 10). Our family still needs doubly exponentially many states (Figure 1).

Succinctness over automata. Our language checks records over $\{0, 1, @, \#\}$: repeated n-bit keys must have the same Boolean value. Prefixes encode 2^{2^n} possible Boolean tables. An automaton must distinguish these tables in its state; attention can compare each record with earlier records. Splicing accepting runs gives a 2^{2^n}-state lower bound even for NFAs. A constant-depth softmax transformer recognizes the language with description length $O(n^3)$, while the smallest equivalent NFAs and DFAs have $2^{\Theta(2^n)}$ states (Theorems 4 and 8). Every accepted word has length $\Theta(n2^n)$. The construction also works under fixed-precision average-hard attention (Corollary 20).

Succinctness over C-RASP$_+$. A language may require a large automaton yet admit a compact counting program. Yang et al. (2026) prove expressive equivalence between $(1, 1)$-precision transformers and C-RASP$_+$, the positive fragment of Counting RASP. They leave polynomial-size translation from transformers to C-RASP$_+$ open. We show that, under their shared-definition size measure with binary-encoded constants, a C-RASP$_+$ program of size m has a DFA with $2^{O(m)}$ states. Every C-RASP$_+$ program recognizing our language therefore needs size $\Omega(2^n)$, and a direct construction has size $2^{O(n)}$. The same language thus has minimum C-RASP$_+$ size $2^{\Theta(n)}$, ruling out polynomial-size translation (Theorem 12). Our downward-rounding construction belongs to their model.

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.

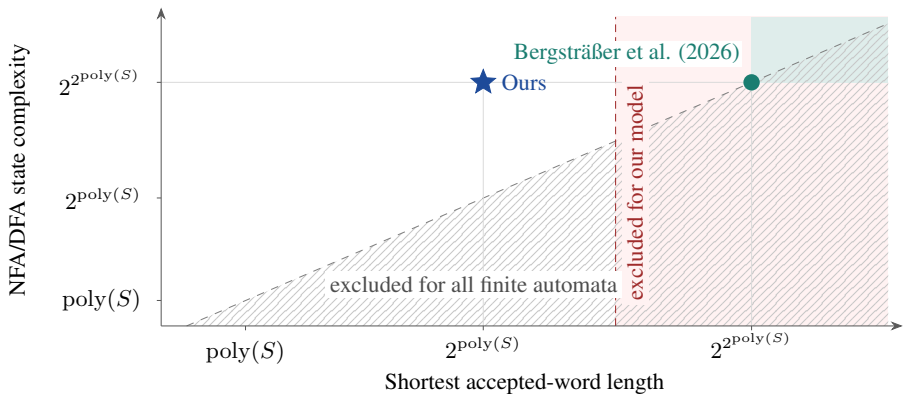

Figure 1: **Singly exponential word length, doubly exponential state complexity.** Growth in description size S, up to polynomial rescaling. An N-state automaton accepting a nonempty word accepts one of length at most N (hatched region). Shortest nonempty accepted words in our model have length at most $2^{\mathrm{poly}(S)}$ (shaded exclusion, Corollary 10). Our family has doubly exponential state complexity and matching bounds on both axes. The green quadrant shows the lower bounds of Bergsträßer et al. (2026).

Nonemptiness. Sälzer et al. (2025) prove NEXP-hardness of deciding whether a fixed-precision transformer with average-hard attention and positional encodings accepts any word. They leave open whether similar hardness holds for softmax. For our causally masked softmax model, we prove NEXP-completeness with constant depth and no positional encodings (Theorem 13).

Proof strategy. Our consistency recognizer compares records written one bit per token. At each valid boundary, one attention head reconstructs the record bits in separate value coordinates. A dot-product score selects BOS and exactly the earlier records with the same key and opposite value. Rounding sets all other unnormalized attention weights to zero. Signed threshold tests remain correct even when normalized coefficients round to zero.

2 Preliminaries

We write $\mathbb{N} \coloneqq \{1, 2, \ldots\}$, $\mathbb{N}_0 \coloneqq \mathbb{N} \cup \{0\}$, and $[m] \coloneqq \{1, \ldots, m\}$. All logarithms are base two. A word $w = a_1 \cdots a_N$ is a finite sequence over an alphabet Σ. The set Σ^* contains all finite words over Σ, including the empty word ε. A language is a subset of Σ^*. For words u, v, their concatenation uv is u followed by v. We distinguish the family parameter n, input length $N \coloneqq |w|$, and description length $S \coloneqq |T|$. Bold letters denote vectors or matrices. The functions $\mathrm{ReLU}(x) \coloneqq \max\{0, x\}$ and $\mathrm{clip}_{[0,1]}(x) \coloneqq \min\{1, \max\{0, x\}\}$ act coordinatewise. We write $\mathbb{1}\{\, P \,\}$ for the indicator of P.

2.1 Language representations

A finite automaton reads a word from left to right using a finite set of states. A run follows allowed transitions from the initial state and is accepting if it ends in an accepting state after reading the whole word. A deterministic finite automaton (DFA) has one run on each word; a nondeterministic finite automaton (NFA) may have several and accepts whenever one is accepting. Languages recognized by finite automata are regular. For a regular language L, let $\mathrm{nfa}(L)$ and $\mathrm{dfa}(L)$ denote the minimum numbers of states of an NFA and DFA recognizing L.

For any representation A, let $L(A)$ be its language and $|A|$ its description length. We use a dense encoding of transformers: every entry of the weight matrices, bias vectors, and token embeddings, including zeros, is stored using p bits. Dimensions, layer count, and the precision parameters p, s are recorded in binary. Thus a dense $d \times d$ matrix at p-bit precision has description length $O(d^2 p)$.

Definition 1 (Succinctness (Bergsträßer et al., 2026))**.** A representation class $\mathcal{C}$ is f-more succinct than $\mathcal{D}$ if there are representations $A_n \in \mathcal{C}$ with $|A_n| \to \infty$ such that every equivalent $B_n \in \mathcal{D}$

satisfies $|B_n| \geq f(|A_n|)$. It is exponentially more succinct if this holds for some $f(S) = 2^{\Omega(S^c)}$, with $c > 0$, and doubly exponentially more succinct if it holds for some $f(S) = 2^{2^{\Omega(S^c)}}$.

For automata, we first bound the number of states, independently of how the automaton is encoded.

2.2 Fixed-precision softmax transformers

The notation $(1, 1)$ in Yang et al. (2026) assigns $O(1)$ bits as functions of input length, both outside and inside attention. Each T_n fixes its precision for every input length. A bound $p_n = O(n)$ describes variation across the family, not growth while processing a longer word.

Numerical representation. Fix integers $p > s \geq 1$. We use p total bits, of which s are fractional bits. Put $\delta \coloneqq 2^{-s}$ and define

$$\mathbb{F}_{p,s} \coloneqq \{m\delta : -2^{p-1} \leq m < 2^{p-1},\ m \in \mathbb{Z}\}. \tag{1}$$

The quantizer round rounds to this grid, clipping at its endpoints. On vectors it acts coordinatewise. We take downward rounding as the primary convention. The same construction works with round-to-nearest, with any fixed tie-breaking rule. For decision problems, the tie rule is fixed and polynomial-time computable. A transformer fixes p, s, and its quantizer independently of input length. Affine maps and dot products are evaluated exactly and then rounded. ReLU is applied to the resulting grid values.

Attention and layers. The model supplies a beginning-of-sequence marker BOS at position 0, before the input word w. Ordinary positions are those of w. The initial state $\mathbf{h}_i^0 \in \mathbb{F}_{p,s}^d$ is the symbol embedding, with a separate embedding for BOS. At layer ℓ, affine maps of $\mathbf{h}_i^{\ell-1}$ give attention query, key, and value vectors $\mathbf{q}_i, \mathbf{k}_i, \mathbf{v}_i$. We suppress ℓ on all attention quantities below. The single head sees exactly positions $0 \leq j \leq i$, called visible positions. Write

$$S_{ij} \coloneqq \mathrm{round}(\langle \mathbf{q}_i, \mathbf{k}_j \rangle), \quad U_{ij} \coloneqq \mathrm{round}(e^{S_{ij}}), \quad Z_i \coloneqq \sum_{j=0}^{i} U_{ij}, \tag{2}$$

$$\alpha_{ij} \coloneqq \mathrm{round}(U_{ij}/Z_i), \quad \mathbf{c}_i \coloneqq \mathrm{round}\left(\sum_{j=0}^{i} \alpha_{ij}\mathbf{v}_j\right), \quad \mathbf{h}_i^{\ell} \coloneqq f^{\ell}(\mathbf{h}_i^{\ell-1} + \mathbf{c}_i). \tag{3}$$

The coefficients are used as computed, so $\sum_{j=0}^{i} \alpha_{ij}$ may differ from 1. The denominator and context sums are exact, with an equivalent $O(p)$-bit accumulator implementation, possibly wider than $\mathbb{F}_{p,s}$ (Proposition 25). If $Z_i = 0$, we set the context to zero. In the constructed networks, BOS always has score zero and hence $Z_i > 0$. Each f^{ℓ} is an affine–ReLU–affine map, with rounding and clipping after each affine map. The exact residual sum may exceed the grid range. It enters f^{ℓ} directly, whose first affine output is rounded and clipped. A fixed number of such layers implements a constant-depth position-wise circuit. Exponentials and normalized coefficients are rounded as in the $(1, 1)$ configuration of Yang et al. (2026).

The transformer accepts when a designated coordinate at the final position is positive, using position 0 for the empty word. In our constructions this coordinate is always Boolean. All weights are dyadic. The model has no positional encodings or layer normalization. Its width bounds the dimensions of hidden states, queries, keys, and values, and the hidden dimensions of the feed-forward networks. BOS is outside the input alphabet. Appendix A.2 gives the compilation of our circuits into these layers.

3 Functional consistency

Fix $n \geq 1$, let $M \coloneqq 2^n$ and $R \coloneqq 2M$, and take the fixed alphabet $\Sigma \coloneqq \{0, 1, @, \#\}$. For a record value $v \in \{0, 1\}$ and a record key $x = (x_1, \ldots, x_n) \in \{0, 1\}^n$, define

$$r(v, x) \coloneqq v x_1 \cdots x_n @ \#. \tag{4}$$

Definition 2 (Consistency language)**.** The language L_n consists of exactly R records such that equal record keys have equal record values. Superscripts index records, and the product concatenates them in order:

$$L_n := \left\{ \prod_{j=1}^{R} r(v^{(j)}, x^{(j)}) : x^{(j)} = x^{(k)} \Longrightarrow v^{(j)} = v^{(k)} \text{ for all } j, k \in [R] \right\}. \tag{5}$$

Every word in L_n has length $2^{n+1}(n+3)$. The records describe a partial Boolean function: a key may occur repeatedly or not at all. The choice $R = 2M$ accommodates the lower-bound words below, with M records encoding a full table followed by M records that copy or query it.

Example ($n = 2$)**.** In key order $00, 01, 10, 11$, take tables $f := (0, 1, 0, 1)$ and $g := (0, 1, 1, 1)$. Their record lists are

$$u_f = \texttt{000@\#}\ \texttt{101@\#}\ \texttt{010@\#}\ \texttt{111@\#},$$
$$u_g = \texttt{000@\#}\ \texttt{101@\#}\ \texttt{110@\#}\ \texttt{111@\#}.$$

Spaces are for readability, not input symbols. The word $u_f u_f$ contains two copies of every key, and both copies carry the same value $f(x)$. Hence $u_f u_f \in L_2$. In $u_f u_g$, key 10 appears as `010@#` in the first half and `110@#` in the second. These records assign different values to the same key, so $u_f u_g \notin L_2$. The following argument applies this distinction to all 2^{2^n} tables, including for nondeterministic automata.

Proposition 3. *For every $n \geq 1$,*

$$2^{2^n} \leq \mathrm{nfa}(L_n) \leq \mathrm{dfa}(L_n) \leq 3^{2^n} 2^{O(n)}. \tag{6}$$

Proof. For each function $f : \{0,1\}^n \to \{0,1\}$, let u_f list the records $r(f(x), x)$ in lexicographic key order. Then $u_f u_f \in L_n$. If $f \neq g$, the word $u_f u_g$ contains conflicting records and is rejected. Fix an accepting run on each $u_f u_f$. If two runs visit the same state after the first M records, their prefix and suffix runs can be spliced to accept a crossed word. Thus an NFA needs one distinct state for every f, giving 2^{2^n} states. For the upper bound, a DFA stores one of $\{\mathrm{unseen}, 0, 1\}$ for each key, together with $O(n)$ bits for parsing, the current record, and a saturated record counter. A rejecting sink handles every format error and conflict. ■

4 A CONSTANT-DEPTH RECOGNIZER

Theorem 4 (Constant-depth recognition)**.** *There is an absolute constant L_0 such that, for every $n \geq 1$, a transformer T_n of depth at most L_0 recognizes L_n. It has one head per layer, a causal mask that includes the current position, no positional encodings, $O(n)$ width and precision, and $|T_n| = O(n^3)$. The weights are constructible in polynomial time. The statement holds for downward rounding and round-to-nearest.*

How the construction works. We call an `@` position a boundary. Delimiter counts assign record addresses (Stage 1), bit ranks assign attention value coordinates (Stage 2), and validity tests check boundaries (Stage 3). One head reconstructs each valid record (Stage 4). An attention score detects earlier records with the same key and opposite value (Stage 5). Format and zero-conflict tests determine acceptance at the final position (Stages 6–10). Figure 2 follows the stages of Appendix A.3. Capped addresses, fixed count thresholds, and validity gates ensure correctness on malformed inputs.

We describe a parser for b bits per record and exactly R records, then set $b = n + 1$ and $R = 2^{n+1}$. Section 6 reuses it with different b, R. A register is a scalar stored in hidden coordinates. Token indicators and a constant-one register persist. We choose a score gap H, address scale Λ, and decoder scale D.

Lemma 5 (Parameter choice)**.** *For $b, R \geq 1$, one can compute $p > s \geq 1$ with $p, s = O(\log(R + 1) + \log(b + 2))$ and representable positive integers H, Λ, D in polynomial time such that, for either quantizer,*

$$e^{-H} < \delta/2, \qquad Rb + 1 < 1/(4\delta), \qquad D \geq b + 1, \qquad D\delta < 1/2.$$

Let $\rho_m := m\,\mathrm{round}(1/(m+1))$ be the context from m selected hashes of value 1 and BOS of value 0. Without rounding, this average is $m/(m+1)$. Then $\Lambda(\rho_m - \rho_t) \geq 4H$ for $0 \leq t < m \leq R$. All finite affine and ReLU computations in the construction are exact on the grid and remain within range.

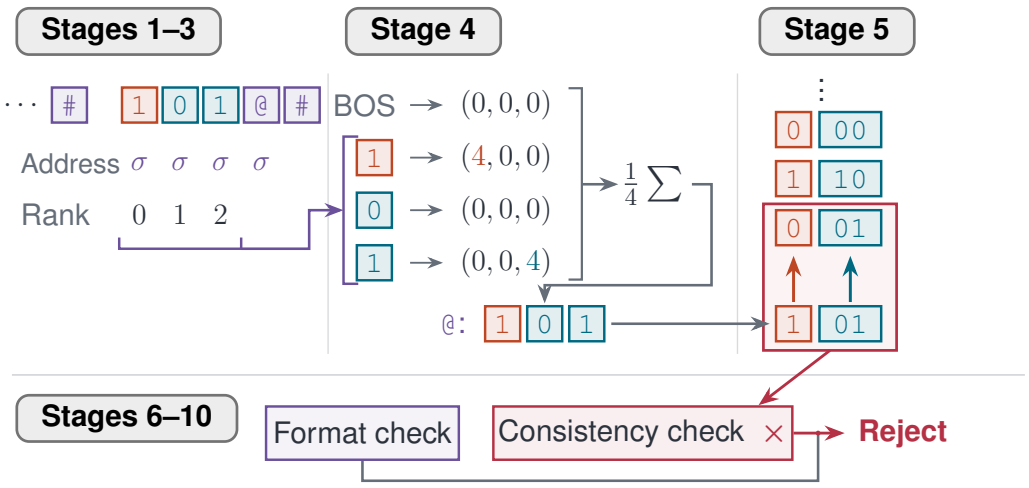

Figure 2: **From serialized bits to a consistency decision.** Here $n = 2$, $b = 3$, $D = 4$, and $R = 8$. Record values are orange and record keys are teal. Stages 1–3 assign segment addresses and bit ranks and mark valid boundaries. Stage 4 places each bit in its rank's coordinate and scales by D. Averaging with zero at BOS recovers value 1 and key 01 at @. Stage 5 compares this record with earlier records. The highlighted pair has the same key 01 and opposite values, triggering a conflict. BOS is also selected. Stages 6–10 check format and consistency. The red cross marks failure.

The first bound makes unwanted attention weights round to zero. The second preserves count tests through $Rb + 1$. The bounds on D amplify averaged bits while controlling rounding error. The address gap exceeds the H needed in (9). Appendix A.1 gives explicit choices and range bounds.

Exact selection and counting. Following Yang et al. (2026), give m targets value 1 and BOS value $-(t - \frac{1}{2})$. Their unrounded average $(m - t + \frac{1}{2})/(m + 1)$ is nonnegative exactly when $m \geq t$. The next lemma preserves this test under rounding. For row i, let $\mathcal{P}_i$ contain BOS and all positions of score 0, with scores at most $-H$ elsewhere. Since $e^{-H} < \delta/2$, $U_{ij} = \mathbb{1}\{\, j \in \mathcal{P}_i \,\}$ under either quantizer. We call these positions selected. Their shared normalized coefficient is $\mathrm{round}(1/|\mathcal{P}_i|)$, which may be zero.

Lemma 6. *Suppose a row has score 0 at BOS and $m \in \mathbb{N}_0$ ordinary targets, and scores at most $-H$ elsewhere, with $e^{-H} < \delta/2$. Let $t \in \mathbb{N}$ satisfy $1 \leq t \leq 1/(4\delta)$. Assign attention value $-(t - \frac{1}{2})$ to BOS and 1 to each target. Its context $\zeta_t(m)$ satisfies*

$$m < t \implies \zeta_t(m) \leq -\delta, \qquad m \geq t \implies \zeta_t(m) \geq 0. \tag{7}$$

On the δ-grid, $\mathrm{Step}_\delta(z) := \delta^{-1}(\mathrm{ReLU}(z + \delta) - \mathrm{ReLU}(z))$ is 0 for $z < 0$ and 1 for $z \geq 0$. Thus $\mathrm{Step}_\delta(\zeta_t(m))$ computes $\mathrm{GE}_t := \mathbb{1}\{\, m \geq t \,\}$ exactly.

Proof. Put $\alpha := \mathrm{round}(1/(m + 1))$. The context is $\mathrm{round}(\alpha(m - t + \frac{1}{2}))$. When $m < t$, integrality gives $m + 1 \leq t$. The bound $t \leq 1/(4\delta)$ ensures $\alpha \geq 1/t - \delta \geq 3\delta$, providing a rounding margin. The argument is therefore at most $-3\delta/2$ and the rounded value is at most $-\delta$. When $m \geq t$, both the argument and its rounded value are nonnegative, including when $\alpha = 0$. ■

Exact equality is $\mathrm{GE}_t - \mathrm{GE}_{t+1}$. Several thresholds share one head by using different value coordinates. The test handles arbitrarily large m, including when $\alpha = 0$.

Segment addresses. A boundary must select the bits in its own record. A segment consists of the non-hash positions after BOS or a hash and before the next hash or the end of the word. Let κ_i count visible hashes, including position i. Thus a hash has the address of the following segment. A head with attention value 0 at BOS and 1 at hashes returns ρ_{κ_i}. For sufficiently large m, $\mathrm{round}(1/(m+1)) = 0$ and ρ_m drops to zero, destroying address monotonicity. To prevent this, compute $g_i := \mathbb{1}\{\, \kappa_i \geq R \,\}$

using Lemma 6 and store the capped address

$$\sigma_i \coloneqq \begin{cases} \Lambda \rho_{\kappa_i} & \kappa_i < R, \\ \Lambda \rho_R & \kappa_i \geq R. \end{cases} \tag{8}$$

The cap uses a gate controlled by g_i. We have $\sigma_0 = 0$. By Lemma 5, addresses increase by at least $4H$ up to the cap and remain constant thereafter. Thus $j \leq i$ implies $\sigma_j \leq \sigma_i$ on every input.

Let $Q_{\mathrm{BOS}}(j) \coloneqq \mathbb{1}\{\, j = 0 \,\}$, and let P_j be a Boolean target predicate excluding BOS and hashes. We need score 0 at BOS and targets sharing i's address, and at most $-H$ elsewhere. The score

$$S_P(i,j) \coloneqq \sigma_j - \sigma_i + \sigma_i Q_{\mathrm{BOS}}(j) - H(1 - Q_{\mathrm{BOS}}(j) - P_j) \tag{9}$$

is a query–key dot product of affine maps. The address difference is 0 or at most $-4H$. The BOS term cancels $-\sigma_i$ at $j = 0$, and the final term penalizes ordinary nontargets by H. Thus exactly the desired positions are selected. Before the cap, same-address targets lie in the current segment. After the cap, several segments may share an address, but the score bounds still hold.

Ranks and valid boundaries. Set $P_j \coloneqq \mathbb{1}\{\, a_j \in \{0,1\} \,\}$ in (9) and apply Lemma 6 to compute G_t for $t = 1, \ldots, b+1$. Before the cap, G_t indicates at least t visible bits in the segment. Once R hashes have been seen, G_t instead tests whether at least t visible bits occur after the R-th hash. Those positions are declared invalid. At a bit position j before the cap, the count includes the current bit. For $0 \leq u < b$, the difference $G_{u+1}(j) - G_{u+2}(j)$ is 1 exactly at the $(u+1)$-th bit, which has rank u because ranks start at 0. Store

$$W_j(u) \coloneqq \mathbb{1}\{\, a_j = 1 \,\}\big(G_{u+1}(j) - G_{u+2}(j)\big). \tag{10}$$

Before the cap, $W_j(u)$ is 1 exactly when position j contains a bit of value 1 at rank u, and is 0 otherwise. These are the coordinates used in Stage 4 of Figure 2. A further head counts `@` symbols in the segment. A boundary is valid exactly when it sees b bits and exactly one `@` in its segment and has $g_i = 0$. These tests are Boolean on all inputs.

Lemma 7. *One attention head with b value coordinates recovers every record bit at every valid boundary. The decoded coordinates are zero at all other positions.*

Proof. Select all bits in the segment together with BOS. Give position j the value vector $D(W_j(0), \ldots, W_j(b-1))$ and give BOS zero. At a valid boundary i, there are $b+1$ selected positions and exactly one possible contributor to each coordinate. Writing $\xi_{i,u}$ for the actual bit at rank u in the record ending at i, the context coordinate is

$$y_{i,u} = D \,\mathrm{round}(1/(b+1))\xi_{i,u}. \tag{11}$$

For $\xi_{i,u} = 0$ this is zero, and for $\xi_{i,u} = 1$ it is at least $1 - D\delta > 1/2$. Hence $\mathrm{clip}_{[0,1]}(2y) = \mathrm{ReLU}(2y) - \mathrm{ReLU}(2y-1)$ recovers the bit. On every nonnegative input, $\mathrm{clip}_{[0,1]}(2y) \in [0,1]$. Gating it by the Boolean validity indicator therefore sets every invalid position to zero. All b value coordinates use the same attention coefficients. ■

Conflicts. Now set $b = n + 1$. Let $\beta_{i,0}$ be the decoded record value and $\beta_{i,1}, \ldots, \beta_{i,n}$ the decoded record key. Put $\nu_i \coloneqq \mathbb{1}\{\, i \text{ is a valid boundary} \,\}$, $\eta_i \coloneqq 2\beta_{i,0} - \nu_i$, and $z_{i,u} \coloneqq 2\beta_{i,u} - \nu_i$. Valid boundaries carry signs in $\{-1, 1\}$. All other positions carry zero. For two valid boundaries, let h count differing key bits. We want score $-2Hh - H\mathbb{1}\{\, \beta_{i,0} = \beta_{j,0} \,\}$, which is zero exactly for equal keys and opposite values. Since $\langle \mathbf{z}_i, \mathbf{z}_j \rangle = n - 2h$, use

$$S_{\mathrm{con}}(i,j) \coloneqq H\big(\langle \mathbf{z}_i, \mathbf{z}_j \rangle - n\big) - \frac{H}{2}(1 + \eta_i \eta_j) + (Hn + H/2) Q_{\mathrm{BOS}}(j). \tag{12}$$

The first two terms give the desired score at valid boundaries. The last cancels the baseline $-Hn - H/2$ at BOS, where $\mathbf{z}_0 = \mathbf{0}$ and $\eta_0 = 0$. Every invalid ordinary candidate has score at most $-H$. Lemma 6 with $t = 1$ thus tests for a selected non-BOS position, flagging an earlier record with the same key and opposite value. The attention value $\nu_j - Q_{\mathrm{BOS}}(j)/2$ is 1 at each selected target and $-1/2$ at BOS, as required. The current record scores $-H$ against itself.

Format. Write N_P for the number of ordinary input positions satisfying a predicate P. Four global tests require

$$N_{\#} = R, \quad N_{@} = R, \quad N_{\text{bit}} = Rb, \quad N_{\text{invalid-@}} = 0. \tag{13}$$

At the final position, Lemma 6 computes the equalities using threshold pairs $(R, R+1)$, $(R, R+1)$, and $(Rb, Rb+1)$. Negating the threshold-1 test gives the zero-invalid-boundary condition. These checks force exactly one valid @ in each segment preceding a hash and none after the final hash. The b bits preceding each valid boundary already account for all Rb bits, so there are no trailing bits or extra symbols. Thus (13) holds precisely on words of the form $(\{0,1\}^b @ \#)^R$, where the b-bit string may differ between records.

Proof of Theorem 4. The output conjoins (13) with the absence of flagged conflicts. The format argument yields exactly R records. If two records conflict, the later one is flagged by (12). If they are consistent, no boundary is flagged. The output thus recognizes L_n, including rejection of every malformed word and the empty word.

The ten attention stages use $O(n)$ registers and constant-depth position-wise circuits. Appendix A.2 compiles them into layers of width $O(n)$, with their number independent of n. Lemma 5 gives precision $O(n)$. A constant number of dense $O(n) \times O(n)$ matrices with $O(n)$ bits per entry gives $O(n^3)$ description bits, constructible in polynomial time. ■

Fixed-precision average-hard attention gives the same hidden states (Corollary 20). The same matrices also recognize L_n under two accumulation rules that round and clip after every operation (Proposition 26).

5 Succinctness and logical descriptions

5.1 Automata and witness lengths

Theorem 8 (Automata succinctness)**.** *Constant-depth fixed-precision softmax transformers are doubly exponentially more succinct than NFAs and DFAs. For the family in Theorem 4,*

$$\text{nfa}(L(T_n)) = 2^{\Theta(2^n)}, \qquad \text{dfa}(L(T_n)) = 2^{\Theta(2^n)}. \tag{14}$$

Proof. Proposition 3 gives both state bounds, and Theorem 4 gives transformer descriptions of size $O(n^3)$. ■

Linear temporal logic (LTL) describes words using symbol tests, Boolean connectives, and temporal operators such as next and until. Its exponential translation to NFAs gives an $\Omega(2^n)$ formula-size lower bound for L_n (Appendix A.4, Corollary 19). For $S_n := |T_n|$, the automata lower bound is $2^{2^{\Omega(S_n^{1/3})}}$. The following bound holds for every transformer in our model. Appendix A.4 gives the proof.

Proposition 9 (DFA upper bound)**.** *Let T have L full attention/feed-forward blocks as in* (3)*, residual hidden-state dimension at most d, and fixed-point parameters $p > s \geq 1$. Here d counts persistent hidden coordinates, independently of the internal feed-forward width. Set $K := 2^{p+s}$. Under downward rounding or round-to-nearest with any fixed tie-breaking rule,*

$$\text{dfa}(L(T)) \leq 2(K+1)^{L2^{pd}} \leq 2^{1+L(p+s+1)2^{pd}}.$$

Consequently, both $\text{dfa}(L(T))$ and $\text{nfa}(L(T))$ are at most $2^{2^{\text{poly}(S)}}$ for the dense description length $S = |T|$.

Each layer counts occurrences of at most 2^{pd} possible hidden vectors from the preceding layer. The $L2^{pd}$ counters take values $0, \ldots, K$, giving the power of $K+1$. An acceptance bit supplies the factor 2. These capped counts determine attention because relevant counts are exact or all coefficients round to zero. Their sum is only singly exponential in S. Every nonfinal symbol of a shortest witness must increase this sum. Otherwise, deleting it leaves the subsequent computation unchanged.

Corollary 10 (Short accepted and distinguishing words)**.** *Put $W_T := L2^{pd+p+s}$ for the parameters in Proposition 9. If T accepts a nonempty word, its shortest such word has length at most $W_T + 1$. If T, T' over the same alphabet recognize different languages, some word of length at most $W_T + W_{T'} + 1$ distinguishes them. These bounds are singly exponential in the transformers' description lengths under both rounding conventions, also for the average-hard convention of Corollary 20.*

5.2 Exponential succinctness over C-RASP$_+$

Counting RASP combines Boolean tests with prefix counts. We use the positive fragment C-RASP$_+$ and the shared-definition representation of Yang et al. (2026). Programs are acyclic sequences of Boolean definitions using symbol tests, Boolean connectives, and counting comparisons. Each definition may refer to earlier definitions in *program order*. Write $\overleftarrow{\#}[\psi](i) := |\{j \in [i] : \psi(j) = 1\}|$ for the count up to and including position i. To bound the size of an equivalent DFA, we make the counting comparisons explicit. We index the program's counting comparisons by a. Within comparison a, ψ_{ar} denotes its r-th counted Boolean predicate. At position i, comparison a first adds the weighted prefix counts and the constant C_a, then compares the sum with k_a:

$$\left(C_a + \sum_r \lambda_{ar} \overleftarrow{\#}[\psi_{ar}](i) \right) \bowtie_a k_a, \qquad C_a, \lambda_{ar}, k_a \in \mathbb{N}_0, \tag{15}$$

where $\bowtie_a \in \{<, \leq, =, \geq, >\}$. The result is 1 if the comparison holds and 0 otherwise. Positive refers to nonnegative numerical coefficients and constants. Boolean negation is allowed. Constants are binary-encoded. Each shared definition is stored once, with unit-size references. The final definition at the last position determines acceptance of nonempty words. A separate bit specifies the empty-word decision. Our programs for L_n reject the empty word.

Lemma 11 (Finite-state simulation of C-RASP$_+$)**.** *Every C-RASP$_+$ program φ has an equivalent DFA with at most $2^{O(|\varphi|)}$ states.*

Proof sketch. For comparison a, store values $0, \ldots, k_a$ exactly and represent every larger value by $k_a + 1$. This preserves all comparisons with k_a. Nonnegative increments allow updates from these capped values, and the definition order determines the current Boolean values. Each counter has $k_a + 2$ possible values. Together with one acceptance bit, the counters give at most $2 \prod_a (k_a + 2) = 2^{O(|\varphi|)}$ states under binary encoding. Appendix A.5 gives the full update and induction. ■

Theorem 12 (C-RASP$_+$ succinctness)**.** *The minimum C-RASP$_+$ description size of L_n is $2^{\Theta(n)}$. Since $|T_n| = O(n^3)$, every equivalent C-RASP$_+$ program has size $2^{\Omega(|T_n|^{1/3})}$. Consequently, $(1,1)$-precision transformers are exponentially more succinct than C-RASP$_+$, and there is no polynomial-size translation from $(1,1)$-precision transformers to C-RASP$_+$.*

Proof. An equivalent program φ satisfies $2^{2^n} \leq \mathrm{dfa}(L_n) \leq 2^{O(|\varphi|)}$ by Proposition 3 and Lemma 11, so $|\varphi| = \Omega(2^n)$. Appendix A.5 gives a direct $2^{O(n)}$-size C-RASP$_+$ program for L_n. The construction with downward rounding is an instance of the $(1,1)$-precision model of Yang et al. (2026, Definition B.3). Lemma 21 verifies the correspondence, including BOS. Thus a polynomial-size translation is impossible. ■

Streaming consequence. For any fixed $\epsilon < 1/2$, randomized one-pass recognition of L_n with error at most ϵ requires $\Omega(2^n)$ bits of persistent state. The one-way INDEX reduction uses a prefix to specify a Boolean table and a continuation to query one entry. Appendix A.6 gives the model and bound $B \geq (1 - h_2(\epsilon))2^n$, where B is the number of bits of persistent memory and h_2 is binary entropy. This measures the state needed to resume a one-pass recognizer; transformer inference also retains earlier token representations for attention.

6 Complexity of nonemptiness

Nonemptiness asks whether a given transformer accepts any word. Emptiness asks whether it accepts none. These questions range over all input lengths. NEXP is the class of decision problems solvable

in nondeterministic time $2^{\mathrm{poly}(S)}$, where S is the instance size. The class coNEXP consists of the complements of NEXP problems.

To prove NEXP-hardness of nonemptiness, we reduce from the following tiling problem. Given n in unary, a finite tile set, and a seed tile, decide whether a $2^n \times 2^n$ square can be tiled with the seed in its top-left corner, matching adjacent edge colors, and white outside boundaries. This problem is NEXP-complete (Schwarzentruber, 2019, Definition 1 and Theorem 3).

Theorem 13 (Nonemptiness and emptiness)**.** *Nonemptiness is NEXP-complete for fixed-precision softmax transformers over a fixed alphabet, and emptiness is coNEXP-complete. These bounds hold with an absolute bound on depth, one head per layer, no positional encodings, and the causal mask $j \leq i$. They hold for downward rounding, round-to-nearest, and the fixed-precision average-hard convention of Corollary 20.*

Proof outline. For a tile set of size q, encode each cell by two n-bit coordinates and a tile index. Require exactly 2^{2n} records and reject repeated coordinates. Together, these conditions give every cell exactly once. Four successive attention stages retrieve earlier neighbors, checking each edge at its later endpoint. Position-wise circuits check tile labels, outside boundaries, and the seed. The resulting transformer has constant depth, polynomial width, and precision $O(n + \log(q + 1))$ by Lemma 5.

For membership in NEXP, first evaluate the empty word. Otherwise, Corollary 10 supplies a certificate: an accepted word of length at most $2^{\mathrm{poly}(|T|)}$, which can be guessed and checked in exponential time. Proposition 25 bounds accumulator sizes, and Lemma 27 gives correctly rounded exponentiation in exponential time. Appendix A.7 proves the full reduction and evaluation bounds. ■

Equivalence and universality are likewise coNEXP-complete under the same architectural and rounding conventions (Appendix A.8).

7 Related work and conclusion

Succinctness and logical characterizations. Bergsträßer et al. (2026) prove doubly exponential succinctness over automata for unique-hard attention. Li & Cotterell (2025) characterize fixed-precision soft-attention transformers with the mask $j < i$ and no positional encodings by LTL[P]. Yang et al. (2026) characterize $(1, 1)$-precision transformers by C-RASP$_+$. Zetzsche et al. (2026) bound compressed length complexity in terms of C-RASP$_+$ program size.

Retrieval and computational resources. Index lookup, copying, and associative recall separate attention from recurrent or convolutional architectures (Bhattamishra et al., 2024; Wen et al., 2025; Jelassi et al., 2024; Arora et al., 2024). Our construction serializes records one bit per token and yields NFA state and randomized memory lower bounds. Related work studies state-space recall and tracking (Nishikawa & Suzuki, 2025; Huang et al., 2025; Merrill et al., 2024), depth and precision (Merrill & Sabharwal, 2023; Liu et al., 2023; Sanford et al., 2024), hard-attention languages (Yang et al., 2024), and softmax transformers with intermediate generation (Jiang et al., 2026).

Decision problems. Sälzer et al. (2025) prove NEXP-hardness of nonemptiness for fixed-precision transformers with average-hard attention and positional encodings. Sälzer et al. (2026) establish undecidability for unmasked transformers with finite but unbounded precision. Our fixed-precision, causally masked softmax model has NEXP-complete nonemptiness. Arithmetic robustness in Proposition 26 concerns the consistency recognizer. The universal upper bounds apply to the model in Section 2.2 and its equivalent bounded-accumulator implementation.

Conclusion. Constant-depth fixed-precision softmax transformers can check consistency of serialized records. Finite automata must distinguish doubly exponentially many prefix tables. Accepted words have singly exponential length, and equivalent C-RASP$_+$ programs require exponentially larger descriptions. The construction separates the size of the checking rule from the information distributed across the input.

REPRODUCIBILITY STATEMENT

The main text specifies the language families, transformer architecture, arithmetic conventions, and resource bounds. Appendix A gives complete proofs of the theoretical claims, including the circuit compilation, numerical range analysis, correspondence with the $(1, 1)$-precision model, and the hardness reduction. Appendix A.9 describes additional exact-arithmetic checks of the explicit constructions. An anonymized supplementary code package reproduces these finite checks.

AI USE STATEMENT

The authors formulated the research problem, directed the investigation, and selected among candidate approaches. OpenAI ChatGPT (GPT-5.6 Sol) was used under the authors' guidance to assist in developing the main construction, formulating mathematical claims, developing and drafting proofs, and deriving mathematical consequences. OpenAI ChatGPT and Anthropic Claude Fable 5.1 were also used in later stages for literature discovery and review, proof auditing, manuscript drafting and editing, figure design and editing, LaTeX and verification-code assistance, and terminology and consistency checks. References and claims identified through AI-assisted literature search were checked by the authors against the original sources, and issues identified during proof audits were reviewed and corrected. The authors reviewed the AI-assisted content and take full responsibility for the final manuscript and its correctness.

REFERENCES

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*, 2024. URL https://arxiv.org/abs/2312.04927.

Pascal Bergsträßer, Ryan Cotterell, and Anthony W. Lin. Transformers are inherently succinct. In *International Conference on Learning Representations*, pp. 58169–58189, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5f7804e8855efe5554025217abc49315-Abstract-Conference.html.

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*, volume 37, pp. 36002–36045. Curran Associates, Inc., 2024. doi: 10.52202/079017-1135. URL https://proceedings.nips.cc/paper_files/paper/2024/hash/3f630b20b7b3ac76d3a0016fe29b6dc0-Abstract-Conference.html.

Giuseppe De Giacomo and Moshe Y. Vardi. Linear temporal logic and linear dynamic logic on finite traces. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pp. 854–860, 2013. URL https://www.ijcai.org/Proceedings/13/Papers/132.pdf.

Ningyuan Teresa Huang, Miguel Sarabia, Abhinav Moudgil, Pau Rodriguez, Luca Zappella, and Federico Danieli. Understanding input selectivity in Mamba: Impact on approximation power, memorization, and associative recall capacity. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 25693–25727. PMLR, 2025. URL https://proceedings.mlr.press/v267/huang25ab.html.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat after me: Transformers are better than state-space models at copying. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 21502–21521. PMLR, 2024. URL https://proceedings.mlr.press/v235/jelassi24a.html.

Hongjian Jiang, Michael Hahn, Georg Zetzsche, and Anthony W. Lin. Softmax transformers are Turing-complete. In *International Conference on Learning Representations*, pp. 43932–43952,

2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/48c218b01802a712ee8d7243d27fd9cf-Abstract-Conference.html.

Ilan Kremer, Noam Nisan, and Dana Ron. On randomized one-round communication complexity. *Computational Complexity*, 8:21–49, 1999. doi: 10.1007/s000370050018.

Jiaoda Li and Ryan Cotterell. Characterizing the expressivity of fixed-precision transformer language models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 159510–159554. Curran Associates, Inc., 2025. doi: 10.52202/085713-5329. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/e9e250537b0345111d50a5f8f392cffc-Abstract-Conference.html.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=De4FYqjFueZ.

Charles London and Varun Kanade. Pause tokens strictly increase the expressivity of constant-depth transformers. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/817c0ed5303e396e803d6a972f83c475-Abstract-Conference.html.

William Merrill and Ashish Sabharwal. The parallelism tradeoff: Limitations of log-precision transformers. *Transactions of the Association for Computational Linguistics*, 11:531–545, 2023. doi: 10.1162/tacl_a_00562.

William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35492–35506. PMLR, 2024. URL https://proceedings.mlr.press/v235/merrill24a.html.

Naoki Nishikawa and Taiji Suzuki. State space models are provably comparable to transformers in dynamic token selection. In *International Conference on Learning Representations*, pp. 72645–72687, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/b4f6c4a82ff58ff125b13adf816956a7-Abstract-Conference.html.

Franz Nowak, Ryan Cotterell, and Reda Boumasmoud. A compositional theory of causally masked transformers. *arXiv preprint arXiv:2607.26988v1*, 2026. URL https://arxiv.org/abs/2607.26988v1.

Marco Sälzer, Eric Alsmann, and Martin Lange. Transformer encoder satisfiability: Complexity and impact on formal reasoning. In *International Conference on Learning Representations*, pp. 19720–19741, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/31f7bb23166893956c570d5bf26f4d18-Abstract-Conference.html.

Marco Sälzer, Chris Köcher, Alexander Kozachinskiy, Georg Zetzsche, and Anthony W. Lin. The counting power of transformers. In *International Conference on Learning Representations*, pp. 27080–27099, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2d663ddbad415a8105473a9e7f048da1-Abstract-Conference.html.

Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Transformers, parallel computation, and logarithmic depth. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43276–43327. PMLR, 2024. URL https://proceedings.mlr.press/v235/sanford24a.html.

François Schwarzentruber. The complexity of tiling problems. *arXiv preprint arXiv:1907.00102v2*, 2019. URL https://arxiv.org/abs/1907.00102v2.

Kaiyue Wen, Xingyu Dang, and Kaifeng Lyu. RNNs are not transformers (yet): The key bottleneck on in-context retrieval. In *International Conference on Learning Representations*, pp. 48813–48856, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/79dc391a2c1067e9ac2b764e31a60377-Abstract-Conference.html.

Andy Yang, David Chiang, and Dana Angluin. Masked hard-attention transformers recognize exactly the star-free languages. In *Advances in Neural Information Processing Systems*, volume 37, pp. 10202–10235, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/13d7f172259b11b230cc5da8768abc5f-Abstract-Conference.html.

Andy Yang, Pascal Bergsträßer, Georg Zetzsche, David Chiang, and Anthony W. Lin. Length generalization bounds for transformers. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2603.02238v2. Manuscript version: arXiv:2603.02238v2.

Georg Zetzsche, Hongjian Jiang, Andy Yang, Pascal Bergsträßer, Marco Sälzer, David Chiang, and Anthony W. Lin. Length generalization for transformers via compression. *arXiv preprint arXiv:2609.08851v1*, 2026. URL https://arxiv.org/abs/2609.08851v1.

A Proofs and constructions

A.1 Fixed-point arithmetic

We give explicit choices for Lemma 5 and verify the numerical bounds used in Theorem 4. For $b, R \geq 1$, set

$$k \coloneqq \max\{1, \lceil \log(R+1) \rceil, \lceil \log(b+2) \rceil\}, \quad s \coloneqq 6k+20, \quad p \coloneqq 2s+14, \quad H \coloneqq 2s, \tag{16}$$

$$\Lambda \coloneqq 2^{\lceil \log(16HR^2) \rceil}, \qquad D \coloneqq 2^{\lceil \log(b+1) \rceil}. \tag{17}$$

Write $Q \coloneqq 2^s = \delta^{-1}$ and $A \coloneqq 2^{p-s-1} = 8192Q$. The grid in Equation (1) has endpoints $-A$ and $A - \delta$.

Rounding and clipping. For real x, let $\mathrm{clip}(x) \coloneqq \min\{2^{p-1}-1, \max\{-2^{p-1}, x\}\}$. Downward rounding is

$$\mathrm{round}_{\downarrow}(x) \coloneqq \delta\, \mathrm{clip}(\lfloor Qx \rfloor). \tag{18}$$

For round-to-nearest, replace $\lfloor Qx \rfloor$ by a nearest integer, with a fixed deterministic tie rule. Clipping defines both maps on every real input. The bounds below show that saturation is never used in the construction under Section 2.2. Away from the endpoints, downward error has magnitude less than δ and nearest error at most $\delta/2$.

Lemma 14. *The parameters satisfy*

$$R \leq 2^k - 1, \quad b \leq 2^k - 2, \quad D \leq 2^k, \quad \Lambda < Q/8192, \quad Rb + 1 < Q/4, \quad R\delta \leq \frac{1}{4R^2}. \tag{19}$$

They are representable and computable with $O(k)$ bits.

Proof. The first three inequalities follow from the definition of k. Also $H = 12k + 40 \leq 2^{4k+2}$ for $k \geq 1$, whence

$$\Lambda < 32HR^2 < 2^{6k+7} = Q/8192.$$

Since $Rb + 1 < 2^{2k}$ and $Q = 2^{6k+20}$, the threshold bound follows. The same choice gives $4R^3 < Q$. The constants are integers or dyadic numbers with $O(k)$ integer and fractional bits. Computing the least enclosing powers of two uses their binary lengths. ■

Lemma 15 (Selection and the sum of attention coefficients)**.** *Fix an attention row i. If its scores are zero on a nonempty set $\mathcal{P}_i$ and at most $-H$ elsewhere, then its rounded unnormalized weights are exactly $\mathbb{1}\{\, j \in \mathcal{P}_i \,\}$. Its selected coefficients equal $\alpha \coloneqq \mathrm{round}(1/|\mathcal{P}_i|)$. They satisfy $|\mathcal{P}_i|\alpha \leq 1$ for downward rounding and $|\mathcal{P}_i|\alpha \leq 2$ for round-to-nearest.*

Proof. For $s \geq 1$, $e^{-2s} < 2^{-s-1} = \delta/2$. Exponentials of scores outside $\mathcal{P}_i$ therefore round to zero for either rule, while $\mathrm{round}(e^0) = 1$. This proves the coefficient formula. The downward inequality follows from $\alpha \leq 1/|\mathcal{P}_i|$. For round-to-nearest, if $\alpha > 0$, then $1/|\mathcal{P}_i| \geq \delta/2$, so $|\mathcal{P}_i|\delta \leq 2$ and $|\mathcal{P}_i|\alpha \leq 1 + |\mathcal{P}_i|\delta/2 \leq 2$. The zero case satisfies the bound directly. This argument includes nearest ties. ■

It follows that values in $[-V, V]$, with representable endpoints, yield a context in $[-2V, 2V]$ whenever those endpoints also fit. The bound is independent of the number of positions. Although the exact denominator may grow with input length, it can be capped without changing any rounded coefficient (Proposition 25).

Threshold circuits. The proof of Lemma 6 applies to every integer $m \geq 0$. Its context has the sharper range

$$-(t - \tfrac{1}{2}) \leq \zeta_t(m) \leq 2. \tag{20}$$

For the lower endpoint, use $\alpha \leq 1$ and $m \geq 0$. For the upper endpoint, use $m\alpha \leq 2$. Both endpoints are grid points. In the step circuit, the ReLU outputs are at most $2 + \delta$. Even if its final affine map forms the two weighted terms separately, their magnitudes are at most $2Q + 1$. Their difference is exactly 0 or 1.

The value map implementing a threshold is

$$V_t(j) \coloneqq P_j - (t - \tfrac{1}{2})Q_{\mathrm{BOS}}(j). \tag{21}$$

The half-integer coefficient multiplies a Boolean input, so this affine output is on the grid. The product $\alpha V_t(j)$ can be a multiple of $\delta/2$ without being a multiple of δ. Equation (3) sums these products exactly before rounding the context. Appendix A.11 also proves the threshold test correct when these products and their sum are rounded operation by operation.

Gating and Boolean operations. For $v \in \{0, 1\}$ and $0 \leq y \leq K$, with representable K, define

$$\gamma_K(v, y) \coloneqq \mathrm{ReLU}(y) - \mathrm{ReLU}(y - Kv). \tag{22}$$

The gate $\gamma_K(v, y)$ passes y when $v = 1$ and outputs 0 when $v = 0$. Boolean conjunction and disjunction are implemented by

$$\bigwedge_{u=1}^{m} x_u = \mathrm{ReLU}\left(\sum_{u=1}^{m} x_u - m + 1\right), \qquad \bigvee_{u=1}^{m} x_u = \mathrm{ReLU}\left(\sum_{u=1}^{m} x_u\right) - \mathrm{ReLU}\left(\sum_{u=1}^{m} x_u - 1\right). \tag{23}$$

These formulas apply to Boolean operands. Negation is $1 - x$. All products in these affine maps are integer times a grid value or a dyadic constant times a Boolean value.

Capping. For arbitrary $m \geq 0$, Lemma 15 gives $0 \leq \rho_m = m\,\mathrm{round}(1/(m + 1)) < 2$. The context computing ρ_m is already a grid point. The cap in Equation (8) is the explicit circuit

$$\sigma_i = \gamma_{2\Lambda}(1 - g_i, \Lambda\rho_{\kappa_i}) + (\Lambda\rho_R)g_i. \tag{24}$$

The parameter $\Lambda\rho_R$ is obtained by integer division followed by multiplication by a power of two. Each term in Equation (24) is on the grid. Both gate preactivations lie in $[-2\Lambda, 2\Lambda)$, so their magnitudes are at most 2Λ. For $0 \leq t < m \leq R$, downward rounding gives

$$\rho_m - \rho_t \geq \frac{m - t}{(m + 1)(t + 1)} - m\delta,$$

and round-to-nearest gives the same inequality with error $(m + t)\delta/2$. Both imply

$$\rho_m - \rho_t \geq \frac{1}{R(R + 1)} - R\delta \geq \frac{1}{4R^2}. \tag{25}$$

Since $\Lambda \geq 16HR^2$, we obtain $\Lambda(\rho_m - \rho_t) \geq 4H$. Therefore the capped addresses are separated up to R and constant from R onward, for every word.

Exact arithmetic and bounds. Dyadic parameters suffice to encode the circuits. To show that rounding preserves these computations, we check that the operands make every affine result lie on the grid. The address scale and step coefficients are integers. The half-integer threshold values multiply $Q_{\mathrm{BOS}}(j) \in \{0, 1\}$. The segment score multiplies an address only by a Boolean or an integer. The conflict score multiplies signs in $\{-1, 0, 1\}$ by integers. In the decoder, D and the clipping coefficients are integers. Gates use Boolean controls. Hence these affine products, sums, and ReLU outputs are on the grid. Bounds for the principal quantities are collected below. In the conflict row, $n = b - 1$.

Quantity	Bound in real units
Counting context $\zeta_t(m)$	$[-(t-1/2), 2]$
Weighted step terms	magnitude at most $2Q+1$
Raw scaled address	$[0, 2\Lambda)$
Address and segment score computations	magnitude below $4\Lambda + H < Q$
Decoder context	$[0, 2D]$
Decoder affine preactivations	magnitude at most $4D+1$
Decoded bits at valid positions (zero elsewhere)	$\{0, 1\}$
Conflict scores	$[-(2n+1)H, 0]$
Boolean sums in the main recognizer	at most $b + O(1)$

A conservative bound of $8Q$ covers the finite products, partial sums, and stored values of these circuits, well within $[-8192Q, 8192Q - \delta]$. The operations used below to preserve signed registers only change signs or add a fixed number of copies. The two attention accumulations retain the exact semantics in Equations (2) and (3). Their number of summands is unrestricted.

Proof of Lemma 5. The choices in Equations (16) and (17) have $p, s = O(\log(R{+}1){+}\log(b{+}2))$ and are computable in polynomial time. Lemma 14 gives representability and the threshold bound. The choice $H = 2s$ gives $e^{-H} < \delta/2$, as in Lemma 15. By definition, $D \geq b{+}1$, and $D\delta \leq 2^k/Q < 1/2$. Equation (25) and $\Lambda \geq 16HR^2$ give the address separation. The preceding grid and range checks establish exactness of the finite affine and ReLU computations. ■

A.2 Implementing registers with transformer layers

The construction is given in terms of fixed-point registers at each position. We explain how these registers are implemented by the architecture of Section 2.2.

Lemma 16. *Put $A \coloneqq 2^{p-s-1}$, so the grid has endpoints $-A$ and $A - \delta$. Suppose a program has r registers per position and a constant number of stages. Each stage consists of one attention operation with affine query, key, and value maps of dimension $O(r)$, followed by an affine/ReLU circuit of constant depth and width $O(r)$. Suppose all parameters are representable, every finite affine/ReLU operation is exact on the grid, and all parameters, stored registers, and inputs, intermediate values, and outputs of these operations lie in $[-K, K]$ for some $K \geq 1$ with $2K \leq A - \delta$. Then the program has an implementation using one head per layer, constant depth, width $O(r)$, and a dense binary description of $O(pr^2)$ bits.*

Proof. Partition the hidden coordinates into permanent registers, temporary registers, and a block reserved for the attention output. Every coordinate in this last block is zero before each attention stage. Set the value map to write only into that block. The residual sum then contains both the old registers and the new context, in disjoint coordinates. Affine query and key maps ignore unused coordinates.

Implement each original affine or ReLU layer separately, preserving a signed register x through a ReLU by passing $\mathrm{ReLU}(x)$ and $\mathrm{ReLU}(-x)$ and reconstructing their difference. In an affine row, reconstruction replaces each coefficient a by a and $-a$ on these two coordinates. The bound on K makes the inserted negations and reconstructions exact and representable. Nonnegative registers may be copied directly. Keeping the original affine operations separate increases width by a constant factor. Each circuit layer uses a constant number of affine–ReLU–affine maps. Insert transformer layers with zero value map between attention stages. Their context is zero on every input. Reset the reserved attention coordinates after each stage. The total number of transformer layers is constant.

All matrix coefficients and biases are original parameters, their negatives, or elements of $\{0, 1, -1\}$. They are representable because $K \geq 1$ and $2K \leq A - \delta$. The original exactness assumptions and the bound on inserted values ensure that rounding preserves the computation. There is a constant number of matrices of dimensions $O(r) \times O(r)$, each with p-bit entries. The encoding length is $O(pr^2)$, and assembling the matrices takes time polynomial in their total size. ■

The bounds in Appendices A.1 and A.7 permit $K = 8Q$ for the consistency recognizer and the tiling checker. Since $A = 8192Q$, we have $2K = 16Q < A - \delta$. Thus both constructions satisfy the range condition with the stated precision and parameters.

Queries and keys. For global selection of a Boolean predicate P excluding BOS, use query 1 and key $-H(1 - Q_{\mathrm{BOS}}(j) - P_j)$. The segment score Equation (9) factors as

$$\mathbf{q}_i \coloneqq (1, -\sigma_i, \sigma_i, -H), \qquad \mathbf{k}_j \coloneqq (\sigma_j, 1, Q_{\mathrm{BOS}}(j), 1 - Q_{\mathrm{BOS}}(j) - P_j). \tag{26}$$

The conflict score Equation (12) factors as

$$\mathbf{q}_i \coloneqq (Hz_{i,1}, \ldots, Hz_{i,n}, -H\eta_i/2, 1), \quad \mathbf{k}_j \coloneqq (z_{j,1}, \ldots, z_{j,n}, \eta_j, (Hn + H/2)(Q_{\mathrm{BOS}}(j) - 1)). \tag{27}$$

Zero coordinates embed these vectors into the hidden dimension. Each factor is an affine map of available registers. Since $H = 2s$ is even, the displayed coefficients are integers.

A.3 Recognition construction

We specify the values computed at each stage and the properties that hold for every input in Theorem 4. We write $Q_a(j)$ for the indicator that position j carries symbol a, including $Q_{\mathrm{BOS}}(j) = \mathbb{1}\{\, j = 0 \,\}$. We omit the position argument when it is clear. Ordinary token indicators are $Q_0, Q_1, Q_{@}, Q_{\#}$, and $Q_{\mathrm{bit}} \coloneqq Q_0 + Q_1$. The embedding includes a register whose value is always one. All these registers persist.

Stage 1: addresses and the cap. Select $Q_{\mathrm{BOS}} \vee Q_{\#}$. Two value coordinates are $Q_{\#}$ and $Q_{\#} - (R - 1/2)Q_{\mathrm{BOS}}$. The first context is ρ_{κ_i}. Applying Lemma 6 to the second gives $g_i = \mathbb{1}\{\, \kappa_i \geq R \,\}$. Apply Equation (24) to obtain σ_i. This address is nondecreasing and obeys $\sigma_0 = 0$.

Stage 2: bit ranks. Use Equation (26) with $P = Q_{\mathrm{bit}}$. For $t = 1, \ldots, b + 1$, use value coordinate $Q_{\mathrm{bit}} - (t - 1/2)Q_{\mathrm{BOS}}$ and apply Step_δ to its context. Denote the resulting Boolean outputs by G_t. The quantities $G_{u+1} - G_{u+2}$ are Boolean, even after the R-th hash, because they are differences of the outputs for consecutive thresholds. At each position j, compute

$$W_j(u) = \mathrm{ReLU}(Q_1(j) + G_{u+1}(j) - G_{u+2}(j) - 1), \qquad \chi_{\mathrm{bit}}(j) \coloneqq G_b(j) - G_{b+1}(j).$$

The flag $\chi_{\mathrm{bit}}(j)$ tests whether exactly b bit positions are selected. Before the cap, $W_j(u) = 1$ exactly when position j contains a bit with value 1 at rank u in its segment.

Stage 3: validity. Use the segment selector with $P = Q_{@}$ and thresholds $1, 2$. If their outputs are A_1, A_2, put $\chi_{@}(i) \coloneqq A_1(i) - A_2(i)$. This flag tests whether exactly one @ position is selected. Define

$$\nu_i \coloneqq \mathrm{ReLU}(Q_{@}(i) + \chi_{\mathrm{bit}}(i) + \chi_{@}(i) + (1 - g_i) - 3).$$

This is the conjunction of four Boolean conditions. The predicate indicating an invalid boundary is $I_i \coloneqq Q_{@}(i) - \nu_i$, also Boolean. Both ν_0 and I_0 are zero.

Stage 4: parallel decoding. At query position i, select BOS and the bits in its segment. For each selected ordinary position j, use value coordinate $v_{j,u} \coloneqq DW_j(u)$, and assign BOS zero. Let $y_{i,u}$ be the returned context coordinate. Then

$$\beta_{i,u} \coloneqq \gamma_1(\nu_i, \mathrm{clip}_{[0,1]}(2y_{i,u})).$$

Every $y_{i,u}$ is nonnegative and at most $2D$. At a valid boundary i, it is either zero or greater than $1/2$, by Equation (11). Consequently $\beta_{i,u}$ is the exact record bit there. At every invalid position it is zero. Thus the resulting register is Boolean on every word, even though the ungated clipped value may be fractional at an invalid position.

Stage 5: conflicts. Set $b = n + 1$, construct the signs from the decoded bits and ν_i, and use Equation (27). The attention value $\nu_j - Q_{\mathrm{BOS}}(j)/2$ equals 1 at each selected ordinary position and $-1/2$ at BOS. For a valid query, the selected ordinary positions are precisely the conflicting valid boundaries. The counting circuit with threshold one, conjoined with ν_i, gives a Boolean flag C_i. For invalid queries, every ordinary candidate has score $-Hn - H/2$, so only BOS is selected and $C_i = 0$.

Stages 6–10: global checks. Global selectors use the factorization above Equation (26). Select, in succession, hashes, boundaries, bits, invalid boundaries, and conflicts. For the first three predicates, use paired thresholds $(R, R+1)$, $(R, R+1)$, and $(Rb, Rb+1)$ to obtain the equality flags. For each remaining predicate use threshold one and negate its result. Retain all five flags and output their conjunction. Each equality pair shares one attention head. These registers are evaluated at every position and read at the final position.

Lemma 17. *The three equality flags and the flag requiring all boundaries to be valid accept exactly* $(\{0,1\}^b\texttt{@}\texttt{\#})^R$.

Proof. Suppose these flags hold. The R hashes determine R segments preceding a hash and one final tail. Every @ is valid and hence lies before the cap. No such segment can contain two @ symbols: the later one would see at least two and have $\chi_{\texttt{@}} = 0$. Since there are R valid boundaries and R segments preceding a hash, each segment contains one boundary. Each boundary has exactly b preceding bits in its segment. Their disjoint sets contain all Rb bits in the word. Hence no bit occurs after a boundary or in the tail. The alphabet has only bits and the two delimiters, and every delimiter is already accounted for. The word is therefore in $(\{0,1\}^b\texttt{@}\texttt{\#})^R$.

Conversely, every word in this format has the stated global counts, exactly b bits and one boundary in each segment, and no boundary at or after the cap. Every boundary is valid. The empty word fails the hash equality. All count arguments apply to arbitrary input lengths by Lemma 6. ■

Lemma 18. *On a formatted word,* $C_i = 1$ *exactly at boundaries with an earlier record of the same key and opposite value. On arbitrary words, every* C_i *is Boolean and every attention row used to detect conflicts has scores zero or at most* $-H$.

Proof. All valid boundary signs are in $\{-1, 1\}$ and all invalid signs are zero. For two valid boundaries at key Hamming distance h, the inner product of the sign-encoded keys is $n - 2h$. Substitution in Equation (12) gives score $-2Hh$ for opposite values and $-H - 2Hh$ for equal values. A valid query and invalid ordinary candidate give $-Hn - H/2$. The same score occurs at every ordinary candidate of an invalid query. At BOS the score is zero. Since $n \geq 1$, each negative case is at most $-H$. The test for at least one selected non-BOS candidate is therefore exact by Lemma 6, and gating it by validity preserves its Boolean output. The score of a record against itself is $-H$, so a selected conflicting record is strictly earlier. ■

Lemmas 17 and 18 prove both directions of recognition. In particular, a malformed prefix cannot create undefined later computations: the capped addresses, the gates that zero invalid decoded bits, and the Boolean conflict flags ensure that every attention row satisfies the stated score bounds independently of the final format decision.

Resource accounting. The construction reserves $b + 1$ registers for the outputs of threshold tests, b for the values $W(u)$, and b for the decoded bits. Apart from the n signs, all other permanent quantities use a constant number of registers. All position-wise stages have a fixed number of affine/ReLU operations. By Lemma 16, the layer count is independent of n, the width is $O(n)$, and the dense description is $O(n^3)$. For $b = n + 1$ and $R = 2^{n+1}$, the choices give $k = n + 2$, $s = 6n + 32$, and $p = 12n + 78$. The implemented network has 29 transformer blocks, hidden dimension $10n + 38$, and maximum hidden dimension of a feed-forward network $22n + 80$. These explicit constants provide one realization of the asymptotic bounds.

A.4 Succinctness bounds

Nondeterministic state lower bound. The proof based on accepting runs in Proposition 3 also covers automata with ε transitions: choose a state on each accepting run immediately after the first u_f, assigning any intervening ε moves consistently to either side. A shared state permits the same splice. There are 2^M distinct tables, so the state bound is 2^M.

Deterministic upper bound for L_n. The ternary table has 3^M possibilities. During a record, the automaton retains its value, at most n key bits, and a position in $\{0, \ldots, n + 2\}$. Its record counter ranges from 0 to $2M$, with a single overflow state. These additional components have $2^{O(n)}$

possibilities. Once the current key is complete, the automaton either assigns an unseen table entry or checks its stored value. It accepts only at a complete record boundary with exactly $2M$ records and no error. This proves the upper bound $3^M 2^{O(n)}$.

LTL lower bound and existence. We use LTL over finite words, built from symbol tests and Boolean connectives with next (X) and until (U). The constant $\top$ is always true. At position i of a word $w = a_1 \cdots a_N$, Q_a tests whether $a_i = a$. The formula $\mathsf{X}\phi$ holds when $i < N$ and ϕ holds at $i + 1$. The formula $\phi \,\mathsf{U}\, \psi$ holds when ψ holds at some $j \in \{i, \ldots, N\}$ and ϕ holds at every position from i to $j - 1$. A nonempty word satisfies a formula when it holds at position 1. We measure formula size by the number of nodes in its syntax tree. A formula of size m has an NFA with $2^{O(m)}$ states (De Giacomo & Vardi, 2013) and (Bergsträßer et al., 2026).

Corollary 19 (LTL succinctness)**.** *The language L_n is definable in LTL over finite words, and every such formula has size $\Omega(2^n)$. Hence constant-depth fixed-precision softmax transformers are exponentially more succinct than LTL over finite words.*

Proof. Each word can be specified using symbol tests and next, with $\neg\mathsf{X}\top$ marking its last position. Since L_n is finite, a disjunction of these formulas defines it. If a formula φ defines L_n, the exponential translation to an NFA and Proposition 3 give $2^{2^n} \leq 2^{O(|\varphi|)}$. Therefore $|\varphi| = \Omega(2^n)$. Theorem 4 gives transformer descriptions of size $O(n^3)$. ■

Average-hard attention.

Corollary 20. *Replace each softmax head in the constructed networks by a head selecting the set $\mathcal{P}_i$ of all score maximizers in row i, assign each selected position coefficient* $\mathrm{round}(1/|\mathcal{P}_i|)$*, and round the context as in Equation* (3)*. Every hidden state is unchanged. Consequently, the same succinctness lower bounds hold for this fixed-precision average-hard model.*

Proof. Every row has maximum score zero and all other scores at most $-H$. Softmax and the stated average-hard rule therefore produce identical coefficients and contexts. ■

DFA simulation. We give a direct finite-state simulation for the numerical convention of Section 2.2.

Proof of Proposition 9. Write $\delta := 2^{-s}$ and $U_{\max} := 2^{p-s-1} - \delta$. Every positive rounded unnormalized weight is at least δ and at most $U_{\max}$, including when clipping occurs. Fix a layer and a current query. The preceding layer has at most 2^{pd} possible hidden vectors. Since the maps have no positional dependence, each such vector $\mathbf{h}$ determines a key, a value $\mathbf{v}(\mathbf{h})$, and a rounded unnormalized weight $u(\mathbf{h})$ for this query. Let $N_{\mathbf{h}}$ count its visible occurrences.

If some $u(\mathbf{h}) > 0$ has $N_{\mathbf{h}} \geq K$, then $Z \geq K\delta = 2^p$. Every normalized weight satisfies

$$0 \leq \frac{u(\mathbf{h}')}{Z} \leq \frac{2^{p-s-1} - \delta}{2^p} = \frac{\delta}{2} - \frac{\delta}{2^p} < \frac{\delta}{2}.$$

Every attention coefficient therefore rounds to zero under either rule, and the context is zero. Otherwise, every hidden vector with positive weight has multiplicity below K. Counts capped at K retain these multiplicities exactly. Vectors whose weights are zero contribute nothing to either sum. Thus the capped counts determine the denominator, the rounded coefficients, and the context computed by Equation (3). They also determine the specified zero context when $Z = 0$.

A deterministic monitor stores, for each layer ℓ, a histogram of the previously processed vectors $\mathbf{h}^{\ell-1}$, with every count capped at K. On reading a symbol, it computes its embedding. At layer ℓ, it first inserts the current $\mathbf{h}^{\ell-1}$ into that histogram, so the causal mask includes the current position. The histogram and current vector determine the context by the preceding argument. The residual and position-wise map then determine $\mathbf{h}^{\ell}$. Repeating this procedure computes the current output. Causality ensures that earlier representations remain unchanged, so each updated histogram continues to count the appropriate vectors from the preceding layer.

Initialize the monitor by processing the externally supplied BOS once from empty histograms using the same procedure. Store an additional bit recording the current acceptance value. Its initial value handles the empty word. Subsequent transitions depend only on the histograms and the next input

symbol. There are $L2^{pd}$ counters, each with $K+1$ possible values, and two values for the acceptance bit. This gives the first bound. The second uses $\log_2(K+1) \le p+s+1$. The declared dense encoding bounds p, d, L by polynomials in S. Finally, every DFA is also an NFA. ■

The same simulation and bounds hold for the average-hard convention of Corollary 20. A capped histogram preserves the set of present hidden vectors and hence the maximizing types. If any maximizing type has count at least K, then the common reciprocal is at most $1/K < \delta/2$, so every coefficient rounds to zero. Otherwise all maximizing counts are exact. The monitor therefore determines the context in either case.

Proof of Corollary 10. Let $\mathcal{H}$ denote the complete collection of histograms, and let $\Phi(\mathcal{H})$ be the sum of its counters. Each counter is nondecreasing, and $\Phi(\mathcal{H}) \le L2^{pd}K = W_T$. Consider a shortest accepted word of positive length. If a nonfinal symbol leaves Φ unchanged, it leaves every counter unchanged. Deleting that symbol preserves $\mathcal{H}$ and hence every subsequent transition and the final output: the stored acceptance bit is used only for readout. Minimality therefore forces every nonfinal symbol to increase the integer Φ by at least one. The word has at most $W_T + 1$ symbols.

For distinction, run both monitors on the same word and sum all counters in both monitors, obtaining a bound of $W_T + W_{T'}$. The same deletion argument applies to a shortest word on which their acceptance bits differ, giving the second bound. The empty word also satisfies this bound. ■

A.5 C-RASP$_+$ AND THE $(1,1)$-PRECISION MODEL

Equation (15) is a linear-size normal form of the syntax of Yang et al. (2026, Definition 4.1). Their numeric terms are prefix counts or nonnegative integer constants, and comparisons place a nonnegative linear combination of these terms against a nonnegative threshold. Collecting constant terms produces C_a without changing the Boolean dependency DAG. Its binary length is at most linear in the total written coefficient and constant lengths, so Lemma 11 applies to the original syntax. Signed constant offsets can also be allowed: move the offset to the threshold, evaluate a comparison with a negative threshold directly, and otherwise cap at the shifted threshold plus one. The total binary length of these thresholds is bounded by the original program size.

Proof of Lemma 11. For each comparison a, store its left-hand side capped at $k_a + 1$. The constant C_a is included in the initial value $q_a(0) \coloneqq \min\{C_a, k_a + 1\}$. Since $\psi_{ar}(i) \in \{0, 1\}$,

$$\overleftarrow{\#}[\psi_{ar}](i) = \overleftarrow{\#}[\psi_{ar}](i-1) + \psi_{ar}(i).$$

Thus the new position contributes λ_{ar} whenever $\psi_{ar}(i) = 1$. On reading position i, evaluate the definitions in topological order and update

$$q_a(i) \coloneqq \min\left\{k_a + 1,\ q_a(i-1) + \sum_r \lambda_{ar}\psi_{ar}(i)\right\}. \tag{28}$$

All increments are nonnegative. Induction on positions and definitions shows that $q_a(i)$ is exactly the capped left-hand side of Equation (15). Evaluate comparison a as $q_a(i) \bowtie_a k_a$. Values above k_a are represented by $k_a + 1$, preserving the result. The counter values and the current symbol determine all current Boolean definitions, including negations. Thus the counters and one acceptance bit form a DFA with at most $2\prod_a(k_a+2)$ states. Binary encoding gives $\sum_a \log(k_a+2) = O(|\varphi|)$, proving the bound. Shared definitions are evaluated once per position. The acceptance bit is initialized to the specified empty-word value and updated to the final definition after each symbol. ■

Lemma 21 (Correspondence with $(1,1)$-precision attention)**.** *The transformer with downward rounding T_n of Theorem 4, on inputs beginning with BOS, is a $(1,1)$-precision transformer in the sense of Yang et al. (2026, Definitions B.1–B.3). Its depth, width, precision, and description bounds are unchanged. Any C-RASP$_+$ program that agrees with this transformer on inputs beginning with BOS has size $\Omega(2^n)$.*

Proof. Both models use $\mathbb{F}_{p,s}$, downward rounding, the mask $j \le i$, a residual connection, and rounding of exponentials, normalized coefficients, and contexts. The compiled query, key, value,

and feed-forward maps are permitted choices of their defining functions. Appendices A.1 and A.2 place every finite affine, dot-product, residual, and ReLU computation on the grid and within range. Additional rounding is therefore exact and clipping inactive. BOS has score zero, so every denominator is positive. The attention definitions agree on every input beginning with BOS, and both models accept when the final output is positive.

For a C-RASP$_+$ program φ agreeing on BOSw, $w \in \Sigma^*$, Lemma 11 gives a DFA with $2^{O(|\varphi|)}$ states. Process BOS once and make the resulting state initial. The DFA then recognizes L_n with the same state count, so Proposition 3 gives $|\varphi| = \Omega(2^n)$. This covers both supplied and explicit BOS conventions. ■

A matching upper bound for L_n. Let $N := R(n+3)$ be the required word length. At position i, let $Q_a(i)$ test for symbol $a \in \Sigma$, and define $\top(i) := Q_0(i) \vee Q_1(i) \vee Q_@(i) \vee Q_\#(i)$. For $t \in [N]$ and $a \in \Sigma$, put

$$P_{t,a}(i) := (\overleftarrow{\#}[\top](i) = t) \wedge Q_a(i), \qquad E_{t,a}(i) := (\overleftarrow{\#}[P_{t,a}](i) \geq 1).$$

Here $P_{t,a}(i)$ holds exactly when $i = t$ and $a_i = a$, while $E_{t,a}(i)$ records whether position t has been read and carries a. All final tests below are evaluated at $i = |w|$. Index records by $j = 0, \ldots, R-1$ and set $p(j,u) := j(n+3) + u + 1$ for $u = 0, \ldots, n$, where $u = 0$ is the value bit and $u = 1, \ldots, n$ are the key bits. The format formula requires $\overleftarrow{\#}[\top](i) = N$, so $i = N$ whenever it passes. It also requires $E_{p(j,u),0}(i) \vee E_{p(j,u),1}(i)$ for every bit slot, and $E_{j(n+3)+n+2,@}(i)$ and $E_{j(n+3)+n+3,\#}(i)$ for every record j.

For $j < k$, define

$$\mathrm{KeyEq}_{j,k}(i) := \bigwedge_{u=1}^{n} \big((E_{p(j,u),0}(i) \wedge E_{p(k,u),0}(i)) \vee (E_{p(j,u),1}(i) \wedge E_{p(k,u),1}(i))\big)$$

and

$$\mathrm{ValDiff}_{j,k}(i) := (E_{p(j,0),0}(i) \wedge E_{p(k,0),1}(i)) \vee (E_{p(j,0),1}(i) \wedge E_{p(k,0),0}(i)).$$

The conjunction of the format formula with $\neg(\mathrm{KeyEq}_{j,k}(i) \wedge \mathrm{ValDiff}_{j,k}(i))$ for all $j < k$ recognizes exactly L_n. There are $O(N)$ position and symbol predicates and $O(R^2 n)$ references in the pairwise consistency checks. All written position constants have $O(n)$ bits. Under the DAG size convention this gives total size $O(Nn + R^2 n) = 2^{O(n)}$, proving the upper bound in Theorem 12.

A.6 Randomized one-pass recognition

A one-pass recognizer reads left to right and uses B bits if it has at most 2^B persistent configurations at each position, including all information needed to resume. Fresh independent randomness and shared input-independent randomness are allowed. A fixed-precision RNN with d hidden coordinates of p bits and no other persistent state has $B = pd$. Let $h_2(u) := -u \log u - (1-u)\log(1-u)$, with $0 \log 0 = 0$.

For a finite-valued random variable Z, write $H(Z) := -\sum_z \Pr[Z = z] \log \Pr[Z = z]$ for its Shannon entropy in bits. Conditional entropy $H(Z \mid Y)$ is the remaining entropy of Z, averaged over the observed value of Y. We use that conditioning cannot increase entropy, joint entropy is at most the sum of coordinate entropies, and the conditional chain rule $H(Z, Z' \mid Y) = H(Z \mid Y) + H(Z' \mid Z, Y)$.

Theorem 22 (Memory lower bound)**.** *Any randomized one-pass recognizer for L_n with error at most $\epsilon < 1/2$ requires*

$$B \geq (1 - h_2(\epsilon))2^n \tag{29}$$

bits of memory. The bound holds for worst-case error and for average error under the distribution below.

Proof. We use the standard one-way INDEX reduction (Kremer et al., 1999), specialized to L_n. Choose a uniform table $F \in \{0,1\}^M$ and an independent uniform key $X \in \{0,1\}^n$. The hard input is

$$w_{F,X} := u_F \underbrace{r(1,X) \cdots r(1,X)}_{M \text{ copies}}. \tag{30}$$

The suffix repeats the record with value 1 and key X. The word $w_{F,X}$ has exactly $2M$ records and belongs to L_n if and only if $F(X) = 1$, where $F(X)$ is the table entry at key X.

All prefixes u_F have the same length. Alice streams this prefix and sends the recognizer's entire persistent configuration S to Bob. The fresh random coins may be indexed by input position and included in a shared tape C, independent of F and X. Given (S, C, X), Bob resumes the recognizer on the prescribed suffix. Any stored random seed, data-dependent control position, or other information required for resumption is included in S. The reduction therefore sends at most B bits.

For clarity, the binary prediction bound used below follows directly from entropy. If $Z \in \{0, 1\}$ is predicted from Y with error probability e, let E indicate a prediction error. Given Y, the variables Z and E determine each other, so $H(Z \mid Y) = H(E \mid Y) \le H(E) = h_2(e)$.

For each key x, let e_x^* be the optimal prediction error for $F(x)$ given (S, C). It is at most $1/2$ and no greater than the error of the simulated recognizer. A distributional error bound ϵ implies $M^{-1} \sum_x e_x^* \le \epsilon$. Conditional subadditivity, Fano's inequality for a binary variable, and concavity give

$$H(F \mid S, C) \le \sum_x H(F(x) \mid S, C) \le \sum_x h_2(e_x^*) \le M h_2(\epsilon).$$

The uniform table contains M bits of uncertainty, so independence gives $H(F \mid C) = M$. Given C, observing S leaves at most $M h_2(\epsilon)$ bits. The state must account for this reduction, but holds at most B bits. Formally, $H(S \mid C) \le B$ and the chain rule give

$$B \ge H(S \mid C) \ge H(F \mid C) - H(F \mid S, C) \ge M(1 - h_2(\epsilon)),$$

proving Equation (29). Setting $\epsilon = 0$ recovers the deterministic bound $B \ge 2^n$. For $\epsilon = 1/3$, the multiplicative constant is approximately 0.0817042. ■

A fixed-precision recurrent state with d coordinates of p bits each must satisfy $pd \ge (1 - h_2(\epsilon))2^n$. An exact one-pass recognizer stores a ternary entry for each key and $O(n)$ bits of parser and counter state, using $O(2^n)$ bits. This matches the lower bound for any fixed $\epsilon < 1/2$. In terms of accepted length $N = \Theta(n2^n)$, the lower bound is $\Omega(N/\log N)$. The transformer's description counts its encoded rule; its earlier token representations remain available through attention.

A.7 Tiling exponentially large grids

We reduce from the following tiling problem. An instance consists of n in unary, a finite set $\mathcal{D}$ of tile types, and a seed $t_0 \in \mathcal{D}$. A tile specifies its left, top, right, and bottom colors. The question is whether a $2^n \times 2^n$ square admits a tiling with matching adjacent colors, white outside edges, and tile t_0 in the top-left corner. This is precisely Definition 1 and the $k = 1$ case of Theorem 3 of Schwarzentruber (2019), and is NEXP-complete. With $q := |\mathcal{D}|$, we may assume $n \ge 1$ and $q \ge 1$. Trivial invalid instances map to a transformer that rejects every word.

Index the tiles by $0, \dots, q-1$, identifying the seed with its index. Let $G := 2^n$ be the grid side length and set $\ell := \max\{1, \lceil \log_2 q \rceil\}$. Serialize a cell as $xyt@\#$, where $x, y \in \{0, 1\}^n$ are its coordinates and $t \in \{0, 1\}^\ell$ is its tile index. Apply the record parser with $b = 2n + \ell$ and $R = G^2$. It supplies a Boolean validity register ν_i, decoded bits at valid boundaries, and zero decoded registers elsewhere. Coordinate bits are ordered from most significant to least significant, with x increasing rightwards and y downwards. All input symbols belong to the fixed alphabet $\{0, 1, @, \#\}$.

Coordinate lookup and duplicates. Let $\mathbf{z}_i \in \{-1, 0, 1\}^{2n}$ be the coordinate signs at a valid boundary, and zero elsewhere. For any binary target coordinate, let $\mathbf{a}_i \in \{-1, 1\}^{2n}$ be its sign vector. With $Q_{\mathrm{BOS}}(j) = \mathbb{1}\{\, j = 0 \,\}$, form each target sign from its decoded bit β as $2\beta - 1$, including at invalid queries, and use the score

$$S_a(i, j) := H\langle \mathbf{a}_i, \mathbf{z}_j \rangle - 2nH + 2nHQ_{\mathrm{BOS}}(j).$$

BOS has score zero. A valid candidate has score zero exactly when its coordinate equals the target. Otherwise its score is at most $-2H$. An invalid non-BOS candidate has score $-2nH$. These statements hold for every query, including invalid queries, and all scores lie in $[-4nH, 0]$. Thus rounded exponentials are exactly zero or one, with BOS always selected.

A lookup at the current record's coordinate, followed by the counting circuit with threshold two, detects duplicates: at a valid boundary, the causal mask includes the current occurrence, so two selected candidates mean an earlier duplicate. Gate this flag by ν_i. Exact record count and absence of duplicates then force all G^2 coordinates to occur exactly once.

Neighbors and local labels. Coordinate increments and decrements modulo 2^n use Boolean carry and borrow circuits. Write $x_0, \ldots, x_{n-1}$ for the bits ordered from most significant to least significant, and set $\chi_k \coloneqq \bigwedge_{j>k} x_j$, with the empty conjunction equal to one, and $x'_k \coloneqq x_k + \chi_k - 2(x_k \wedge \chi_k)$. Each conjunction is one ReLU of an affine sum. All bits are obtained in constant depth using $O(n)$ units and $O(n^2)$ coefficients. Decrement substitutes $1 - x_j$ in the borrow conjunction. Endpoint tests disable wrapped queries. Four successive attention stages retrieve the coordinates $(x+1, y)$, $(x-1, y)$, $(x, y+1)$, and $(x, y-1)$.

Use r, t as tile indices in the local label checks. For $r \in \{0, \ldots, q-1\}$, let $e_r(i)$ be the conjunction of ν_i with the literals specifying tile index r. Every e_r is Boolean, and $\sum_r e_r(i) \le \nu_i$ on every input. The flag for an illegal tile label is therefore the Boolean value $\nu_i - \sum_r e_r(i)$. In a neighbor lookup, assign BOS the zero vector and each candidate the value $2(e_0, \ldots, e_{q-1})$.

Lemma 23 (Checking neighboring tiles)**.** *For each direction there is a constant-depth ReLU circuit with $O(q)$ intermediate units and $O(q^2)$ coefficients that produces a Boolean violation flag on every input. When the valid records have legal tile labels and distinct coordinates, it is one exactly when an enabled lookup returns an incompatible earlier neighbor.*

Proof. Suppose the row selects BOS and m candidate boundaries, of which m_t have legal tile index t. Put $\alpha \coloneqq \mathrm{round}(1/(m+1))$. The returned coordinates are

$$c_t = 2\alpha m_t \in \delta\mathbb{Z}_{\ge 0}.$$

The expression already lies on the grid, so final context rounding is exact. Since $\sum_t m_t \le m$ and the rounded attention coefficients sum to at most two, $\sum_t c_t \le 4$ on every input. Under downward rounding the stronger bound two holds.

Let $A_{rt} \in \{0, 1\}$ indicate an incompatible ordered tile pair for this direction. For the rightward query, for example, $A_{rt} = 1$ exactly when the right color of tile r differs from the left color of tile t. Define

$$u_r \coloneqq \sum_t A_{rt} c_t, \qquad v_r \coloneqq \mathrm{ReLU}(u_r) - \mathrm{ReLU}(u_r - 4e_r(i)), \qquad \mu \coloneqq \sum_r v_r.$$

We have $0 \le u_r \le 4$, $v_r = e_r(i) u_r$, and $0 \le \mu \le 4$, because at most one local tile indicator is one. In particular $\mu \in \delta\mathbb{Z}$. The circuit

$$\mathrm{Step}_\delta(\mu - \tfrac{1}{2}) = \delta^{-1}\big(\mathrm{ReLU}(\mu - \tfrac{1}{2} + \delta) - \mathrm{ReLU}(\mu - \tfrac{1}{2})\big)$$

is exactly $\mathbb{1}\{\, \mu \ge 1/2 \,\}$: on the grid, a negative argument is at most $-\delta$, and a nonnegative argument yields a difference δ. Conjoin this Boolean output with the direction's enable flag, which includes ν_i and the appropriate endpoint test.

On inputs with legal tile labels and distinct coordinates, the selected positions are either BOS alone or BOS and one legal neighbor. Since $\mathrm{round}(1/2) = 1/2$, the returned vector is respectively zero or the neighbor's exact one-hot label. Thus $\mu \in \{0, 1\}$ and the output has the stated meaning. On other inputs the output is still Boolean and bounded, even when returned coordinates are fractional. ■

Global check and ranges. At each position form the OR of eleven Boolean flags: duplicate coordinate, illegal label, four nonwhite boundary edges, wrong seed at $(0, 0)$, and four enabled neighbor violations. To check an outside edge, select the precomputed indicator that the corresponding side of the local tile is not white, and conjoin it with validity and the corresponding endpoint flag. The seed flag is $\nu_i \wedge \mathbb{1}\{\, x = 0 \,\} \wedge \mathbb{1}\{\, y = 0 \,\} \wedge \neg e_{t_0}(i)$. All flags vanish at invalid positions. A Boolean OR uses $\mathrm{ReLU}(w) - \mathrm{ReLU}(w-1)$, where the integer $w \in [0, 11]$ is their sum. Apply the global counting circuit with threshold one to the resulting violation flag, and conjoin its negation with the parser's format conditions.

Use the parser's parameters $k = \max\{1, \lceil \log_2(R+1) \rceil, \lceil \log_2(b+2) \rceil\}$, $s = 6k + 20$, $p = 2s + 14$, $\delta = 2^{-s}$, and $H = 2s$. The new lookup scores lie in $[-4nH, 0]$. The sums u_r, gated values v_r, and

total μ lie in $[0, 4]$. Gate preactivations lie in $[-4, 4]$. The step preactivations lie in $[-1/2, 7/2 + \delta]$. The largest new coefficient is $\delta^{-1} = 2^s$. Even the separately weighted terms of the step have magnitude at most $(7/2 + \delta)2^s$, below the positive representable bound $2^{p-s-1} - \delta = 2^{s+13} - \delta$. The sums used to compute carry bits, borrow bits, and tile indicators are bounded by b. Since $n \le b \le 2^k - 2$, all these affine/ReLU computations fit in the same format and are exact on the grid. The attention accumulations are evaluated as specified in Equation (3).

Hardness reduction. *There is a polynomial-time map from each instance $(n, \mathcal{D}, t_0)$ of the tiling problem defined above to a fixed-precision softmax transformer $T_{n,\mathcal{D},t_0}$ of constant depth, with one head per layer, no positional encodings, and a causal mask that includes the current position, such that*

$$L(T_{n,\mathcal{D},t_0}) \neq \varnothing \quad \Longleftrightarrow \quad (n, \mathcal{D}, t_0) \text{ admits a valid tiling.}$$

Its width is polynomial in $n + q$, its precision is $O(n + \log(q + 1))$, and its binary description has polynomial length. Every accepted word has length $2^{2n}(2n + \ell + 2)$. Consequently nonemptiness is NEXP-hard for this architecture, and emptiness is coNEXP-hard. The statement holds for downward rounding and round-to-nearest, and for average-hard attention with the rounding convention of Corollary 20.

Proof. A valid tiling, serialized in any order, satisfies all checks. Conversely, acceptance supplies exactly G^2 distinct coordinates with legal labels, so every cell occurs. For each grid edge, exactly one endpoint occurs later in the input. Its corresponding directional lookup returns the earlier endpoint's tile exactly, and checks the oriented matching colors. Thus every adjacency holds, and the boundary and seed checks establish a tiling. The parser and global checks are combined with five attention stages for coordinate queries: one checks for repeated coordinates and four check neighbors. The total number of attention stages is bounded by an absolute constant. All position-wise circuits have constant depth and polynomial size. Their coefficients are dyadic with polynomial bit length. The color compatibility tables are computed in polynomial time and use only Boolean coefficients. The precision recipe gives $p, s = O(\log(R + 1) + \log(b + 2))$ with $R = 2^{2n}$ and $b = 2n + \ell$. This is bounded by $O(n + \log(q + 1))$, the convenient upper bound stated above. These observations establish the claimed uniform reduction and resource bounds. ■

Membership for arbitrary transformers. Let T be any transformer in the model of Section 2.2, with dense description length S. First test the empty word. If T accepts a word of positive length, Corollary 10 gives one of length at most

$$W_T + 1 = L2^{pd+p+s} + 1 = 2^{\mathrm{poly}(S)}.$$

Guess such a word and evaluate it. Direct causal evaluation uses $O(LN^2 \,\mathrm{poly}(S))$ scalar operations on a word of length N. By Proposition 25, sums and rational operations use only $\mathrm{poly}(S)$ bits. Lemma 27 computes every correctly rounded exponential in $2^{\mathrm{poly}(p)}$ time. Thus the complete evaluation takes $2^{\mathrm{poly}(S)}$ time. Average-hard attention uses rational arithmetic alone and has the same short-word bound. Together with the reduction, this proves NEXP-completeness of nonemptiness and coNEXP-completeness of emptiness for both models.

A.8 Equivalence and universality

Corollary 24 (Equivalence and universality)**.** *Language equivalence and universality are coNEXP-complete under the architectural and rounding conventions of Theorem 13.*

Proof of Corollary 24. For inequivalent transformers T, T', Corollary 10 supplies a distinguishing word of length at most $W_T + W_{T'} + 1$. Guessing and evaluating this word takes nondeterministic exponential time by the membership argument in Appendix A.7. Thus inequivalence belongs to NEXP and equivalence belongs to coNEXP. Equivalence to a fixed transformer that rejects every word is exactly emptiness, giving coNEXP-hardness.

A transformer that accepts every word gives the same upper bound for nonuniversality through a short distinguishing word. To obtain hardness, let $x \in \mathbb{F}_{p,s}$ be the designated output coordinate of T. Append one layer with zero value map and output

$$\mathrm{ReLU}\left(\mathrm{round}(\delta - x)\right), \qquad \delta = 2^{-s}.$$

This is an allowed affine–ReLU–affine map. If $x > 0$, the grid gives $x \geq \delta$, so the new output is zero. If $x \leq 0$, then $\delta - x \geq \delta$. Rounding and clipping preserve strict positivity. The new transformer therefore recognizes $\Sigma^* \setminus L(T)$. This adds one layer, leaves the precision unchanged, and preserves the fixed alphabet, one head per layer, causal mask, and absence of positional encodings. It works under both quantizers and for fixed-precision average-hard attention because the added head has zero values. Consequently T is empty exactly when its complement is universal. The constant-depth hardness family gains only one layer, proving the result under an absolute depth bound. ■

A.9 Executable verification

As an additional check of the explicit construction, the supplementary code contains two separately structured implementations for the main consistency language: an evaluator of the register program and a compiler that produces the neural network's matrices, together with an exact evaluator of those matrices. The matrix compiler emits affine query, key, value, and feed-forward maps for all 29 transformer blocks. Its evaluator executes these maps, preserves signed registers, checks that temporary attention registers are reset to zero, and verifies representability of affine products, sums, and outputs. A score row is replaced by its zero/one rounded exponentials only after verifying that each score is zero or at most $-H$. Both implementations support downward and round-to-nearest-even.

For each quantizer, the matrix checks cover all 256 formatted $n = 1$ words, all 85 words of length at most three, 35 edited words, and 18 cases at $n = 2$. All 788 classifications agree with an independent semantic recognizer and the register evaluator. The full matrix suite generates both dense and COO exports and reloads them, reproducing every matrix coefficient exactly. The implementation also checks the fixed layer count and dimension formulas across $n = 1, \ldots, 4$.

For tiling, the register evaluator executes the circuit in Appendix A.7 that checks neighboring tiles, including queries at invalid positions and positions that encode illegal tile labels. Its checks include 748 tiling classifications, 66,594 tests of the local circuit, and 4,416 comparisons against an independent function that parses each word and directly checks the resulting tiling. They cover both quantizers, duplicate coordinates with mixed labels, malformed records, endpoint conditions, and numbers of selected positions near the point where the rounded reciprocal becomes zero. All checks pass. The matrix compilation tests concern the main consistency recognizer. The tiling tests evaluate its explicit register circuits.

Correctness for every input and the asymptotic bounds follow from the mathematical proofs. The executable checks establish agreement of the supplied finite implementations on the reported cases. Source files, test commands, and machine-readable results are included in the anonymized supplementary code package.

A.10 Accumulator size

The exact sums in Equations (2) and (3) specify the attention output. They admit an exactly equivalent implementation using $O(p)$ bits per scalar accumulator, independently of the number of summands. The accumulator format may be wider than $\mathbb{F}_{p,s}$. The claim is independence from input length, not accumulation in the original p-bit format. This observation applies to every transformer in Section 2.2, not only to the constructed family.

Proposition 25 (Bounded accumulation)**.** *For either rounding convention, every attention operation in Section 2.2 has an exactly equivalent implementation using $O(p)$ bits per scalar accumulator, independently of input length. Exact evaluation of affine maps and dot products of dimension at most d requires $O(p + \log(d + 1))$ bits per accumulator.*

Proof. Put $A := 2^{p-s-1}$ and $\delta := 2^{-s}$. Fix row i and write $U_j := \mathrm{round}(e^{S_{ij}})$ for $0 \leq j \leq i$. Put $Z := \sum_{j=0}^{i} U_j$ and first suppose $Z > 0$. Since $0 \leq U_j \leq A - \delta$, whenever $Z \geq 2^p$ we have

$$\frac{U_j}{Z} \leq \frac{A - \delta}{2^p} = \frac{\delta}{2} - \frac{\delta}{2^p} < \frac{\delta}{2}.$$

All normalized coefficients then round to zero. The capped running sum

$$\widehat{Z}_0 := 0, \qquad \widehat{Z}_{j+1} := \min\{\widehat{Z}_j + U_j, 2^p\}, \qquad 0 \leq j \leq i,$$

ends at $\widehat{Z}_{i+1} = \min\{Z, 2^p\}$. All coefficients use this completed denominator:

$$\alpha_j \coloneqq \operatorname{round}\left(\frac{U_j}{\widehat{Z}_{i+1}}\right) = \operatorname{round}\left(\frac{U_j}{Z}\right), \qquad 0 \le j \le i.$$

The equality holds because the completed denominator equals Z below the cap, and both denominators give zero coefficients at the cap. Its grid is $\delta\mathbb{Z}$, so $p + s + O(1)$ bits suffice, including a carry bit. If $Z = 0$, the same running sum detects that case exactly.

For the context, put $r_j \coloneqq U_j/Z$ and $\alpha_j \coloneqq \operatorname{round}(r_j)$. Downward rounding gives $\sum_j \alpha_j \le 1$. Under round-to-nearest, $\alpha_j > 0$ implies $r_j \ge \delta/2$ and

$$\alpha_j \le r_j + \delta/2 \le 2r_j.$$

Clipping can only reduce a nonnegative coefficient, so $\sum_j \alpha_j \le 2$ in this case as well. Every value coordinate has magnitude at most A. Consequently, for every subset J of the visible positions,

$$\left|\sum_{j\in J} \alpha_j v_j\right| \le \sum_j \alpha_j |v_j| \le 2A.$$

All products lie on the δ^2 grid. Thus the format $\mathbb{F}_{p+s+2,2s}$ stores every product and partial sum exactly, in any order. Rounding the final sum to $\mathbb{F}_{p,s}$ gives precisely Equation (3).

Finally, a product of two p-bit grid values has at most $2p + O(1)$ bits on the δ^2 grid. Each coordinate of an affine map, or a dot product, sums at most $d + 1$ terms, including a possible bias, so a further $O(\log(d + 1))$ guard bits suffice. These reductions are finite and their length is independent of the input length. ■

A.11 Rounding after each operation

We consider two changes to the evaluation of attention. They use the same grid and quantizer as Section 2.2, with clipping after every operation. Iterated sums are evaluated from left to right. Rounding and saturation after each operation appear in other fixed-point transformer models (London & Kanade, 2025). Nowak et al. (2026) also study attention in which the numerator and denominator are summed before division. The definitions here fix the precise rules for the following result.

In both rules, round every scalar product and addition in affine maps, dot products, and residual additions. For affine maps, start with the bias and add products in coordinate order. Compute $U_j \coloneqq \operatorname{round}(e^{S_{ij}})$ and the denominator by

$$\widehat{Z}^{(-1)} \coloneqq 0, \qquad \widehat{Z}^{(j)} \coloneqq \operatorname{round}(\widehat{Z}^{(j-1)} + U_j), \qquad 0 \le j \le i.$$

In the first rule, normalize each weight before summation. Put $\alpha_j \coloneqq \operatorname{round}(U_j/\widehat{Z}^{(i)})$ and compute

$$Y^{(-1)} \coloneqq 0, \quad Y^{(j)} \coloneqq \operatorname{round}\left(Y^{(j-1)} + \operatorname{round}(\alpha_j v_j)\right), \quad c_i \coloneqq Y^{(i)}.$$

In the second rule, sum the numerator before normalizing:

$$N^{(-1)} \coloneqq 0, \quad N^{(j)} \coloneqq \operatorname{round}\left(N^{(j-1)} + \operatorname{round}(U_j v_j)\right), \quad c_i \coloneqq \operatorname{round}(N^{(i)}/\widehat{Z}^{(i)}).$$

Vector operations are coordinatewise, and zero denominators return zero contexts.

Proposition 26 (Arithmetic robustness)**.** *The matrices of the consistency recognizer in Theorem 4, with the same parameters, recognize L_n under either of these two rules and either quantizer. In particular, the architecture and description bounds are unchanged.*

Proof. Write $Q \coloneqq 2^s$, $A \coloneqq 8192Q$, and $C \coloneqq A - \delta$. All nonzero attention stages use scores zero on the selected positions, including BOS, and at most $-H$ elsewhere. The proof below preserves this property stage by stage. Their rounded unnormalized weights are therefore zero or one. The products lie on the grid, and the bounds on partial sums in Appendix A.1 ensure that rounding after each operation preserves the affine computations. We verify the ranges of the changed attention contexts below.

First rule: normalize before summation. For m ordinary targets and BOS, the denominator is $\min\{m+1, C\}$. If it saturates, both its reciprocal and the reciprocal of $m+1$ round to zero, since $1/C < \delta/2$. Thus the common coefficient remains $\alpha = \mathrm{round}(1/(m+1))$. Write $\alpha = k\delta$ and

$$u \coloneqq \mathrm{round}\big(-\alpha(t-\tfrac{1}{2})\big).$$

The unrounded product lies on the $\delta/2$ grid, so

$$|u + \alpha(t-\tfrac{1}{2})| \le \delta/2$$

for both quantizers. Each ordinary contribution is exactly α. The negative contribution has magnitude at most t, and $m\alpha \le 2$, so no context partial sum saturates. The resulting context is $u + m\alpha$. If $m < t$, Lemma 6 gives $\alpha \ge 3\delta$, whence

$$u + m\alpha \le -\alpha/2 + \delta/2 \le -\delta.$$

If $m \ge t$ and $k \ge 1$, then

$$u + m\alpha \ge \delta\big(k(m-t) + (k-1)/2\big) \ge 0.$$

For $k = 0$ it is zero. Hence the original step circuit still computes GE_t.

Address contributions lie on the grid, so the original capped addresses are unchanged. Decoder products are integer multiples of α and hence also lie on the grid. Every coordinate has total nonnegative mass at most $2D$. Thus the decoder is unchanged as well. The conflict and global tests use the same threshold argument. All contexts retain the bounds of Appendix A.1. This proves correctness under the first rule, including malformed inputs.

Second rule: normalize after summation. The first selected position is BOS, with value $-(t-\frac{1}{2})$, and every later selected position contributes one. Thus the scalar numerator and denominator are

$$N_m \coloneqq \min\{m - t + \tfrac{1}{2}, C\}, \qquad \widehat{Z}_m \coloneqq \min\{m+1, C\}.$$

For $m < t$, neither saturates and

$$N_m/\widehat{Z}_m \le -\frac{1}{2t} \le -2\delta.$$

For $m \ge t$, the ratio increases from $1/[2(t+1)]$: before denominator saturation it is $1-(t+\frac{1}{2})/(m+1)$, then its numerator increases with fixed denominator until the ratio is one. Since $t \le Q/4$ and $Q \ge 4$,

$$N_m/\widehat{Z}_m \ge \frac{1}{2(t+1)} \ge \delta.$$

Both quantizers therefore preserve the sign with at least one grid unit of margin. The same step circuit works for all m, and the stored context lies in $[-(t-\frac{1}{2}), 1]$.

Addresses and decoding. With m visible hashes before the cap, the raw address now equals

$$\widetilde{\rho}_m = \mathrm{round}(m/(m+1)), \qquad m < R.$$

We retain the original cap constant $\rho_R = R\,\mathrm{round}(1/(R+1))$. Distinct addresses before the cap have gap at least $1/[R(R+1)] - \delta$, and the gap to the cap satisfies

$$\rho_R - \widetilde{\rho}_{R-1} \ge \frac{1}{R(R+1)} - (R+1)\delta > \frac{1}{4R^2}.$$

The last inequality follows from $Q > 16R^3$, implied by Equation (16). Raw addresses lie in $[0,1]$ even after denominator saturation, and the Boolean cap is correct by the threshold argument. Thus the original scale Λ still gives the required address separation and the segment selectors still have exactly the required zero/negative score pattern. The rank and validity tests follow.

At a valid boundary i, the decoder has $b+1$ selected positions and exactly one possible contributor to coordinate u. Its output is now

$$y_{i,u} = \mathrm{round}\left(\frac{D\xi_{i,u}}{b+1}\right).$$

It is zero if $\xi_{i,u} = 0$ and at least one otherwise, so the same clipping circuit recovers each bit. On any row, a decoder coordinate is nonnegative and at most D: before denominator saturation its numerator is at most D times the denominator, and afterwards its capped numerator is at most C. The validity gate zeros every invalid position. Consequently the decoded signs, conflict scores, and Boolean global checks have the same meanings as in the original proof.

All altered contexts satisfy the original bounds for affine and ReLU computations. The compilation in Appendix A.2 preserves signed registers exactly. Its negations, reconstructions, and residual additions remain exact on the grid. Therefore the same matrices recognize L_n under the second rule as well. ■

The proposition asserts equality of recognized languages. Temporary contexts need not coincide. Proposition 9, Corollary 10, and the complexity upper bounds derived from them apply to the model in Section 2.2 and its equivalent implementation in Proposition 25.

A.12 Effective evaluation of rounded exponentials

For the constructed networks, the score gap makes exponentiation a zero/one test. Membership in NEXP for arbitrary input transformers also requires a uniform evaluation bound for other scores.

Lemma 27. *For $x \in \mathbb{F}_{p,s}$, the value $\mathrm{round}(e^x)$ is computable in $2^{\mathrm{poly}(p)}$ bit time under downward rounding or round-to-nearest with a fixed polynomial-time tie rule.*

Proof. Correct rounding requires locating e^x relative to rational grid boundaries or midpoints. A lower bound on its distance from these thresholds tells us how accurately to approximate it. We first prove such a bound for positive rational exponents. Let a, b, u, v be positive integers at most J, where $J \geq 2$, and put $r \coloneqq a/b$ and $m \coloneqq 16J^2$. We claim

$$\left| e^r - \frac{u}{v} \right| \geq \eta_J \coloneqq \frac{1}{3J(36J^2m^2)^m}.$$

The following polynomials turn a positive integral into an integer linear combination of e^r and 1, allowing us to bound the distance from e^r to u/v. Define

$$f(t) \coloneqq \frac{t^m(a-bt)^m}{m!}, \qquad F(t) \coloneqq \sum_{j=0}^{2m} (-1)^j f^{(j)}(t).$$

All derivatives of f at zero are integers: the nonzero ones, of orders $m+j$ for $0 \leq j \leq m$, equal

$$(-1)^j \binom{m}{j} a^{m-j} b^j \frac{(m+j)!}{m!}.$$

The symmetry $f(r-t) = f(t)$ gives the same integrality at r. Thus $P \coloneqq F(r)$ and $Q_0 \coloneqq F(0)$ are integers. Since $F + F' = f$,

$$I \coloneqq \int_0^r e^t f(t)\, dt = Pe^r - Q_0 > 0.$$

Using $t(a-bt) \leq a^2/(4b)$, $m! \geq (m/e)^m$, and $e < 4$ gives

$$I \leq Je^J 16^{-m} < \frac{1}{2J} \leq \frac{1}{2v}.$$

In particular $P \neq 0$, since otherwise I would be a positive integer smaller than one. Bounding the endpoint derivatives gives

$$|P| \leq (2m)^m (a+b)^m \leq (4mJ)^m.$$

On $[r/3, 2r/3]$, we have $t(a-bt) \geq a^2/(9b)$, so

$$I \geq \frac{1}{3J(9Jm)^m}.$$

Now $vI = P(ve^r - u) + (Pu - vQ_0)$. If the integer $Pu - vQ_0$ is nonzero, then $|P|\,|ve^r - u| > 1/2$, which implies a stronger bound than claimed. Otherwise $|e^r - u/v| = I/|P| \geq \eta_J$. This proves the claim.

The score and every positive grid boundary or midpoint between adjacent grid values are rationals with $O(p)$-bit numerators and denominators. Thus $J = 2^{O(p)}$ suffices and $\log(1/\eta_J) = 2^{O(p)}$. Certified Taylor approximation to absolute error less than $\eta_J/4$ decides each boundary comparison in $2^{\mathrm{poly}(p)}$ bit time: a number of terms polynomial in $J + \log(1/\eta_J)$ suffices, and their rational numerators and denominators have polynomial bit length in that quantity. For $x < 0$, compare e^{-x} with the reciprocal of a positive boundary instead. For $x = 0$, use $e^0 = 1$ exactly. Zero boundaries are immediate. Binary search among the grid bins uses $O(p)$ such comparisons. Clipping at the endpoints and the fixed tie rule complete the evaluation. ■