CHEAP IN MEMORY, NOT IN CAPACITY: DEPTH-WEIGHTED SAMPLE COMPLEXITY OF GATED LOOKUP INJECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Lookup tables are treated as "free" parameters because they are indexed rather than used as matrix multiplies. We show they are not free statistically. Working in the tight VC framework for hard-attention Transformers, we study *gated lookup injection* (GLI), a class covering per-layer embeddings, n-gram tables, and conditional-memory modules, and give two results that together settle the question. First, an effective-depth accounting $\Phi = \sum_\ell w_\ell(L - \ell + 1)$ survives Transformer-specific T^3 attention branching and an input-keyed multiplicative gate, giving $\mathrm{VCdim} = O(\Phi \log(TL\Phi))$. Second and centrally, we construct a *lookup-only* recursive-retrieval witness that *attains* $\Omega(\Omega_\Lambda \log T)$ in the exact-arithmetic tightness window: every capacity-carrying parameter in the construction sits in the table, so indexed entries realise the same statistical capacity rate as ordinary parameters despite never appearing in a matrix multiply. At b-bit precision the depth weighting saturates at $\Theta(b/\log T)$ retrieval steps, yet each table entry still realises $\Omega(b)$ bits of capacity, matching the counting ceiling any parameter obeys: the capacity result survives finite precision intact, even though the placement lever collapses. State-keyed lookup admits only a weaker upper bound. Exact experiments verify the construction and branching claims; a GPU implementation confirms the precision law within 2% and scales the witness to 1.3×10^5 keys, checking $449{,}184$ labels with zero mismatches. A controlled placement study with a frozen backbone is consistent with Φ governing realised capacity, sublinearly, though SGD is not shown to attain the VC ceiling itself.

1 INTRODUCTION

Parameter counts have stopped being a single number. Small Gemma 3n and Gemma 4 variants report an "effective" count alongside a total that is roughly twice as large, the difference being Per-Layer Embedding (PLE) tables: one lookup per token per layer, indexed and added into the residual stream, never multiplied by another parameter (Google DeepMind, 2025; 2026). The design recurs elsewhere. Liu et al. (2026) scale n-gram embedding tables instead of experts and report a better Pareto frontier than mixture-of-experts, and Cheng et al. (2026); Sadhukhan et al. (2026); Zhou et al. (2026) build related lookup capacity into the layer stack. The shared engineering argument concerns memory traffic: a table row is fetched, so it can live in flash or CPU RAM, and only rows for present tokens are touched. Consistently, the largest Gemma 4 variants drop PLE, there being no flash/VRAM asymmetry to exploit on datacenter GPUs.

Nothing in that argument is statistical: a parameter never multiplied by another still changes the function computed and still has to be estimated from data. We answer this inside the tight parametric characterisation of hard-attention Transformers, $O(LW \log(TW))$ with a nearly matching $\Omega(LW \log(TW/L))$ (Yang et al., 2026), whose organising finding is that *placement matters*: attention's $O(NWT^3)$ branching polynomials give the signature multiplicative $\log T$ that feed-forward's $O(NTW)$ lacks. Lookup tables are the extreme case of that logic, which raises the question directly: are they cheap twice over, or are practitioners trading memory for estimation error at an unknown rate?

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.

Table 1: The gated-lookup-injection family. Definition 1 requires the key and the context signal to be functions of the input alone; state-keyed mechanisms receive the weaker upper bound of Theorem 7 (no lower bound). †Cheng et al. (2026); Sadhukhan et al. (2026); ‡Lample et al. (2019); Berges et al. (2025); He (2024).

Mechanism	Lookup key	Keyed by	Covered by
Per-layer embeddings (Google DeepMind, 2025; 2026)	token id	input	Thm. 6, 8
engram tables (Liu et al., 2026; Zhou et al., 2026)	hashed n-gram	input	Thm. 6, 8
Conditional memory, embedding modules†	input key	input	Thm. 6, 8
Input-side embedding table (baseline)	token id	input	Prop. 15
Product-key memory, memory layers‡	learned query	state	Thm. 7 (UB only)

A mechanism class rather than a product. We define *gated lookup injection* (Section 2): a table indexed by a key computed from the input, combined with a projection of the input embeddings, and injected at a chosen layer set Λ through a gate driven by the running hidden state. PLE is the deployed instance and Table 1 places the others. The decisive distinction is whether the lookup is input-keyed or state-keyed, and we price the latter separately.

Our contributions are:

- **An effective-depth upper bound for GLI.** Running the layerwise partition argument with the *live* parameter count at each layer yields

$$\Phi(\mathcal{A}) \;:=\; \sum_{\ell=1}^{L} W_{\le \ell} \;=\; \sum_{\ell=1}^{L} w_\ell\,(L-\ell+1), \tag{1}$$

  and Theorem 6 gives $\mathrm{VCdim} = O(\Phi \log(TL\Phi))$: the effective-depth quantity of Bartlett et al. (2019) survives Transformer-specific branching and a multiplicative lookup gate rather than collapsing to a coarser global-W bound. State-keyed lookup receives only a weaker bound (Theorem 7), isolating input-keying as the property that makes the next result possible.
- **A lookup-only witness that attains it, our central result.** Theorem 8 constructs an architecture in which *every* capacity-carrying parameter lives in the table and the backbone uses only $O(pL)$ parameters, yet shatters $\Omega(\Omega_\Lambda \log T)$ points; under uniform placement this is $\Theta(W_{\mathrm{lut}} L \log T)$, the rate of ordinary parameters. The table is thus not merely covered by the upper bound, its entries *realise* the depth-weighted capacity, matching it up to constants in the regime $\Phi = T^{O(1)}$ (Proposition 11).
- **A precision-saturation law.** At b-bit precision the depth lever saturates after $\Theta(b/\log T)$ retrieval steps, yet each table entry still carries $\Omega(b)$ bits, matching the universal Wb counting ceiling (Corollary 13): the capacity result survives quantisation intact even though the placement advantage collapses.
- **A placement law, and a factor-of-two baseline.** Moving a fixed table budget into the last λ of L layers cuts its contribution to Φ by $\Theta(L/\lambda)$ at unchanged parameters, memory and FLOPs (Corollary 14), answering a depth-allocation question Liu et al. (2026) leave open; uniformly placed GLI tables already beat an equally sized input-side table by a factor of two before this lever is even pulled (Proposition 15).

2 Setup

2.1 Backbone

We adopt the architecture class of Yang et al. (2026). A depth-L Transformer maps $H^{(0)} \in \mathbb{R}^{T\times d}$ through L blocks, each an average-hard causal multi-head attention sublayer followed by a position-wise ReLU FFN with at most $D_0 = O(1)$ internal affine layers; residuals are permitted, layer normalisation is not. Hard attention, standard in this literature (Hahn, 2020; Merrill et al., 2022), takes $M_i = \{j \le i : \langle Q_i, K_j\rangle = \max_{k\le i}\langle Q_i, K_k\rangle\}$ and $\mathsf{Attn}(H)_i = |M_i|^{-1}\sum_{j\in M_i} V_j$. Positional encodings are arbitrary but fixed and not counted as parameters; classifiers read $f_\theta^{01}(H) =$

$\mathbf{1}[w_{\text{out}}^{\top} H_{T-1}^{(L)} > 0]$; arithmetic is exact over $\mathbb{R}$. Since a lookup table is indexed by a discrete key, inputs are token sequences $X \in \Sigma^T$ with $|\Sigma| = K$, and we fix a non-trainable injective embedding $\mathsf{TE}_0 : \Sigma \to \mathbb{R}^d$ with $H_i^{(0)} = \mathsf{TE}_0(x_i) + \mathsf{PE}(i)$. This is a restriction rather than an extension: it removes trainable parameters relative to Yang et al. (2026), so the lower bound is stronger for it, while the upper bound applies verbatim because $H^{(0)}$ is constant in θ.

2.2 Gated lookup injection

Definition 1 (Gated lookup injection)**.** Fix a slice set $\Lambda \subseteq [L]$, a width p and a key map $\kappa : \Sigma \to [K]$. A single packed table $E \in \mathbb{R}^{K \times |\Lambda| p}$ is looked up once per token and reshaped, writing $E^{(\ell)}[\kappa(x_i)] \in \mathbb{R}^p$ for its slice at $\ell \in \Lambda$. A dense map $P_{\text{proj}} : \mathbb{R}^d \to \mathbb{R}^{|\Lambda| p}$ is applied to the layer-0 embeddings. With a fixed scalar $s > 0$,

$$z_i^{(\ell)} \;=\; s\big(P_{\text{proj}}^{(\ell)} H_i^{(0)} \;+\; E^{(\ell)}[\kappa(x_i)]\big) \;\in\; \mathbb{R}^p, \tag{2}$$

injected as a third residual sub-block after the attention+FFN block of layer ℓ completes:

$$h_i^{(\ell)} \;\leftarrow\; h_i^{(\ell)} \;+\; P_{\text{out}}^{(\ell)}\big(\sigma\big(P_{\text{in}}^{(\ell)} h_i^{(\ell)}\big) \odot z_i^{(\ell)}\big), \tag{3}$$

with $P_{\text{in}}^{(\ell)} : \mathbb{R}^d \to \mathbb{R}^p$, $P_{\text{out}}^{(\ell)} : \mathbb{R}^p \to \mathbb{R}^d$ and σ a fixed piecewise-linear activation. The trainable parameters of the mechanism are

$$W_{\text{lut}} \;=\; \underbrace{K\,|\Lambda|\,p}_{\text{table}} \;+\; \underbrace{3\,|\Lambda|\,d\,p}_{P_{\text{proj}}, P_{\text{in}}, P_{\text{out}}}, \tag{4}$$

and the table dominates whenever $K \gg d$, as it does in practice.

We take $\sigma = \mathrm{ReLU}$, any fixed piecewise-linear choice behaving identically. Deployed implementations normalise, and the two places they might do so behave differently: normalisation on the injection path is free, because its normalisers are input-keyed and therefore few, while normalisation on the residual stream we cannot price (Section B). The backbone class below carries neither. Together with hard rather than softmax attention these are the standing idealisations of this line of work, and we do not claim they are innocuous. Definition 1 follows the reference implementation of PLE rather than the model cards, and the difference matters.

Remark 2 (Input-keyed versus state-keyed)**.** Both $\kappa(x_i)$ and the argument of P_{proj} in (2) depend on the input alone. Two consequences follow. First, $z_i^{(\ell)}$ is degree 1 in θ, which prevents the multiplicative gate in (3) from doubling the polynomial degree per layer (Lemma 4). Second, the pipeline reads position i only, so it adds no attention-type comparisons (Proposition 5). Mechanisms whose lookup is addressed by a learned query on the hidden state (Lample et al., 2019; Berges et al., 2025) violate the first; Theorem 7 prices that.

2.3 Liveness and the depth-weighted budget

Definition 3 (Liveness, effective depth)**.** A parameter θ_j is *live at layer* ℓ if some layer-ℓ hidden representation or branching polynomial depends on it. Let w_ℓ be the number of parameters whose first live layer is ℓ, let $W_{\le \ell} = \sum_{\ell' \le \ell} w_{\ell'}$, and set $\Phi(\mathcal{A}) = \sum_{\ell=1}^{L} W_{\le \ell} = \sum_{\ell=1}^{L} w_\ell (L - \ell + 1)$. Equivalently $\Phi = \bar{L} W$ for the effective depth $\bar{L}$ of Bartlett et al. (2019).

Slice ℓ of E and of P_{proj} is first live at layer ℓ, because $z^{(\ell)}$ enters the computation only at (3); that the row is fetched earlier for engineering reasons is irrelevant, since liveness is a property of the computation graph of the branching polynomials. Writing $w_\ell^{\text{lut}} = Kp + 3dp$ for $\ell \in \Lambda$ and 0 otherwise,

$$\Omega_\Lambda \;:=\; \sum_{\ell \in \Lambda} \big(Kp + 3dp\big)\,(L - \ell + 1) \;=\; \Theta\Big(Kp \sum_{\ell \in \Lambda} (L - \ell + 1)\Big). \tag{5}$$

An input-side embedding table, by contrast, is live at layer 0 and weighted by the full depth L. That is the standard situation, and the reason the uniform bound reads WL.

A parameter live at several layers is charged its *first* live layer (Remark 19), which is what makes the depth weighting a property of *untied* per-layer slices — how the deployed mechanisms are built — rather than a bookkeeping artefact.

3 Upper Bound

3.1 Degree recursion

Within a region of parameter space on which all discrete decisions are frozen, each hidden coordinate is a polynomial in θ; let $\deg_\ell$ bound its degree at layer ℓ.

Lemma 4 (Additive degree under input-keyed gating). *On any region where all attention argmax sets and all ReLU signs are fixed, the sub-block (3) satisfies* $\deg_\ell \le \deg_{\ell-1} + c$ *for an absolute constant* c*, so* $\deg_\ell = O(\ell)$ *and the pre-threshold output has degree* $O(L)$*. A state-keyed gate* $P_{\mathrm{out}}(\sigma(P_{\mathrm{in}}h^{(\ell)}) \odot Qh^{(\ell)})$ *instead satisfies* $\deg_\ell \ge 2\deg_{\ell-1}$ *in general, giving* $2^{\Theta(L)}$*.*

Proof sketch. $H^{(0)}$ is constant in θ, so $z_i^{(\ell)}$ is affine in θ by (2); on the region σ is a fixed mask, the Hadamard product adds degrees, and P_{out} adds one. State-keyed, both factors have degree $\deg_{\ell-1}+1$ and generic weights attain $2\deg_{\ell-1}+2$ (Section A). □

This is the distinction Yang et al. (2026) draw for gated activations, where degree $2^{\Theta(L)}$ inflates the bound by a factor of L. The gate of Definition 1 resembles the expensive case and behaves like the cheap one, because of Remark 2. Section 7.1 measures both branches exactly.

3.2 Branching budget

Proposition 5 (Branching budget). *Over* N *inputs of length* T*, the sub-block at layer* ℓ *contributes exactly* NTp *branching polynomials, the pre-activations of* $P_{\mathrm{in}}^{(\ell)}$*, and no attention-score comparisons. Hence* $|\mathcal{B}^{(\ell)}_{\mathsf{Attn}}| = O(Nw_\ell T^3)$*,* $|\mathcal{B}^{(\ell)}_{\mathsf{FFN}}| = O(Nw_\ell T)$ *and* $|\mathcal{B}^{(\ell)}_{\mathsf{GLI}}| = O(NpT)$*.*

What matters is not that pT is smaller than wT^3, but that the family carries no T-dependence beyond linear, so it cannot generate the multiplicative $\log T$ signature of attention.

3.3 Main bounds

Theorem 6 (Depth-weighted VC upper bound). *Let* $\mathcal{A}$ *be any GLI-augmented hard-attention architecture as in Section 2, with depth* L*, context length* T*, parameter count* W *and depth-weighted budget* $\Phi = \Phi(\mathcal{A})$*. Then* $\mathrm{VCdim}(\mathcal{F}^{01}_{\mathcal{A}}) = O(\Phi \log(TL\Phi))$*. For a depth-uniform architecture with* $w_\ell = \Theta(W/L)$ *we have* $\Phi = \Theta(WL)$ *and recover* $O(WL\log(TW))$*; for a GLI architecture,*

$$\mathrm{VCdim} \;=\; O\big((\Phi_{\mathrm{bb}} + \Omega_\Lambda)\log(TL(\Phi_{\mathrm{bb}} + \Omega_\Lambda))\big), \tag{6}$$

with Ω_Λ *as in* (5)*.*

Proof sketch. Refine the parameter space layer by layer as in Bartlett et al. (1998; 2019); Yang et al. (2026), but apply Warren's theorem (Warren, 1968; Goldberg & Jerrum, 1995) at layer ℓ in the $W_{\le\ell}$ variables on which the layer-ℓ branching polynomials actually depend. This is valid because a parent region's children are counted by sign patterns of polynomials independent of $\theta_{>\ell}$. The attention and FFN branchings at layer ℓ number $O(Nw_\ell T^3)$ and $O(Nw_\ell T)$ with $w_\ell \le W_{\le\ell}$, so the ratio $m_\ell/W_{\le\ell}$ remains $O(NT^3)$; by Lemma 4 the degrees are $O(\ell)$, and by Proposition 5 the injection adds nothing of a new type. Taking logarithms gives $N \le O(\Phi\log(NTL))$, and inverting gives the claim. Full proof in Section A. □

Theorem 7 (State-keyed lookup). *If the gate reads the running hidden state as in Lemma 4, the same argument gives* $\mathrm{VCdim} = O\big(\Phi(L + \log(TL\Phi))\big)$*, an extra additive* ΦL*.*

This is an upper bound only; we have no matching state-keyed witness.

4 Lower Bound

The upper bound leaves open the possibility that lookup tables cannot use their capacity and that (6) is loose. We rule this out. The witness adds a gather layer 0 and a chain loaded at ℓ_0 performs

$L - \ell_0$ retrievals, so $(L - \ell)$ replaces $(L - \ell + 1)$, a factor at most 2 (Section C). The construction is a lookup-driven variant of the recursive retrieval machine of Yang et al. (2026): there the parameters that scale are an affine map sending a one-hot group identifier to a packed address, and here that role is played by the table itself.

Theorem 8 (Lower bound). *There are absolute constants $c, C_0 > 0$ such that for every $L \geq 2$, context length $T \geq C_0$, $p \geq 3$, $\Lambda \subseteq \{0, \ldots, L-1\}$ and key-space size $K \geq C_0(L+p)$, there is a GLI-augmented architecture with depth $L+1$, context length T and backbone parameter count $O(pL)$ such that*

$$\mathrm{VCdim}(\mathcal{F}_{\mathcal{A}}^{01}) \;\geq\; c\left(Kp\sum_{\ell\in\Lambda}(L-\ell)\right)\log T \;=\; \Omega(\Omega_\Lambda \log T). \tag{7}$$

With uniform allocation $\Lambda = [L]$ and $W_{\mathrm{lut}} = \Theta(W)$ this gives $\mathrm{VCdim} = \Omega(WL\log T)$: per parameter, lookup parameters are as statistically expensive as ordinary ones.

The construction. Fix $m = \lfloor p/3 \rfloor$, let T_c context positions satisfy $2^B \leq T_c$ with $B = \lfloor \log_2 T_c \rfloor$, and index samples by (τ, ℓ_0, k, i, u): τ a group token, $\ell_0 \in \Lambda$ a start layer, $k \in \{\ell_0 + 1, \ldots, L\}$ a readout layer, $i \in [B]$ a bit index, $u \in [m]$ a channel. The input places $\mathrm{bit}_i(t)$ at context position t, three markers encoding (ℓ_0, k, u), and τ at the query position. Four mechanisms interlock. The table row of τ stores, for each (ℓ_0, u), the $L - \ell_0$ addresses $s_{\tau,\ell_0,u,k} = \sum_i Y_{\tau,\ell_0,k,i,u} 2^i$ packed into a single real by base-$2T_c$ encoding. A hat gate on three replicated coordinates loads it into an address register at exactly the layer named by the start marker, which is the only use of $p \geq 3$. Each subsequent layer retrieves the context position nearest the register, which the packing makes exactly the next address, and reads the bit stored there. The FFN then performs the base-$2T_c$ left shift and a guarded accumulation, so the read-out returns the requested bit.

Table 4 in Section C traces one chain in numbers, which is the part of the argument worth seeing concretely: one real in one table entry drives $L - \ell_0$ successive exact retrievals, and it is the base-β shift that converts remaining depth into independently settable labels.

Distinct triples (τ, ℓ_0, u) occupy disjoint entries, so all $N = n\,m\,B\sum_{\ell_0\in\Lambda}(L - \ell_0)$ labels are independent; with $n = \Theta(K)$, $m = \Theta(p)$ this is $\Theta(\Omega_\Lambda \log T)$, and the backbone needs $O(pL)$ parameters (Section C).

The extra factor of L over a single-slice construction ($\Omega(Kp \log T)$) comes from using all $|\Lambda|$ slices as independent chain origins (Remark 25): slice ℓ_0 of row τ funds a chain of length $L - \ell_0$, which is what makes remaining depth, rather than raw count, the operative quantity.

Remark 9 (Bit complexity). The packing requires $\rho = (L - \min\Lambda)\log_2(2T_c)$ bits of precision, as does the construction of Yang et al. (2026), which is why the setting assumes exact arithmetic. Section 7.3 maps the boundary.

Theorem 8 is a statement about $\mathbb{R}$, and one should ask what survives at the precision models are actually trained at. The answer is a saturating law.

Theorem 10 (Precision-limited lower bound). *Suppose weights and activations carry a b-bit mantissa with round-to-nearest, and set $B_b := \lfloor (b - c_1)/\log_2(2T_c) \rfloor$ for an absolute constant c_1. Under the hypotheses of Theorem 8,*

$$\mathrm{VCdim}(\mathcal{F}_{\mathcal{A}}^{01}) \;\geq\; c\,Kp\log T\sum_{\ell\in\Lambda}\min\{L-\ell,\; B_b\}. \tag{8}$$

The proof truncates every chain after its first B_b addresses, so that the packed real carries at most $b - c_1$ significant bits and the retrieval margin of Lemma 29 survives round-off at every step (Section C).

5 CONSEQUENCES

5.1 HOW TIGHT IS TIGHT

Proposition 11 (Tightness window). *For a GLI architecture with $\Phi_{\mathrm{bb}} = O(\Omega_\Lambda)$, Theorems 6 and 8 give $\Omega_\Lambda \log T \lesssim \mathrm{VCdim} \lesssim \Omega_\Lambda \log(TL\Omega_\Lambda)$, so the two agree up to a factor $O(1+\log(L\Omega_\Lambda)/\log T)$; in particular if $\Omega_\Lambda \le T^c$ and $L \le T$ for a constant c then $\mathrm{VCdim} = \Theta_c(\Omega_\Lambda \log T)$.*

The constant in Θ_c is scale-sensitive: it degrades linearly in c, and the deployed regime has $T \sim 10^5$ and $\Omega_\Lambda \sim 10^{10}$, hence $c \approx 2$ and $\log(L\Omega_\Lambda)/\log T \approx 2$, so Proposition 11 holds at this scale with a factor-of-a-few constant rather than the asymptotically vanishing one small c would give. Sharpening this constant would require packing more than $\log_2 T_c$ bits per chain step, which the block bit-extraction refinement of Bartlett et al. (2019) does not obviously deliver here, since a chain step is a single retrieval that consumes the address register; we pose closing this gap as a concrete open problem for future work on the construction.

5.2 WHAT FINITE PRECISION DOES TO THE LEVER

Two facts bracket the ceiling once precision is bounded, and they temper the placement law considerably.

Proposition 12 (Counting ceiling). *If all W weights carry b bits then $|\mathcal{F}_{\mathcal{A}}^{01}| \le 2^{Wb}$, so $\mathrm{VCdim} \le Wb$. Thus the two independent ceilings imply $\mathrm{VCdim} \le \min\{O(\Phi \log(TL\Phi)), Wb\}$; this proposition does not claim that the minimum is attained in general. The depth-weighted term can bind only when b is large enough that the counting ceiling is above it.*

With $T \approx 10^5$, $\log_2(2T_c) \approx 17.6$, so $B_b = 1$ at $b = 24$ and $B_b \le 3$ at $b = 53$; and since $\bar{L} \approx 18$, $\log_2(TL\Phi) \approx 57$, Proposition 12 makes Wb the binding ceiling at any $b \le 64$. The question that matters in practice is therefore whether lookup entries reach *that* ceiling. They do:

Corollary 13 (Table entries saturate the bit budget). *Under Theorem 10, if $b \ge c_1 + \log_2(2T_c)$ and $\Lambda \subseteq \{0, \dots, L - B_b\}$, then $\mathrm{VCdim} \ge c'\, K|\Lambda| p\,(b - c_1)$ for an absolute constant $c' > 0$. When the table dominates W and $b \ge 2c_1$, this matches $\mathrm{VCdim} \le Wb$ up to constants.*

Proof. Every summand of (8) equals $B_b \ge 1$, and $B_b \log_2 T \ge \frac{1}{2}\frac{b-c_1}{\log_2(2T_c)} \cdot \frac{1}{2}\log_2(2T_c)$, using $\lfloor x \rfloor \ge x/2$ for $x \ge 1$ and $T \ge T_c \ge 2$. □

So the title claim survives bounded precision in the strongest form available: each table entry carries $\Theta(b)$ bits of capacity, as many as any parameter can. What does not survive is the placement lever: the early/late ratio attained by the witness is $\Theta(\min\{L, B_b\}/\min\{\lambda, B_b\})$, equal to 1 once $\lambda \ge B_b$ (Corollary 33). The hypothesis $b \ge c_1 + \log_2(2T_c)$ excludes bf16 at any useful context length; there the witness is silent.

5.3 A PLACEMENT LAW AND ITS BASELINE

Corollary 14 (Allocation). *Fix a table budget $W_{\mathrm{lut}} = K|\Lambda|p$ and write $\lambda = |\Lambda|$. Then Ω_Λ is maximised by placing Λ at the earliest layers, giving $\Omega_\Lambda = \Theta(W_{\mathrm{lut}} L)$, and minimised by the latest layers, giving $\Omega_\Lambda = \Theta(W_{\mathrm{lut}}\lambda)$. The ratio between the extremes is $\Theta(L/\lambda)$, attained in both directions within the window of Proposition 11.*

Proposition 15 (Comparison with an input-side table). *An input-side embedding table of* usable *size W_0, usable because such a table reaches the stream through one map and so saturates at $W_0 = Kd$ (Proposition 24) is live at layer 0 and contributes $\Theta(W_0 L)$ to Φ. A GLI table of the same size spread uniformly over $\Lambda = [L]$ contributes $\Omega_\Lambda = Kp\,L(L+1)/2 = W_{\mathrm{lut}}(L+1)/2$. In the depth-weighted budget, uniform GLI is therefore cheaper per parameter than an input-side table by a factor of $2 + o(1)$, and cheaper by $\Theta(L/\lambda)$ only if confined to λ late layers.*

For a deployed configuration ($L = 35$, $K = 2.6 \times 10^5$, $p = 256$, $W_{\mathrm{lut}} \approx 2.35$B, $W_{\mathrm{bb}} \approx 2.3$B), uniform allocation gives $\Omega_\Lambda \approx 18 W_{\mathrm{lut}}$ against $\Phi_{\mathrm{bb}} \approx 17.5 W_{\mathrm{bb}}$, so the tables carry half the budget; confining them to the last 7 layers cuts $\Phi_{\mathrm{bb}} + \Omega_\Lambda$ by $1.6\times$ at unchanged parameters, memory and

FLOPs. This is an exact-arithmetic exchange rate, not a generalisation guarantee, and by Corollary 13 it does not persist at bounded precision.

Finally, none of this capacity is intrinsic to the table. With every attention sublayer removed the read-out is a function of the query token alone, so at most 2^k labellings are realisable on a set using k distinct query tokens and $\mathrm{VCdim} \leq K$ regardless of L, p or W_{lut} (Proposition 32). Lookup parameters are powerful because attention can address them, which is also why the effective-support refinement one might hope for is vacuous (Remark 21).

6 Related Work

VC dimension of neural networks. The nearly tight $\Theta(WL\log(W/L))$-type bounds for piecewise-linear networks (Harvey et al., 2017; Bartlett et al., 2019) supply the effective-depth quantity Φ our accounting builds on. For Transformers, prior work studies norm-sensitive rates and expressivity (Edelman et al., 2022; Trauger & Tewari, 2024; Sinha et al., 2023; Hahn, 2020; Merrill et al., 2022; Sanford et al., 2023), while Yang et al. (2026) give the tight hard-attention parametric characterisation and identify attention branching as the source of $\log T$. We extend this machinery to input-keyed GLI, proving that effective depth survives sequence-dependent attention branching and a multiplicative lookup gate, and construct a table-only witness showing that lookup entries themselves attain the resulting rate.

Lookup-based capacity. Product-key memories (Lample et al., 2019; He, 2024; Berges et al., 2025; Huang et al., 2025) use hidden-state queries, whereas per-layer embeddings and n-gram tables (Google DeepMind, 2025; 2026; Liu et al., 2026; Zhou et al., 2026; Cheng et al., 2026; Sadhukhan et al., 2026) are input-keyed. We turn this into a statistical distinction: input-keyed lookup admits the table-only witness, while for state-keyed lookup the degree growth weakens the provable upper bound (Theorem 7). Liu et al. (2026) flag depth allocation as unresolved; Corollary 14 gives a capacity-theoretic answer within GLI.

7 Experiments

All experiments are exact synthetic verifications in `numpy`, float64 unless stated, with an exact `Decimal` mode for the high-precision regime; every number comes from a fixed-seed script listed in Section H. AI Assistance was used in this section for code development.

7.1 Degree recursion (Lemma 4 and Theorem 7)

We restrict θ to a random line $\theta_0 + tv$, freeze all branch decisions at $t = 0$, and propagate every hidden coordinate as an exact polynomial in t, so the degree is read off rather than estimated. Frozen masks can annihilate leading terms, so the maximum over 12 seeds is the estimator and the min–max band the spread (Figure 2a, Table 3).

The plain backbone is exactly $3L + 1$ at every depth. The input-keyed gate fits $5.77L + 1.64$, with consecutive-depth ratios settling at $1.24, 1.19, 1.14, 1.12$; the state-keyed gate gives $\log_2(\deg) \approx 1.07L$ with ratios $2.06, 2.03, 1.99$, a doubling per layer. Both halves of Lemma 4 hold, and the separation is the degree gap that costs Theorem 7 its factor of L.

7.2 Branching budget (Proposition 5)

Branching counts are static, so we test the operational consequence: walk a random line and count how often each family of discrete decisions flips, the exponent in T identifying the family. Sweeping $L \in \{2, 3, 4\}$, T up to 45 and 4 seeds, measured flip exponents are $T^{3.22}$ for attention (range $[3.18, 3.26]$ over depths), $T^{1.25}$ for the gate ($[1.18, 1.30]$) and $T^{1.31}$ for the FFN ($[1.23, 1.36]$), against predicted polynomial counts $T^{3.10}$, $T^{1.00}$, $T^{1.00}$ (Figure 2b). The positive bias on the linear families is expected, flip counts inheriting the degree factor; the cubic-versus-linear separation is what keeps the injection out of the $\log T$-generating term.

Table 2: **Controlled placement.** Five seeds per arm, table-only, frozen 12-layer backbone, $\sqrt{2}$ ladder. The table budget W_{lut} is identical within groups A and B, so the slice term is reported in units of Kp, i.e. as $\sum_{\ell\in\Lambda}(L-\ell)$ with layers indexed from 0 (this is (5) re-indexed); absolute Φ is in Table 14. Group A fixes the budget and varies $C_{\max}$; group B fixes $C_{\max}$ and varies the budget; group C widens $|\Lambda|$ to push the budget past the top of group A. The two arms in grey are the *same* Λ measured twice and estimate run-to-run noise. Across all thirteen arms (Table 14), $r(N^*, \Omega_\Lambda) = +0.959$ and $r(N^*, C_{\max}) = +0.726$ with the two predictors themselves correlated ($r = +0.767$); partialling, $r(N^*, \Omega_\Lambda \mid C_{\max}) = +0.911$ and $r(N^*, C_{\max} \mid \Omega_\Lambda) = -0.050$.

	Arm	$\Omega_\Lambda/(Kp)$	$C_{\max}$	N^*
A	$\{3,4,5\}$	24	7	241.0
	$\{0,4,8\}$	24	4	226.0
	$\{0,1,11\}$	24	1	181.0
B	$\{0,1,11\}$	24	1	211.0
	$\{5,10,11\}$	10	1	159.8
	$\{9,10,11\}$	6	1	100.4
C	$\{0,1,2\}$	33	10	241.0
	$\{0,1,2,3\}$	42	9	298.4
	$\{0,1,2,3,4\}$	50	8	362.0

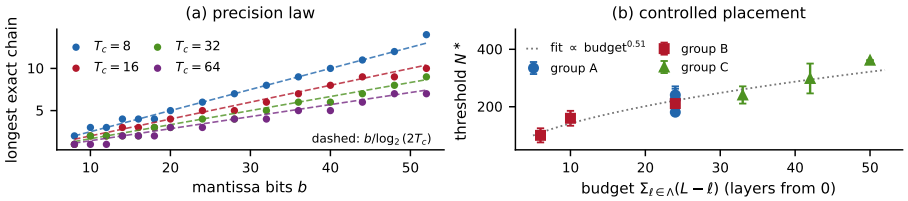


Figure 1: **(a)** Longest exactly realisable chain versus mantissa width; dashed lines are $b/\log_2(2T_c)$ with no fitted parameter. **(b)** The controlled placement experiment of Table 2: threshold N^* against the depth-weighted budget, coloured by group. Groups A and B isolate $C_{\max}$ and the budget respectively; group C extends the budget range and rules out a harness ceiling. Error bars are one standard deviation over five seeds.

7.3 The lower-bound witness (Theorem 8 and Remark 9)

This experiment does not estimate anything: it executes the proof. For each architecture we draw independent uniformly random labellings of the entire sample set, instantiate θ by the closed-form rule of Theorem 8, and check every sample; a single mismatch would falsify the theorem.

Across ten architectures (L from 3 to 16, $p \in \{3,6,9\}$, Λ early, late, uniform and spread), three labellings each, we observe zero mismatches over 6,204 labels (Table 6); one row pair isolates placement, since at identical $W_{\text{lut}} = 39$ the choice $\Lambda = \{0\}$ shatters 60 points and $\Lambda = \{4\}$ shatters 12, the $5\times$ ratio Ω_Λ predicts. A declared sweep of 435 configurations adds 75,298 labels with 5 mismatches, all in one configuration at $\rho = 50$ bits, within three of the double mantissa (Table 7); Section 7.5 pushes the same check far further. The crude predictor $\rho \le 53$ agrees with observed float64 success in all 20 cells of a $\{4,\dots,12\} \times \{8,\dots,64\}$ grid (Figure 3a), and re-running exactly the seven configurations that fail in float64 in exact `Decimal` arithmetic gives $0/778$ mismatches where float64 gives $67/778$ (Table 8).

7.4 Allocation: ceiling versus realised capacity

The ceiling is exact. Holding W_{lut} fixed at 162 and sliding Λ through a depth-10 stack, the shattered set size and Ω_Λ agree in ratio exactly, $4.500/3.000/2.833/1.000$ for earliest, middle, spread and latest placement, maximum deviation 0 (Figure 4a): Corollary 14's achievability direction. Removing attention caps realisable labellings at exactly 2^k on k query tokens, against 2590 of 2^{16} with attention active (Proposition 32, Table 10).

A backbone that cannot route. Ω_Λ counts what a *cooperating* backbone extracts from a parameter with $L - \ell + 1$ layers of remaining depth: in Theorem 8 the backbone spends exactly that depth on one retrieval per layer. Removing attention from the GPU harness by a single flag makes all four placements return the *identical* per-seed vector $(0, 0, 0, 64, 64)$, against $241.0, 211.0, 226.0, 113.2$ with attention active (Table 15): that arm sits at the floor, so placement structure *requires* routing rather than confirming Φ once routing is present (Proposition 32). A routing-free residual network behaves the same way, capacity *anti*-correlated with Φ at full Jacobian rank, so the effect is not first-order (Section E).

7.5 At scale, and on trained models

We use one A6000 (9.4 GPU-hours, peak 6.4 GB) only for the GPU-scale stress tests, using a torch port of the reference witness. These runs are independent validation of the analytic precision/scaling predictions; the main exact constructions and smaller sweeps are reproduced by the CPU scripts listed in Section H.

Precision law. Sweeping $b = 8$–52 and $T_c = 8$–64, the longest exact chain is linear in b, with slopes $0.2550, 0.1992, 0.1686, 0.1437$ versus the parameter-free $1/\log_2(2T_c) = 0.2500, 0.2000, 0.1667, 0.1429$, within 2% throughout (Figure 1a, Table 12).

Witness at scale. Across 24 configurations reaching $K = 131{,}091$, $T = 1028$, $p = 48$, $L = 32$ and shattered sets up to 1.9×10^7, we check $449{,}184$ labels in float64 with **zero mismatches** (Table 11).

Controlled placement. A four-placement sweep is confounded because the budget is collinear with $C_{\max}(\Lambda) := L - \max \Lambda$. We therefore fix one statistic while varying the other and add a third group extending the budget range (Table 2). At $C_{\max} = 1$, a $4\times$ budget change moves N^* by 110%; at fixed budget, group-A arms span 33%, versus 17% across two runs with the same Λ. With the widened range, controlling for $C_{\max}$ leaves $r(N^*, \Phi) = +0.911$, whereas controlling for Φ leaves $r(N^*, C_{\max}) = -0.05$ (Figure 1b, Section F.2). The response is sublinear (log–log slope 0.51), and a frozen random backbone also exposes early slices to more random nonlinear layers, which predicts the same ordering; the design does not separate the two, so we read it as consistent with, not proof of, Φ governing realised capacity.

Language modelling. The same controlled design on 11M-parameter models trained on a random 2-gram source gives train–validation gaps falling from 0.289 to 0.202 as the budget drops $4\times$ at $C_{\max} = 1$, with $r(\text{gap}, \Omega_\Lambda) = +0.916$ versus $r(\text{gap}, C_{\max}) = +0.659$ (Figure 5b, Table 16). One group-A arm sits 16% above the others, so a residual $C_{\max}$ effect survives, and Φ is backbone-dominated here; we treat this as preliminary (Section F.2).

8 Conclusion and Limitations

This paper's central claim is that lookup tables are not statistically free, a lookup-only construction attains the same depth-weighted capacity rate as ordinary, matrix-multiplied parameters, for the input-keyed GLI class and within the exact-arithmetic regime where the underlying upper bound is known to be tight. The claim survives bounded precision as well: each table entry still carries as many bits of capacity as any other parameter would. What does not survive is the placement advantage, which turns out to be a purely exact-arithmetic effect.

Several limitations remain. The analysis is distribution-free, treating token frequencies as fixed and Zipfian rather than modelling any real data distribution. It omits residual-stream normalisation and softmax attention. The state-keyed case has only an upper bound, with no matching construction. And the realised capacity we observe empirically grows sublinearly in the depth-weighted budget, in a way we have not yet managed to separate from the depth of random, untrained features.

Reproducibility Statement

All theoretical claims are proved in Sections A and C, with per-configuration results in Section D. Every CPU experiment is a deterministic `numpy` program with fixed seeds. Section H maps each claim to its script and raw output (Table 18) and lists every file of the supplement; the complete scripts, raw JSON/CSV outputs and figure scripts are in the supplement. The CPU pipeline runs in about eight minutes on one core; the GPU runs of Section 7.5 take about 9.4 GPU-hours on one A6000 (Section F).

References

Peter Bartlett, Vitaly Maiorov, and Ron Meir. Almost linear vc dimension bounds for piecewise polynomial networks. *Advances in neural information processing systems*, 11, 1998.

Peter L. Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019.

Vincent-Pierre Berges, Barlas Oğuz, Daniel Haziza, Wen-tau Yih, Luke Zettlemoyer, and Gargi Ghosh. Memory layers at scale. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 3831–3842, 2025.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.

Xin Cheng, Rui Tian, Wangding Zeng, Damai Dai, Qinyu Chen, Bingxuan Wang, Zhenda Xie, Kezhao Huang, Xingkai Yu, Chengqi Deng, Shangyan Zhou, Chenggang Zhao, Zhewen Hao, Yukun Li, Han Zhang, Zhengyan Zhang, Yixu Wei, M. Y Xu, Huishuai Zhang, Dongyan Zhao, and Wenfeng Liang. Conditional memory via scalable lookup: A new axis of sparsity for large language models, 2026. URL https://arxiv.org/abs/2601.07372.

Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In *International Conference on Machine Learning*, pp. 5793–5831. PMLR, 2022.

Paul W. Goldberg and Mark R. Jerrum. Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers. *Machine Learning*, 18(2):131–148, 1995.

Google DeepMind. Gemma 3n model overview. https://ai.google.dev/gemma/docs/gemma-3n, 2025.

Google DeepMind. Gemma 4 model overview. https://ai.google.dev/gemma/docs/core, 2026.

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

Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension bounds for piecewise linear neural networks. In *Annual Conference Computational Learning Theory*, 2017.

Xu Owen He. Mixture of a million experts. *arXiv preprint arXiv:2407.04153*, 2024.

Zihao Huang, Qiyang Min, Hongzhi Huang, Yutao Zeng, Defa Zhu, Ran Guo, et al. Ultra-sparse memory network. In *International Conference on Learning Representations*, volume 2025, pp. 86557–86575, 2025.

Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Large memory layers with product keys. *Advances in Neural Information Processing Systems*, 32, 2019.

Hong Liu, Jiaqi Zhang, Chao Wang, Xing Hu, Linkun Lyu, Jiaqi Sun, Xurui Yang, Bo Wang, Fengcun Li, Yulei Qian, et al. Scaling embeddings outperforms scaling experts in language models. *arXiv preprint arXiv:2601.21204*, 2026.

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

Ranajoy Sadhukhan, Sheng Cao, Harry Dong, Changsheng Zhao, Attiano Purpura-Pontoniere, Yuandong Tian, Zechun Liu, and Beidi Chen. Stem: Scaling transformers with embedding modules. *arXiv preprint arXiv:2601.10639*, 2026.

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

Ritwik Sinha, Zhao Song, and Tianyi Zhou. A mathematical abstraction for balancing the trade-off between creativity and reality in large language models. *arXiv preprint arXiv:2306.02295*, 2023.

Jacob Trauger and Ambuj Tewari. Sequence length independent norm-based generalization bounds for transformers. In *International Conference on Artificial Intelligence and Statistics*, pp. 1405–1413. PMLR, 2024.

Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity: festschrift for alexey chervonenkis*, pp. 11–30. Springer, 2015.

Hugh E Warren. Lower bounds for approximation by nonlinear manifolds. *Transactions of the American Mathematical Society*, 133(1):167–178, 1968.

Chenxiao Yang, Nathan Srebro, and Zhiyuan Li. Tight sample complexity of transformers. In Steve Hanneke and Tor Lattimore (eds.), *Proceedings of Thirty Ninth Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 6887–6923. PMLR, 29 Jun–03 Jul 2026. URL https://proceedings.mlr.press/v336/yang26a.html.

Wuyang Zhou, Yuxuan Gu, Giorgos Iacovides, Yuning Qiu, Qibin Zhao, and Danilo Mandic. Tensorizing engram: Sharing latents across n-gram embeddings is beneficial in llms, 2026. URL https://arxiv.org/abs/2606.08347.

USE OF AI ASSISTANCE

AI tools were used in two auxiliary roles: to help write and debug the verification scripts for the synthetic experiments, and for document preparation (formatting LaTeX tables, notation and cross-reference consistency). All theoretical results, proofs and claims are the authors' own and remain their responsibility.

A Proof of the Upper Bound

Throughout, $\mathcal{A}$ is a fixed GLI-augmented architecture as in Section 2, with depth L, context length T and trainable parameter vector $\theta \in \mathbb{R}^W$. We write $\mathcal{F} := \mathcal{F}_{\mathcal{A}}^{01}$ and recall the growth function

$$\Pi_{\mathcal{F}}(X_{1:N}) := \left|\{(f_\theta^{01}(X_1), \ldots, f_\theta^{01}(X_N)) : \theta \in \mathbb{R}^W\}\right|. \tag{9}$$

If $X_{1:N}$ is shattered then $\Pi_{\mathcal{F}}(X_{1:N}) = 2^N$, so any upper bound on $\Pi_{\mathcal{F}}$ yields one on $\mathrm{VCdim}(\mathcal{F})$.

Lemma 16 (Warren's bound; see (Warren, 1968; Goldberg & Jerrum, 1995) and (Bartlett et al., 1998, Lem. 1))**.** *Let $g_1, \ldots, g_m$ be polynomials of degree at most D in ν real variables, with $m \geq \nu$. The number of distinct sign vectors $(\operatorname{sgn} g_1(\theta), \ldots, \operatorname{sgn} g_m(\theta)) \in \{-1, 0, 1\}^m$ attained as θ ranges over $\mathbb{R}^\nu$ is at most $2(2emD/\nu)^\nu$.*

A.1 Liveness and the local variable count

Lemma 17 (Locality of layer-ℓ branching)**.** *For every ℓ, every layer-ℓ hidden representation $h_{n,t}^{(\ell)}(\theta)$ and every layer-ℓ branching polynomial is a function of $\theta_{\leq \ell}$ only, where $\theta_{\leq \ell}$ collects the $W_{\leq \ell}$ parameters live at layers $\leq \ell$.*

Proof. Induction on ℓ from Definition 3. $h^{(0)}$ is constant; the layer-ℓ sublayers read $h^{(\ell-1)}$, a function of $\theta_{\leq \ell-1}$ by induction, together with parameters live at layer ℓ. For the injection, $z^{(\ell)}$ depends on $E^{(\ell)}$, $P_{\mathrm{proj}}^{(\ell)}$ and on the *constant* $H^{(0)}$ by (2); slices $\ell' > \ell$ of the packed table are not read at layer ℓ. This is where the packed-table structure enters: because $z^{(\ell)}$ is consumed only at layer ℓ, slice ℓ is not live before layer ℓ. □

Remark 18. It is worth being explicit about why the packed table $E \in \mathbb{R}^{K \times |\Lambda| p}$ does not count as live at layer 0 even though a row of it is *fetched* once at layer 0 in deployed implementations. Liveness is a property of the computation graph of the branching polynomials, not of the order of memory accesses: the value $E^{(\ell)}[\kappa(x_i)]$ first influences a hidden state, and hence a branching decision, at layer ℓ. Reordering the lookup earlier for engineering reasons changes nothing. This is exactly the graph-theoretic notion of layer assignment used by Bartlett et al. (2019), restated for a packed table.

Remark 19 (Sharing)**.** A parameter live at several layers is charged its *first* live layer, which is what Lemma 17 needs. A mechanism tying one table across all layers therefore has every table parameter live at the first slice, so $\Phi = \Theta(WL)$ and the placement structure disappears: the depth weighting is a property of *untied* per-layer slices, which is how the deployed mechanisms are built.

A.2 Degree recursion

Proof of Lemma 4. Fix a region C on which all attention argmax sets $M_{n,i}$ and all ReLU signs, in the FFNs and in the gates, are constant. Let $\deg_\ell$ bound the degree in θ of every layer-ℓ representation on C.

Base case. $H^{(0)}$ is constant in θ, since inputs are token sequences with a fixed embedding, so $\deg_0 = 0$.

Attention. On C the weights $\alpha_{i,j} = \mathbf{1}[j \in M_{n,i}]/|M_{n,i}|$ are fixed rationals, so $\mathsf{Attn}(H)_{n,i} = \sum_{j \in M_{n,i}} \alpha_{i,j} V_{n,j}(\theta)$ is a fixed linear combination of the values, each an affine image of $H^{(\ell-1)}$: degree $\leq \deg_{\ell-1} + 1$. Residual addition does not increase degree.

FFN. At most $D_0 = O(1)$ internal affine maps, each with a frozen ReLU mask, so degree $\leq \deg_{\ell-1} + O(1)$.

Injection. By (2) and $H^{(0)}$ constant, $z_i^{(\ell)}$ is affine in $(E^{(\ell)}, P_{\mathrm{proj}}^{(\ell)})$, hence $\deg z_i^{(\ell)} \leq 1$. Let $\rho := \deg h^{(\ell)}$ for the post-FFN representation, so $\rho \leq \deg_{\ell-1} + O(1)$. Then $P_{\mathrm{in}}^{(\ell)} h^{(\ell)}$ has degree $\leq \rho + 1$, and on C the activation is a fixed mask, so $\sigma(P_{\mathrm{in}}^{(\ell)} h^{(\ell)})$ has degree $\leq \rho + 1$. The Hadamard product with $z^{(\ell)}$ adds degrees, giving $\leq \rho + 2$; applying $P_{\mathrm{out}}^{(\ell)}$ adds 1, and the residual addition takes a maximum. Hence

$$\deg_\ell \;\leq\; \max\{\rho,\ \rho + 3\} \;=\; \deg_{\ell-1} + O(1), \tag{10}$$

so $\deg_\ell = O(\ell)$ and $\deg_L = O(L)$. Including $w_{\text{out}} \in \theta$, the output logit has degree $\deg_{\text{out}} \le \deg_L + 1 = O(L)$.

State-keyed variant. If instead $z_i^{(\ell)} = Q^{(\ell)} h_i^{(\ell)}$ then $\deg z^{(\ell)} \le \rho + 1$, the Hadamard product has degree $\le 2\rho + 2$, the recursion becomes $\deg_\ell \le 2\deg_{\ell-1} + O(1)$, and $\deg_L = 2^{\Theta(L)}$. Generic weights attain this, since the leading coefficient of the product is the product of the two leading coefficients and does not vanish identically. This matches the gated-activation discussion of Yang et al. (2026), and is measured in Section 7.1. □

A.3 Branching sets

Proof of Proposition 5. Fix a layer ℓ and inputs $X_{1:N}$.

Attention. As in Yang et al. (2026), to freeze $M_{n,i}$ it suffices to fix the ternary signs of the pairwise score differences $g_{n,i,p',q}^{(\ell,h)}(\theta) = \langle q_{n,i}^{(\ell,h)}, k_{n,p'}^{(\ell,h)} - k_{n,q}^{(\ell,h)} \rangle$ for $0 \le p' < q \le i$. Each head contributes $\sum_{i=0}^{T-1} \binom{i+1}{2} = O(T^3)$ polynomials per input, and a head is nontrivial only if it owns parameters live at layer ℓ, so there are $O(w_\ell)$ of them: $|\mathcal{B}_{\mathsf{Attn}}^{(\ell)}| = O(N w_\ell T^3)$.

FFN. With $D^{(\ell)} \le D_0 = O(1)$ internal layers there are $O(w_\ell)$ ReLU units, each contributing one pre-activation per input and position: $|\mathcal{B}_{\mathsf{FFN}}^{(\ell)}| = O(N w_\ell T)$.

Injection. The only discrete decision in (3) is the sign of the p pre-activations $P_{\text{in}}^{(\ell)} h_i^{(\ell)}$, one per input and position: exactly NTp polynomials. Once these signs are fixed the gate is a frozen mask and, by Lemma 4, the sub-block output is polynomial. No score comparison arises because the sub-block reads only position i: the map $h_i \mapsto h_i + P_{\text{out}}(\sigma(P_{\text{in}} h_i) \odot z_i)$ involves no pair (i, j) with $i \ne j$. Hence $\mathcal{B}_{\mathsf{GLI}}^{(\ell)}$ contains no attention-type polynomial, and in particular contributes no factor of T^2 or T^3. □

A.4 Proof of Theorem 6

Proof. Assume $\mathrm{VCdim}(\mathcal{F}) \ge N$ and fix shattered inputs $X_{1:N}$; we bound N. If $N < W$ the claimed bound holds after increasing constants, since $\Phi \ge W$, so assume $N \ge W$.

Layerwise refinement. Define partitions $\mathcal{P}_0 = \{\mathbb{R}^W\}$ and, for $\ell = 1, \dots, L$, refine each $C \in \mathcal{P}_{\ell-1}$ by the ternary sign patterns of $\mathcal{B}_{\mathsf{Attn}}^{(\ell)}$, then of $\mathcal{B}_{\mathsf{FFN}}^{(\ell)}$, then of $\mathcal{B}_{\mathsf{GLI}}^{(\ell)}$, in that order, matching the execution order attention $\to$ FFN $\to$ injection. By the argument of Lemma 4, on each resulting region every layer-ℓ representation is a polynomial of degree $\deg_\ell = O(\ell)$, and by Lemma 17 all three branching families are polynomials in $\theta_{\le \ell}$ alone, of degree $D_\ell = O(\ell)$ (for attention, $D_\ell \le 2\deg_{\ell-1} + O(1)$).

Counting children in the local variables. Fix $C \in \mathcal{P}_{\ell-1}$. The children of C are indexed by sign patterns of polynomials that do not depend on $\theta_{>\ell}$. Therefore the number of children of C is at most the number of sign patterns those polynomials attain as $\theta_{\le \ell}$ ranges over $\mathbb{R}^{W_{\le \ell}}$, and Lemma 16 applies with $\nu = W_{\le \ell}$, padding with constant polynomials if $m < \nu$. Writing $\nu_\ell := W_{\le \ell}$,

$$\Delta_\ell \le \prod_{\mathcal{B} \in \{\mathcal{B}_{\mathsf{Attn}}^{(\ell)}, \mathcal{B}_{\mathsf{FFN}}^{(\ell)}, \mathcal{B}_{\mathsf{GLI}}^{(\ell)}\}} 2\Big(\frac{2e\,|\mathcal{B}|\,D_\ell}{\nu_\ell}\Big)^{\nu_\ell}, \qquad |\mathcal{P}_L| \le \prod_{\ell=1}^{L} \Delta_\ell. \tag{11}$$

This is the only place the argument departs from Yang et al. (2026), who use $\nu_\ell = W$ for every ℓ; that choice is also valid and yields $\Phi \le WL$. It is the same substitution Bartlett et al. (2019) make for feed-forward networks, and the content of the present theorem is that it survives the attention and gate families above.

Patterns within a final region. On $C \in \mathcal{P}_L$ every output logit $z_n(\theta) = \langle w_{\text{out}}, h_{n,T-1}^{(L)}(\theta) \rangle$ is a polynomial of degree $\deg_{\text{out}} = O(L)$ in all W variables, so by Lemma 16, $\Pi_C \le 2(2eN \deg_{\text{out}}/W)^W$.

Table 3: Exact output-polynomial degree along a random line ($d = 4$, $T = 4$, $p = 2$), reported as median (min–max) over 12 seeds. Frozen masks can only lower the realised degree, so the maximum is the estimator of the theoretical degree. See Section 7.1.

L	1	2	3	4	5	6	7	8
plain backbone	4	7	10	13	16	19	22	25
(min–max)	4–4	5–7	7–10	11–13	14–16	17–19	20–22	22–25
input-keyed gate	7	13	18	23	28	33	39	44
(min–max)	4–7	9–13	8–19	22–25	25–31	30–37	37–42	39–47
state-keyed gate	10	28	61	126	261	331	665	1162
(min–max)	4–10	12–28	8–64	46–136	70–280	147–568	332–1133	340–1820

Assembling. Using $\Pi_{\mathcal{F}}(X_{1:N}) \le |\mathcal{P}_L| \max_C \Pi_C$ and $2^N = \Pi_{\mathcal{F}}(X_{1:N})$, take base-2 logarithms:

$$N \;\le\; O(L) \;+\; \sum_{\ell=1}^{L} \nu_\ell \Big[\log \tfrac{C|\mathcal{B}^{(\ell)}_{\mathsf{Attn}}|D_\ell}{\nu_\ell} + \log \tfrac{C|\mathcal{B}^{(\ell)}_{\mathsf{FFN}}|D_\ell}{\nu_\ell} + \log \tfrac{C|\mathcal{B}^{(\ell)}_{\mathsf{GLI}}|D_\ell}{\nu_\ell} \Big] \;+\; W \log \tfrac{CNL}{W}. \tag{12}$$

By Proposition 5, $|\mathcal{B}^{(\ell)}_{\mathsf{Attn}}| = O(Nw_\ell T^3)$ with $w_\ell \le \nu_\ell$, so the first ratio is $O(NT^3)$; similarly the second is $O(NT)$. For the third, $|\mathcal{B}^{(\ell)}_{\mathsf{GLI}}| = O(NTp)$ and $p \le \nu_\ell$ whenever layer ℓ carries a slice, since a slice contributes $Kp \ge p$ live parameters, so the ratio is $O(NT)$; if $\ell \notin \Lambda$ the term is absent. With $D_\ell = O(\ell) = O(L)$ each bracket is $O(\log(NT^3L)) = O(\log(NTL))$. Hence

$$N \;\le\; O\Big(\big(\textstyle\sum_\ell \nu_\ell + W\big) \log(NTL) \Big) \;=\; O\big(\Phi \log(NTL)\big), \tag{13}$$

using $\Phi = \sum_\ell \nu_\ell \ge W$. Finally apply the standard inversion: if $N \le A \log N + B$ with $A \ge 2$ then $N = O(A \log A + B)$. With $A = O(\Phi)$ and $B = O(\Phi \log(TL))$ we obtain $N = O(\Phi \log \Phi + \Phi \log(TL)) = O(\Phi \log(TL\Phi))$.

Specialisations. If $w_\ell = \Theta(W/L)$ for all ℓ then $\Phi = \Theta((W/L)\sum_\ell (L - \ell + 1)) = \Theta(WL)$ and the bound reads $O(WL \log(TLW)) = O(WL \log(TW))$ for $L \le W$, recovering Yang et al. (2026). For a GLI architecture, $w_\ell = w_\ell^{\mathrm{bb}} + \mathbf{1}[\ell \in \Lambda](Kp + 3dp)$, so $\Phi = \Phi_{\mathrm{bb}} + \Omega_\Lambda$ with Ω_Λ as in (5), giving (6). □

Proof of Theorem 7. Identical, except that by Lemma 4 the degrees are now $D_\ell = 2^{O(\ell)}$, so each bracket in (12) becomes $O(\ell + \log(NT^3L)) = O(L + \log(NTL))$ instead of $O(\log(NTL))$. Summing gives $N \le O(\Phi(L + \log(NTL)))$ and inverting gives $N = O(\Phi(L + \log(TL\Phi)))$. □

Corollary 20 (Sample complexity)**.** *By the fundamental theorem of statistical learning (Vapnik & Chervonenkis, 2015; Blumer et al., 1989), PAC learning $\mathcal{F}^{01}_{\mathcal{A}}$ in the realisable setting requires $O\big((\Phi \log(TL\Phi) \log(1/\varepsilon) + \log(1/\delta))/\varepsilon\big)$ samples, and $\Omega\big((\Omega_\Lambda \log T + \log(1/\delta))/\varepsilon\big)$ samples are necessary for the class of Theorem 8.*

Proof of Proposition 11. Immediate from Theorems 6 and 8: with $\Phi_{\mathrm{bb}} = O(\Omega_\Lambda)$ we have $\Phi = \Theta(\Omega_\Lambda)$, so the upper bound reads $O(\Omega_\Lambda \log(TL\Omega_\Lambda))$ and the lower $\Omega(\Omega_\Lambda \log T)$; their ratio is $O(\log(TL\Omega_\Lambda)/\log T) = O(1 + \log(L\Omega_\Lambda)/\log T)$. If $\Omega_\Lambda \le T^c$ and $L \le T$ then $\log(TL\Omega_\Lambda) \le (c + 2) \log T$. □

Remark 21 (Why the effective-support refinement fails)**.** One might hope to replace K by $\min(K, NT)$ in Ω_Λ, since on N sequences of length T at most NT table rows are read, and unread rows appear in no branching polynomial. The substitution is legitimate, being an instance of Lemma 17, but useless. Applying Theorem 6 with $\Phi_{\mathrm{eff}} = \Phi_{\mathrm{bb}} + \min(K, NT)p \sum_{\ell \in \Lambda}(L - \ell + 1)$ and evaluating at $N = \mathrm{VCdim}$ gives, in the regime $K \ge NT$, the inequality $N \le O((\Phi_{\mathrm{bb}} + NTpL^2) \log(\cdot))$, whose right-hand side is superlinear in N; the fixed point is vacuous. The refinement bites only when the input domain restricts to a genuine sub-vocabulary $\Sigma' \subsetneq \Sigma$, in which case K is replaced by $|\Sigma'|$ throughout, both here and in Theorem 8.

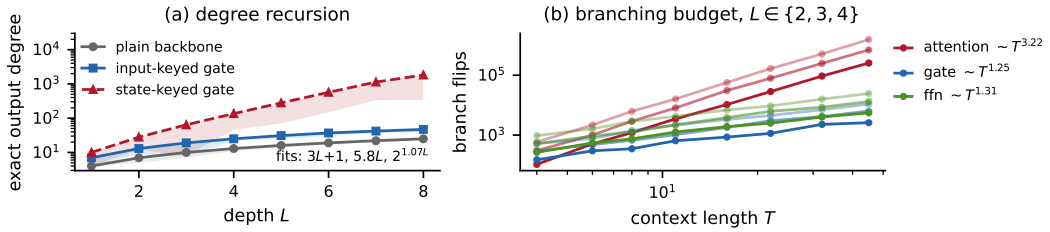


Figure 2: **(a)** Exact output degree versus depth; line is the max over 12 seeds, band is min–max. **(b)** Branch flips along a random line, for $L = 2, 3, 4$ (decreasing opacity), against predicted polynomial counts $T^{3.10}$, $T^{1.00}$, $T^{1.00}$.

B Normalisation, and what the input-side baseline can use

Two questions the main text defers. Both turn on the input-keyed property of Remark 2, which is why they are collected here rather than proved separately.

B.1 Normalisation on the injection path is free

Proposition 22 (Injection-path normalisation)**.** *Replace $z_i^{(\ell)}$ in (3) by its root-mean-square normalisation $\tilde{z}_i^{(\ell)} = z_i^{(\ell)} / \sqrt{p^{-1}\|z_i^{(\ell)}\|^2 + \varepsilon}$, as deployed implementations of PLE do. Then Theorem 6 holds unchanged:* $\mathrm{VCdim}(\mathcal{F}_{\mathcal{A}}^{01}) = O(\Phi \log(TL\Phi))$.

Proof. Write $q^{(\ell)}(x, \theta) := p^{-1}\|z^{(\ell)}(x, \theta)\|^2 + \varepsilon$, so that $\tilde{z} = z/\sqrt{q}$.

Step 1: how many distinct radicals there are. By (2), $z_i^{(\ell)}$ depends on the input only through $\kappa(x_i)$ and on θ only through the slice-ℓ parameters $(E^{(\ell)}, P_{\mathrm{proj}}^{(\ell)})$, because $H^{(0)}$ is constant in θ. Hence $q^{(\ell)}$ takes at most one value per pair $(\kappa, \ell) \in [K] \times \Lambda$, whatever N and T are: two positions carrying the same token, in the same sequence or in different ones, share a radical. Let $\mathcal{Q}$ enumerate the distinct pairs occurring, so $|\mathcal{Q}| \le K|\Lambda|$.

Step 2: adjoining them. Introduce variables $y = (y_{\kappa,\ell})_{(\kappa,\ell)\in\mathcal{Q}}$ and work on

$$\mathcal{V} := \big\{(\theta, y) : \; y_{\kappa,\ell}^2 \, q_\kappa^{(\ell)}(\theta) = 1, \; y_{\kappa,\ell} > 0\big\}, \tag{14}$$

well defined since $q \ge \varepsilon > 0$, and on which $\tilde{z}_i^{(\ell)} = y_{\kappa(x_i),\ell}\, z_i^{(\ell)}$ is a *polynomial* of degree at most 2 in (θ, y). The map $\theta \mapsto (\theta, y(\theta))$ is a bijection onto $\mathcal{V}$, so the sign patterns realised by the branching polynomials as θ ranges over $\mathbb{R}^W$ are exactly those realised on $\mathcal{V}$.

Step 3: the count. Re-run the proof of Theorem 6 on $\mathcal{V}$. Lemma 4 is unaffected: the substitution multiplies z by one adjoined variable, so the injection still adds $O(1)$ to the degree per layer and $\deg_\ell = O(\ell)$. Proposition 5 is unaffected: the discrete decisions are the same NTp gate pre-activations. The only change is that layer ℓ carries $\nu_\ell + |\mathcal{Q}_{\le\ell}|$ local variables instead of ν_ℓ, and that sign patterns are counted on a variety cut out by $|\mathcal{Q}|$ equations of degree 3, for which the Milnor–Thom bound (Warren, 1968; Goldberg & Jerrum, 1995) replaces $2(2emD/\nu)^\nu$ by $2(CmD/\nu)^{\nu+|\mathcal{Q}|}$ with C absolute. Summing over layers replaces Φ by $\Phi + L|\mathcal{Q}|$. Since each slice contributes Kp live parameters at its own layer,

$$L\,|\mathcal{Q}| \;\le\; L\,K|\Lambda| \;\le\; Kp\sum_{\ell\in\Lambda}(L-\ell+1) \;=\; \Omega_\Lambda \;=\; O(\Phi), \tag{15}$$

using $p \ge 1$ and $L \le \sum_{\ell\in\Lambda}(L - \ell + 1)$ for any nonempty $\Lambda \subseteq \{0, \dots, L-1\}$. The budget is therefore unchanged up to an absolute constant, and the inversion gives $N = O(\Phi \log(TL\Phi))$ as before. □

Remark 23 (Residual-stream normalisation is not free)**.** If instead RMSNorm reads the running hidden state, as a pre-norm backbone does, the normaliser is a function of that state and q is

indexed by (n, i, ℓ): there are $\Theta(NTL)$ distinct radicals rather than $K|\Lambda|$. Step 3 then replaces Φ by $\Phi + \Theta(NTL^2)$ and the fixed-point inequality becomes $N \le O((\Phi + NTL^2)\log(NTL))$, whose right-hand side is superlinear in N; no finite N is excluded. This is the same failure as Remark 21, and for the same reason: a refinement indexed by samples rather than by parameters cannot close a fixed point in N. We leave pre-norm backbones open, and note that the distinction is again exactly input-keyed versus state-keyed.

B.2 The input-side baseline is rank-capped

Proposition 24 (Input-side tables saturate at $p' = d$)**.** *An input-side table $E_0 \in \mathbb{R}^{K\times p'}$ reaches the residual stream through a single map $A \in \mathbb{R}^{d\times p'}$, so the per-key injection it realises is a row of $M := E_0 A^\top \in \mathbb{R}^{K\times d}$, and the reachable set is exactly $\{M : \operatorname{rank} M \le \min(p', d)\}$. Its usable parameter count is therefore $K\min(p', d)$ and its contribution to Φ is $\min(W_0, Kd)\,L$; for $p' > d$ the extra columns change no function in the class.*

Proof. The injection is $x \mapsto A\,E_0^\top e_{\kappa(x)}$, the $\kappa(x)$-th row of M. Since $\operatorname{rank} M \le \min(K, p', d)$, the reachable set is contained in the stated one; conversely, given any M of rank at most $\min(p', d)$, a rank factorisation supplies E_0 and A. For $p' \ge d$ the constraint is vacuous and the reachable set is all of $\mathbb{R}^{K\times d}$, so increasing p' further adds nothing. The class is thus parametrised by $K\min(p', d)$ usable reals, all live at layer 0, giving $\min(W_0, Kd)L$ by Definition 3. □

Consequently Proposition 15's factor of two is a comparison at equal *usable* size. Comparing raw parameter counts beyond $p' = d$ would flatter the input-side baseline, crediting it with columns that change no function.

C Proof of the Lower Bound

We give the construction in full. It is convenient to index layers $0, 1, \ldots, L$, where layer 0 is a *gather* block and layers $1, \ldots, L$ are *retrieval* blocks; the total depth is $L + 1 = O(L)$. Slices are placed at $\Lambda \subseteq \{0, \ldots, L-1\}$. Because the injection at layer ℓ executes *after* that layer's attention, an address loaded at layer ℓ_0 is first usable at layer $\ell_0 + 1$, so a chain started at ℓ_0 performs $L - \ell_0$ retrievals. This is why Theorem 8 carries $(L - \ell)$ where (5) carries $(L - \ell + 1)$; the two agree up to a factor ≤ 2 for $\Lambda \subseteq \{0, \ldots, L-1\}$ and $L \ge 2$.

C.1 Sizes and sample set

Let T_c be the number of context positions and

$$B := \lfloor \log_2 T_c \rfloor, \qquad \beta := 2T_c, \qquad m := \lfloor p/3 \rfloor, \qquad T = T_c + 4. \tag{16}$$

Positions $0, \ldots, T_c - 1$ are context, positions $T_c, T_c + 1, T_c + 2$ carry markers, and position $T - 1$ is the query. The sample set is

$$S := \big\{(\tau, \ell_0, k, i, u) \;:\; \tau \in [n],\ \ell_0 \in \Lambda,\ k \in \{\ell_0 + 1, \ldots, L\},\ i \in [B],\ u \in [m]\big\}, \tag{17}$$

so that

$$N := |S| \;=\; n\,m\,B \sum_{\ell_0 \in \Lambda} (L - \ell_0). \tag{18}$$

Key space. Σ contains: two context tokens $\mathrm{tok}_0, \mathrm{tok}_1$; n group tokens τ; $|\Lambda|$ start markers; L readout markers; and m channel markers. Thus $K = n + |\Lambda| + L + m + 2$, and $K \ge C_0(L + p)$ permits $n = \Theta(K)$. The key map κ is the identity on token ids; any injective κ works, which is why the theorem applies verbatim to n-gram or hashed keys as long as the n group keys are distinct.

Fixed input embedding. The positional encoding supplies three coordinates $(\textsc{one}, \textsc{t}, \textsc{tsq}) = (1, t, -t^2)$. The fixed token embedding TE_0 supplies: $\eta = 1$ and $\textsc{bit} = b$ for tok_b; $\textsc{qi} = 1$ for group tokens; $(\textsc{mu}, \textsc{muv}) = (1, \ell_0)$, $(\textsc{nu}, \textsc{nuv}) = (1, k)$ and $(\textsc{pi}, \textsc{piv}) = (1, u)$ for the three marker families. All other coordinates are 0. Note $\eta = 0$ at markers and at the query, which will exclude them from retrieval.

Sample encoding. For (τ, ℓ_0, k, i, u): context position t carries $\mathrm{tok}_{\mathrm{bit}_i(t)}$; positions T_c, T_c+1, T_c+2 carry the start marker for ℓ_0, the readout marker for k and the channel marker for u; position $T-1$ carries τ. The label is $Y_{\tau,\ell_0,k,i,u} \in \{0,1\}$.

Hidden-state layout. The hidden state reserves the twelve input coordinates above, never overwritten and carried by the residual path, four work scalars $\mathrm{G}_0, \mathrm{G}, \mathrm{GU}, \mathrm{ACC}$, and, for each $u \in [m]$, three work scalars R_u (address register), B_u (retrieved bit) and S_u (retrieved position). So $d = 16 + 3m$, a constant given p.

C.2 Parameter encoding

Fix a labelling $Y \in \{0,1\}^S$. For each (τ, ℓ_0, u, k) set

$$s_{\tau,\ell_0,u,k} \;:=\; \sum_{i=1}^{B} Y_{\tau,\ell_0,k,i,u}\, 2^{\,i-1} \;\in\; \{0,\dots,2^B-1\} \subseteq \{0,\dots,T_c-1\}, \tag{19}$$

and pack the $L-\ell_0$ addresses of the chain started at ℓ_0 into one real,

$$a_{\tau,\ell_0,u} \;:=\; \sum_{k=\ell_0+1}^{L} \frac{s_{\tau,\ell_0,u,k}}{\beta^{\,k-\ell_0}} \;\in\; [0, \tfrac{1}{2}). \tag{20}$$

The table is set to $E^{(\ell_0)}[\tau]_{3u+j} = s^{-1} a_{\tau,\ell_0,u}$ for $j \in \{0,1,2\}$, $\ell_0 \in \Lambda$, $\tau \in [n]$, and $E^{(\ell)}[x] = 0$ for every non-group token x and every $\ell \notin \Lambda$. We take $P^{(\ell)}_{\mathrm{proj}} = 0$, so by (2) $z^{(\ell_0)}$ equals $a_{\tau,\ell_0,u}$ in each of the three replicated coordinates at the query position and vanishes everywhere else.

Two observations. First, the three replicated coordinates are what let a *coordinatewise* product implement a linear combination of three ReLUs of a common scalar; this is the only place $p \geq 3$ is used. Second, distinct triples (τ, ℓ_0, u) occupy disjoint table entries and each entry encodes its $B(L-\ell_0)$ label bits injectively by (19)–(20); hence all N labels are independently settable.

Remark 25 (Where the extra factor of L comes from)**.** A construction using a single slice, $|\Lambda| = 1$, gives $\Omega(Kp \log T)$: the count N in (18) would then be a sum of one term instead of $|\Lambda|$. The additional factor of (up to) L comes entirely from using every slice $\ell_0 \in \Lambda$ as an independent chain origin: slice ℓ_0 of row τ funds a chain of length $L-\ell_0$, contributing $B(L-\ell_0)$ independently settable bits by the observation above, and these contributions sum over $\ell_0 \in \Lambda$ because distinct (τ, ℓ_0, u) occupy disjoint table entries. This is the mechanism that makes *remaining depth*, rather than raw parameter count, the operative quantity in Ω_Λ.

C.3 Layer 0: gathering the three marker scalars

Layer 0 uses three attention heads. Head 1 has $Q_i = (M)$ read from ONE and $K_t = (\mathrm{MU}_t)$, so the score is $M \cdot \mathrm{MU}_t$ with $M > 0$; exactly one position, the start marker, has $\mathrm{MU} = 1$ and all others 0, so the argmax is unique. Its value projection reads MUV and the output projection writes it to G_0. Heads 2 and 3 do the same for $(\mathrm{NU}, \mathrm{NUV}) \to \mathrm{G}$ and $(\mathrm{PI}, \mathrm{PIV}) \to \mathrm{GU}$. Since the query is the last position it attends to all three markers. The layer-0 FFN is zero.

At positions that do not causally see a marker the head's scores are all 0, so attention averages values that are identically 0; the gathered coordinates are 0 there, which is harmless because only the query position is read out.

Lemma 26. *After layer* 0*, the query position satisfies* $(\mathrm{G}_0, \mathrm{G}, \mathrm{GU}) = (\ell_0, k, u)$ *for the sample's indices.*

C.4 The injection as a start-layer-gated address load

For $\ell \in \Lambda$ set, for each $u \in [m]$ and $j \in \{0,1,2\}$ with offsets $o_j \in \{-\frac{1}{2}, 0, +\frac{1}{2}\}$,

$$\big(P^{(\ell)}_{\mathrm{in}} h\big)_{3u+j} \;=\; \mathrm{G}_0 - (\ell + o_j), \qquad \big(P^{(\ell)}_{\mathrm{out}}\big)_{\mathrm{R}_u,\,3u+j} \;=\; c_j \cdot 2T_c, \quad (c_0,c_1,c_2) = (2,-4,2), \tag{21}$$

the bias being available through the ONE coordinate.

Table 4: The chain traced for $T_c = 8$ ($\beta = 16$, $B = 3$), $L = 3$, $\Lambda = \{0\}$ and addresses $(5, 2, 7)$, i.e. the single table entry $a = 0.322021484375$. Each register is β times the previous remainder and its nearest integer is the next address exactly (`code/worked_example.py`).

layer κ	register r	$t^* = \mathrm{round}(r)$	address s_κ	bits of t^*
1	5.152343750	5	5	$(1, 0, 1)$
2	2.437500000	2	2	$(0, 1, 0)$
3	7.000000000	7	7	$(1, 1, 1)$

Lemma 27 (Hat gate). *For integers g, ℓ, $2\big(\mathrm{ReLU}(g-\ell+\frac{1}{2}) - 2\,\mathrm{ReLU}(g-\ell) + \mathrm{ReLU}(g-\ell-\frac{1}{2})\big) = \mathbf{1}[g = \ell]$.*

Proof. If $g \le \ell - 1$ all three arguments are $\le -\frac{1}{2} < 0$ and the value is 0. If $g = \ell$ the value is $2(\frac{1}{2} - 0 + 0) = 1$. If $g = \ell + 1$ it is $2(\frac{3}{2} - 2 + \frac{1}{2}) = 0$, and for $g \ge \ell + 1$ all three ReLUs are active and $2((g - \ell + \frac{1}{2}) - 2(g - \ell) + (g - \ell - \frac{1}{2})) = 0$. □

Lemma 28 (Address load). *At the query position, the layer-ℓ injection adds $\mathbf{1}[\ell_0 = \ell] \cdot 2T_c\, a_{\tau,\ell,u}$ to R_u and nothing else; at every other position it adds nothing.*

Proof. By construction $z^{(\ell)}_{3u+j} = a$ for all three j at the query position, where $a = a_{\tau,\ell,u}$. Hence $\sum_j c_j\, 2T_c\, \mathrm{ReLU}(\mathrm{G}_0 - \ell - o_j)\, z_{3u+j} = 2T_c\, a \sum_j c_j \mathrm{ReLU}(\mathrm{G}_0 - \ell - o_j) = 2T_c\, a \cdot \mathbf{1}[\mathrm{G}_0 = \ell]$ by Lemma 27 and Lemma 26. At non-query positions the token is not a group token, so $E^{(\ell)}[\kappa(x_i)] = 0$ and $z^{(\ell)}_i = 0$. □

Thus the register is loaded exactly once, at the layer named by the sample's start marker, and the loaded value is $2T_c\, a = s_{\tau,\ell_0,u,\ell_0+1} + \delta$ with $0 \le \delta = \sum_{k \ge \ell_0+2} s_{\tau,\ell_0,u,k} \beta^{-(k-\ell_0-1)} \le (T_c - 1)/(\beta - 1) < \frac{1}{2}$.

C.5 Retrieval layers

Attention. At layer $\kappa \in \{1, \ldots, L\}$, head u takes $Q_i = (2\,\mathrm{R}^{(i)}_u,\ 1,\ M)$ and $K_t = (\mathrm{T}_t,\ \mathrm{TSQ}_t,\ \eta_t)$, so $\langle Q_i, K_t \rangle = 2r\,t - t^2 + M\eta_t$ with $r := \mathrm{R}^{(i)}_u$.

Lemma 29 (Nearest-integer retrieval). *Let $r = s + \delta$ with $s \in \{0, \ldots, T_c - 1\}$ and $|\delta| < \frac{1}{2}$, and let $M \ge 1$. Then the unique maximiser over causally visible positions is the context position $t^* = s$.*

Proof. Write the score as $r^2 - (t - r)^2 + M\eta_t$. Over context positions ($\eta = 1$) it is maximised at the integer nearest r, which is s uniquely since $|\delta| < \frac{1}{2}$, and its value is $r^2 - \delta^2 + M \ge r^2 - \frac{1}{4} + M$. Non-context positions are the three markers and the query, all at $t \ge T_c$, and carry $\eta = 0$; since $r \le T_c - \frac{1}{2}$ they score $r^2 - (t - r)^2 \le r^2 - \frac{1}{4}$, which is smaller by at least $M \ge 1$. Uniqueness makes average-hard attention deterministic. The condition is far weaker than the $M > T_c^2$ one might expect, because the r^2 term is common to both sides; the reference implementation uses $M = 64$ at every T_c. □

The value projection reads $(\mathrm{BIT}, \mathrm{T})$ of t^*, and the output projection writes them to $(\mathrm{B}_u, \mathrm{S}_u)$.

FFN. The layer-κ FFN has two internal ReLU layers. The first computes, for each u: $p_u = \mathrm{ReLU}(\mathrm{R}_u - \mathrm{S}_u)$, exact since $\mathrm{R}_u - \mathrm{S}_u = \delta \ge 0$ by Lemma 29; the pass-throughs $\mathrm{ReLU}(\mathrm{R}_u), \mathrm{ReLU}(\mathrm{B}_u), \mathrm{ReLU}(\mathrm{S}_u)$, exact since all are nonnegative; the three hat units for G versus κ; and the three hat units for GU versus u. The second computes $2T_c\, p_u$, the pass-throughs again, and

$$\mathrm{acc}_u \;=\; \mathrm{ReLU}\big(\mathrm{B}_u + \mathbf{1}[\mathrm{G} = \kappa] + \mathbf{1}[\mathrm{GU} = u] - 2\big), \tag{22}$$

each indicator being the linear combination of Lemma 27 of first-layer units. The output affine map adds $2T_c p_u - \mathrm{R}_u$ to R_u, so the register becomes $2T_c\delta$, subtracts B_u and S_u to clear the scratch for the next layer, and adds $\sum_u \mathrm{acc}_u$ to ACC. Since $D_0 \ge 3$ affine maps are allowed, this is a legal FFN.

Lemma 30 (Chain invariant). *Fix a sample (τ, ℓ_0, k, i, u) and write $s_\kappa := s_{\tau,\ell_0,u,\kappa}$. Then at the query position, for every $\kappa \in \{\ell_0+1, \dots, L\}$, the register entering layer κ equals $\sum_{m'=\kappa}^{L} s_{m'}\beta^{-(m'-\kappa)}$, and layer κ retrieves $t^* = s_\kappa$ and $\mathrm{B}_u = \mathrm{bit}_i(s_\kappa)$. For $\kappa \le \ell_0$ the register is 0, the retrieval returns $t^* = 0$ and $\mathrm{B}_u = \mathrm{bit}_i(0)$, and the register remains 0.*

Proof. Induction on κ. For $\kappa \le \ell_0$: the register starts at 0, since Lemma 28 fires only at $\ell = \ell_0$, Lemma 29 with $r = 0$ gives $t^* = 0$, and the shift yields $2T_c(0-0) = 0$. At $\kappa = \ell_0 + 1$ the register is $2T_c a = \sum_{m' \ge \ell_0+1} s_{m'}\beta^{-(m'-\ell_0-1)}$ by Lemma 28 and (20), which is the claimed form. Given the form at κ, write $r = s_\kappa + \delta_\kappa$ with $\delta_\kappa = \sum_{m'>\kappa} s_{m'}\beta^{-(m'-\kappa)} \in [0, \frac{1}{2})$; by Lemma 29, $t^* = s_\kappa$, and the shift gives $2T_c\delta_\kappa = \sum_{m' \ge \kappa+1} s_{m'}\beta^{-(m'-\kappa-1)}$, the form at $\kappa+1$. □

Lemma 31 (Read-out). *With $w_{\mathrm{out}} = e_{\mathrm{ACC}} - \frac{1}{2}e_{\mathrm{ONE}}$, the classifier outputs $Y_{\tau,\ell_0,k,i,u}$ on the sample (τ, ℓ_0, k, i, u).*

Proof. By (22) and $\mathrm{B}_{u'} \in \{0, 1\}$, the update contributes $\mathrm{B}_{u'}$ when $\kappa = \mathrm{G} = k$ and $u' = \mathrm{GU} = u$, and 0 otherwise: if either indicator is 0 the argument is $\le -1 + 1 = 0$ or $\le 0 - 1 < 0$. Since $k \ge \ell_0 + 1$, Lemma 30 gives $\mathrm{B}_u = \mathrm{bit}_i(s_{\tau,\ell_0,u,k})$ at layer k, which by (19) equals $Y_{\tau,\ell_0,k,i,u}$. Hence $\mathrm{ACC} = Y_{\tau,\ell_0,k,i,u} \in \{0, 1\}$ at the end, and the logit $\mathrm{ACC} - \frac{1}{2}$ has the sign of the label. □

C.6 Parameter budget and conclusion

Proof of Theorem 8. By Lemma 31 the constructed θ realises an arbitrary $Y \in \{0, 1\}^S$, so S is shattered and $\mathrm{VCdim} \ge N$ with N as in (18).

The trainable parameters split as follows. The table contributes $K|\Lambda|p$; the projections $P_{\mathrm{in}}, P_{\mathrm{out}}$ contribute $O(|\Lambda|\, d\, p) = O(|\Lambda|m^2)$ with $d = 16 + 3m$, and $P_{\mathrm{proj}} = 0$. The backbone consists of three gather heads with $O(1)$ nonzero slots each; m retrieval heads per layer with $O(1)$ slots each, i.e. $O(mL)$; per-layer FFNs with $O(m)$ units and $O(m)$ nonzero weights each, i.e. $O(mL)$; plus w_{out}. Hence $W_{\mathrm{bb}} = O(pL)$, as claimed, and the total is $W = \Theta(K|\Lambda|p) + O(pL + |\Lambda|p^2)$. Under $K \ge C_0(L + p)$ the table term dominates and $n = \Theta(K)$.

Substituting $n = \Theta(K)$, $m = \Theta(p)$ and $B = \Theta(\log T)$ into (18),

$$N \;=\; \Theta\Big(K\, p\, \log T \sum_{\ell_0 \in \Lambda} (L - \ell_0)\Big) \;=\; \Omega(\Omega_\Lambda \log T), \tag{23}$$

using (5) and $\sum_{\ell\in\Lambda}(L - \ell + 1) \le 2\sum_{\ell\in\Lambda}(L - \ell)$ for $\Lambda \subseteq \{0, \dots, L-1\}$, $L \ge 2$. The depth is $L + 1 = O(L)$ and the context length is $T = T_c + 4$, so $B = \Theta(\log T)$.

Finally, with $\Lambda = [L]$ we get $\sum_\ell (L - \ell) = \Theta(L^2)$, hence $N = \Theta(KpL^2 \log T)$; since $W_{\mathrm{lut}} = \Theta(KpL)$ this is $\Theta(W_{\mathrm{lut}} L \log T)$, and choosing $K = \Theta(W/(pL))$ makes $W_{\mathrm{lut}} = \Theta(W)$ and $N = \Theta(WL \log T)$. □

Proof of Corollary 14. $\Omega_\Lambda = \Theta(Kp\sum_{\ell\in\Lambda}(L - \ell + 1))$ and $W_{\mathrm{lut}} = \Theta(K\lambda p)$ with $\lambda = |\Lambda|$. Over subsets of size λ, the sum $\sum_{\ell\in\Lambda}(L-\ell+1)$ is maximised at $\Lambda = \{1, \dots, \lambda\}$, giving $\lambda L - \lambda(\lambda-1)/2 = \Theta(\lambda L)$ for $\lambda \le L/2$, and minimised at $\Lambda = \{L - \lambda + 1, \dots, L\}$, giving $\lambda(\lambda+1)/2 = \Theta(\lambda^2)$. Dividing by W_{lut} gives Ω_Λ equal to $\Theta(W_{\mathrm{lut}} L)$ and $\Theta(W_{\mathrm{lut}}\lambda)$ respectively, and the ratio is $\Theta(L/\lambda)$. Achievability in both directions is Theorem 8 applied to each Λ, and the upper direction is Theorem 6; within the window of Proposition 11 the two match up to constants. □

Proof of Proposition 15. An input-side table is live at layer 0, so every one of its W_0 parameters has remaining depth L and contributes $W_0 L$ to Φ by Definition 3. A GLI table of the same total size $W_{\mathrm{lut}} = W_0$ with $\Lambda = [L]$ has $Kp = W_{\mathrm{lut}}/L$ parameters first live at each layer, so $\Omega_\Lambda = (W_{\mathrm{lut}}/L)\sum_{\ell=1}^{L}(L - \ell + 1) = W_{\mathrm{lut}}(L+1)/2$. The ratio is $2L/(L+1) = 2 - o(1)$. Restricting to the last λ layers replaces $(L+1)/2$ by $\Theta(\lambda)$ by Corollary 14. □

C.7 Routing unlocks the table

Proposition 32 (Support cap without attention)**.** *Remove all attention sublayers. By Remark 2 the read-out is a function of* $(x_{T-1}, T-1)$ *alone, so on any sample set whose query positions use* k *distinct tokens at most* 2^k *labellings are realisable, and* $\mathrm{VCdim} \le K$ *regardless of* L*,* p *or* W_{lut}*.*

The gap between $W_{\mathrm{lut}}/(Lp)$ and $\Omega_\Lambda \log T$ is thus attributable to the retrieval backbone: lookup parameters are powerful because attention can address them. This also explains why the effective-support refinement fails. Replacing K by $\min(K, NT)$ in Ω_Λ is legitimate, since unread rows appear in no branching polynomial, but the resulting fixed point is vacuous (Remark 21).

Proof of Proposition 32. Remove all attention sublayers. Then every sublayer, the FFN and the injection (3), acts position-wise, and by (2) $z_i^{(\ell)}$ is a function of (x_i, i) alone. By induction on ℓ, $h_i^{(\ell)}$ is a function of (x_i, i) alone. The classifier reads $h_{T-1}^{(L)}$, so $f_\theta^{01}(X) = \phi_\theta(x_{T-1}, T-1)$ for some ϕ_θ. On a sample set whose query positions use k distinct tokens the map $X \mapsto f_\theta^{01}(X)$ is determined by the k values $\phi_\theta(\cdot, T-1)$, so at most 2^k labellings arise; in particular no set with two inputs sharing a query token is shattered, and $\mathrm{VCdim} \le K$ irrespective of L, p, W_{lut}. □

C.8 Proof of Theorem 10

Proof. Fix b and let $B_b = \lfloor (b - c_1)/\log_2(2T_c) \rfloor$. Modify the labelling map (19)–(20) by truncating each chain: put $s_{\tau,\ell_0,u,k} := 0$ for every $k > \ell_0 + B_b$, leaving $\min\{L - \ell_0, B_b\}$ addresses per triple. Then $a_{\tau,\ell_0,u}$ is a multiple of $\beta^{-\min\{L-\ell_0, B_b\}}$ with $\beta = 2T_c$, so it is exactly representable in b bits once c_1 absorbs the two leading bits of $2T_c a$. We also fix the top address bit to 0, i.e. use only $i \in [B-1]$, so that every address lies in $\{0, \dots, T_c/2 - 1\}$ and $\delta_\kappa \le (T_c/2 - 1)/(\beta - 1) < \frac{1}{4}$: the retrieval of Lemma 29 then tolerates any register error below $\frac{1}{4}$. This costs a factor $(B-1)/B \ge \frac{1}{2}$ in the sample count.

For the chain invariant, write ε_κ for the absolute error in the register entering layer κ. Lemma 30 shifts by β at each step, and each step adds a fresh round-off of at most $C_0 2^{-b} T_c$ for an absolute C_0, because the shift-and-clear FFN performs $O(1)$ operations on values of magnitude $O(T_c)$. Hence $\varepsilon_{\kappa+1} \le \beta \varepsilon_\kappa + C_0 2^{-b} T_c$, the load of Lemma 28 contributing an initial error of the same size, and after j steps $\varepsilon \le C_0 2^{-b} T_c \sum_{i<j} \beta^i \le C_0 2^{-b} \beta^j T_c/(\beta - 1) \le C_0\, 2^{-b} \beta^j$, which is at most $\frac{1}{4}$ once $j \log_2 \beta \le b - \log_2(4C_0)$. So $c_1 = \max\{\log_2(4C_0), 2\}$ is an absolute constant. (Charging the round-off to the register before the shift, i.e. bounding by $2\beta^j 2^{-b} T_c$, would instead force $c_1 \gtrsim \log_2 T_c$.) Hence $|\delta_\kappa + \varepsilon| < \frac{1}{2}$ throughout and Lemma 29 still selects $t^* = s_\kappa$ exactly; every other step of Section C is exact integer or half-integer arithmetic and is unaffected. The truncated sample set $S_b := \{(\tau, \ell_0, k, i, u) \in S : k \le \ell_0 + B_b,\ i \le B - 1\}$ is therefore shattered, and $|S_b| = n\, m\, (B-1) \sum_{\ell_0 \in \Lambda} \min\{L - \ell_0, B_b\} = \Theta\big(Kp \log T \sum_{\ell \in \Lambda} \min\{L - \ell, B_b\}\big)$. □

Proof of Proposition 12. With b-bit weights the parameter space has at most 2^{Wb} elements, so $\mathcal{F}_{\mathcal{A}}^{01}$ contains at most 2^{Wb} distinct functions. A shattered set of size N forces $2^N \le 2^{Wb}$, hence $N \le Wb$. Combining with Theorem 6 gives the stated minimum, and the two terms cross at $b = \Theta((\Phi/W)\log(TL\Phi)) = \Theta(\bar{L} \log(TL\Phi))$. □

Corollary 33 (Saturating allocation)**.** *In* b*-bit arithmetic the ratio of Corollary 14 attained by the witness is* $\Theta(\min\{L, B_b\}/\min\{\lambda, B_b\})$*, equal to* 1 *once* $\lambda \ge B_b$*.*

Proof of Corollary 33. Substitute $\min\{L - \ell, B_b\}$ for $(L - \ell)$ in the computation of Corollary 14. Earliest placement gives $\Theta(\lambda \min\{L, B_b\})$ and latest gives $\Theta(\lambda \min\{\lambda, B_b\})$ up to constants, whence the ratio; for $\lambda \ge B_b$ both are $\Theta(\lambda B_b)$. □

Remark 34 (Bit complexity, revisited)**.** By (20) the register must resolve $\beta^{-(L-\ell_0)}$ relative to an $O(1)$ leading term, i.e. $\rho = (L - \min \Lambda) \log_2 \beta$ bits. Each of the $L - \ell_0$ shifts multiplies the representation error of a by β, so the true float demand is $\rho + \log_2 \beta + O(1)$; the crude count $\rho \le 53$ is therefore accurate except within a few bits of the boundary, which is what Section 7.3 observes. The same $\Theta(L \log T)$ requirement is present in the construction of Yang et al. (2026): it is inherent to packing L addresses of $\log T$ bits into one real, and is the reason the setting posits exact arithmetic over $\mathbb{R}$.

Table 5: Relation to prior parametric bounds. $\Phi = \bar{L}W$ is the same quantity in the first two rows; the contribution of Theorem 6 is that it survives hard attention, and of Theorem 8 that it is attained by parameters sitting in a lookup table.

	Architecture	Upper bound	Lower bound
Bartlett et al. (2019)	piecewise-linear feed-forward	$O(\Phi \log W)$	$\Omega(WL\log(W/L))$
Yang et al. (2026)	hard-attention Transformer	$O(WL\log(TW))$	$\Omega(WL\log(TW/L))$
Theorem 6, Theorem 8	+ gated lookup injection	$O(\Phi\log(TL\Phi))$	$\Omega(\Omega_\Lambda \log T)$

Table 6: Direct verification of Theorem 8: three independent random labellings of the full sample set per row, every sample checked. W_{bb} counts non-lookup trainable slots actually used; Ω_Λ is (5) restricted to the table term. All rows satisfy $\rho \le 53$.

L	T	$\lvert\Lambda\rvert$	p	K	N	W_{lut}	W_{bb}	Ω_Λ	mismatches
3	12	3	3	11	36	99	140	198	0/108
4	20	4	6	15	240	360	330	900	0/720
5	12	1	3	13	60	39	224	195	0/180
5	12	1	3	13	12	39	224	39	0/36
6	36	3	9	16	360	432	710	1728	0/1080
7	20	7	6	21	672	882	567	3528	0/2016
9	20	3	3	17	120	153	392	765	0/360
10	20	2	6	18	304	216	804	2052	0/912
10	20	2	6	18	48	216	804	324	0/144
16	12	8	3	29	216	696	686	3132	0/648
total									**0/6204**

D Per-Configuration Experimental Results

D.1 The witness under injection-path normalisation

Proposition 35 (The witness survives normalisation)**.** *Theorem 8 holds verbatim with p replaced by $p+1$ when the injection path is RMS-normalised as in Proposition 22.*

Proof. Normalisation fixes the root-mean-square of $z^{(\ell)}$ at 1, so a packed address can no longer be stored as a magnitude: the construction of the parameter encoding above would recover only the *direction* of the table row. Devote one further coordinate per slice to a ballast value and set, for each (τ,ℓ) with $\ell \in \Lambda$,

$$\beta_{\tau,\ell} := \sqrt{(p+1) - \textstyle\sum_{u\in[m]} 3\,a_{\tau,\ell,u}^2}, \tag{24}$$

which is real because $a_{\tau,\ell,u} < \frac{1}{2}$ by (20) and hence $\sum_u 3a^2 < 3m/4 < p+1$. The slice then has $\sum_j (z^{(\ell)})_j^2 = (p+1)$, so its root mean square over the $p+1$ coordinates is exactly 1 and $\tilde{z}^{(\ell)}_{3u+j} = a_{\tau,\ell,u}$ for $j \in \{0,1,2\}$: the post-normalisation address coordinates are the intended ones. Rows that are identically zero normalise to zero, as before. Every subsequent step of Section C reads only those coordinates and is unchanged. □

The cost is one coordinate, not a factor, so the rate is untouched: the normalisation deployed implementations apply does not make lookup parameters statistically cheaper. Section G executes the normalised witness.

E The Residual-Backbone Control

The study referred to in Section 7.4 uses a differentiable gated-lookup residual network, $h \leftarrow h + \gamma V_\ell \mathrm{ReLU}(A_\ell h + c_\ell)$ with an input-keyed injection $h \leftarrow h + G_\ell(\mathrm{ReLU}(P_\ell h) \odot Q_\ell x)$ at $\ell \in \Lambda$, everything frozen except the slices Q_ℓ, fitted with Adam and hand-derived gradients checked against central finite differences to a maximum relative error of 8×10^{-10}. The metric is the memorisation

Table 7: Broad sweep, grouped by Λ pattern. The declared grid is $L \in \{2, \ldots, 12\}$, $T_c \in \{8, 16\}$, $n \in \{1, 2\}$ group keys, $p \in \{3, 6\}$, and six Λ patterns, filtered to $\rho \leq 53$ and $|S| \leq 420$; two random labellings per configuration.

Λ pattern	configurations	labels	mismatches
single-early $\{0\}$	80	8,046	0
single-late $\{L-1\}$	88	1,386	0
front half	59	18,130	0
back half	72	11,214	0
every other	69	16,752	0
all layers	67	19,770	5
total	**435**	**75,298**	**5**

Table 8: Exact-arithmetic re-run of *exactly* the configurations that fail in float64: the six failing cells of the precision grid and the one failing sweep configuration. Same seeds and same number of labellings in both columns.

source	L	T_c	$\lvert\Lambda\rvert$	ρ (bits)	float64	exact `Decimal`
precision grid	8	64	1	56	6/48	0/48
precision grid	10	32	1	60	11/50	0/50
precision grid	10	64	1	70	12/60	0/60
precision grid	12	16	1	60	9/48	0/48
precision grid	12	32	1	72	15/60	0/60
precision grid	12	64	1	84	14/72	0/72
broad sweep	10	16	10	50	0/440	0/440
total					**67/778**	**0/778**

threshold N^*, the largest sample size on a geometric ladder fitted perfectly, averaged over per-seed thresholds; being defined by where fitting fails it has dynamic range by construction, unlike a mean training accuracy. Per-placement values are in Table 9. The first-order control computes the Jacobian of the read-out with respect to the slice parameters at initialisation and reports its numerical rank, which is 60, equal to the slice parameter count, at every placement.

F GPU Results in Full

Tables 11 and 12 give the witness-at-scale and precision-law runs of Section 7.5; Section F.1 gives the controlled placement design, Figure 5 its two controls, and Section F.2 how to read it. Raw rows are in `results_gpu/*.csv` of the supplement.

F.1 The controlled placement design in full

Table 14 lists all thirteen memorisation arms behind Table 2 and Figure 1b, Table 15 the attention ablation, and Table 16 the same design on the language-modelling objective.

F.2 Calibrating the controlled placement result

Four things belong beside Table 2, none of which fits in the main text.

The earlier coarse sweep was under-resolved. A preliminary run with three seeds on a factor-2 ladder gave $N^* = 213.3$ for *mid* against 128.0 for *spread* at identical budget, which read as evidence against Φ. At five seeds on the $\sqrt{2}$ ladder the same two placements give 211.0 and 226.0, a difference of 7%, so we report only the five-seed design. In that preliminary run the state-keyed arm fitted the top rung ($N^* = 256$) at every placement and the all-parameters arm was right-censored; both are uninformative and are not reported.

Table 9: Realised memorisation threshold N^* (mean over six seeds) against the depth-weighted budget of the slice block, at fixed slice parameter count 60, for three frozen-backbone residual scales γ. See Section 7.4. The correlation with Φ is negative in all three regimes.

Λ Φ of slice block	$\{1,2\}$ 450	$\{2,3\}$ 390	$\{3,4\}$ 330	$\{4,5\}$ 270	$\{5,6\}$ 210	$\{6,7\}$ 150	$\{7,8\}$ 90	r
$N^*, \gamma = 1$	2.7	5.7	11.0	20.0	22.5	26.3	55.8	-0.92
$N^*, \gamma = 0.3$	8.0	13.8	20.2	14.7	23.3	28.2	36.0	-0.94
$N^*, \gamma = 0.05$	35.0	33.7	40.3	18.7	12.8	42.2	55.2	-0.27
Jacobian numerical rank	60	60	60	60	60	60	60	—

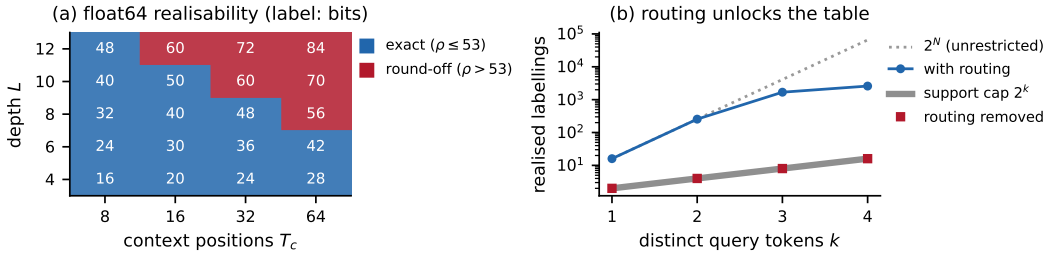


Figure 3: **(a)** Float64 realisability of the witness; each cell is labelled with its required precision ρ in bits. **(b)** Realisable labellings versus token support. With routing removed the class sits *exactly* on the support cap 2^k (the two series coincide) regardless of W_{lut}; with routing the same architecture escapes it.

The noise floor is estimated twice, and the two estimates disagree. The $\{0, 1, 2\}$ replicate returns identical per-seed vectors, a difference of 0%; the $\{0, 1, 11\}$ replicate returns 181.0 against 211.0, a difference of 17%. Both pairs ran in the same session, so session drift does not account for it. We therefore quote 17% as the run-to-run noise, being the conservative of the two, and read the group-A spread of 33% against it: roughly a factor of two above noise, which is why we do not treat group A as a null and rest the separation on the partial correlations instead.

Saturation does not explain group A. Standard *early* (budget 33) and the largest group-A arm (budget 24) both return 241.0, so the top of the range is compressed and a ceiling of the harness near $N^* = 256$ would be an alternative explanation of group A's small spread. Group C rules it out: widening $|\Lambda|$ to push the budget to 42 and 50 gives 298.4 and 362.0, the latter with all five seeds at the same rung. The compression is a property of the budget response, not a wall.

The language-modelling comparison has limited resolution. Φ there is dominated by the backbone, identical across arms ($\Phi_{\text{bb}} \approx 6.9 \times 10^7$ against slice terms of 1.1–5.9×10^6), so Φ_{tot} moves 6.9% across the placement arms while the gap moves 67.5%. The sweep is an ordinal test at fixed backbone and cannot be read quantitatively. It also places the cross-mechanism rows of Table 13 — an input-side table, a state-keyed gate, and no table at all — far inside the resolution of the comparison, which is why they do not order with Φ and why we exclude them from every reported correlation rather than treating them as evidence either way.

The response is sublinear. Regressing $\log N^*$ on $\log$ budget over all thirteen arms gives a slope of 0.509; group B alone gives 0.536. The ceiling of Theorem 6 is linear in Φ, so the realised threshold grows roughly as $\sqrt{\Phi}$. We have no account of the exponent. It is the sharpest question our data raise about realised capacity, and we record it as open rather than fitting a story to it.

G Normalisation and the input-side baseline: experiments

Three checks, one per claim of Section B and Proposition 35. Scripts are `exp8_znorm.py` and `exp9_scope.py`.

Table 10: Distinct labellings realised over 4000 random parameter draws ($L = 4$, $d = 8$, $K = 6$, $p = 3$, $T = 7$, four contexts per token).

k	N	routing removed	with routing	support cap 2^k	unrestricted 2^N
1	4	2	16	2	16
2	8	4	255	4	256
3	12	8	1685	8	4096
4	16	16	2590	16	65536

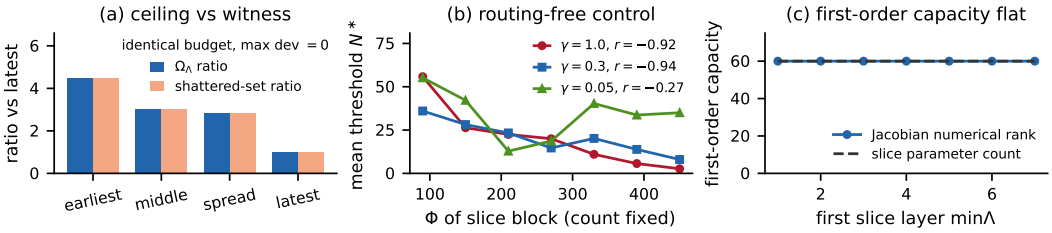


Figure 4: **(a)** At identical budget, shattered-set ratios match Ω_Λ ratios exactly. **(b)** Realised memorisation threshold against slice-block Φ at fixed parameter count, three residual scales γ; the relationship is negative. **(c)** The slice Jacobian has full numerical rank at every placement, so (b) is not a first-order effect.

Counting the radicals (Proposition 22, Remark 23). The claim separating the two normalisations is a structural count, so we count rather than estimate: we enumerate every normaliser value arising in a forward pass and count the distinct ones, as N and T grow. Table 17 gives the result. The injection path produces 12 distinct radicals at every setting, exactly the bound $K|\Lambda|$ with $K = 6$ and $|\Lambda| = 2$, while N grows $8\times$ and T grows $4\times$. The residual stream produces NTL of them exactly. The first count is what Proposition 22 absorbs into Φ; the second is what makes Remark 23 vacuous.

The normalised witness (Proposition 35). Adding the ballast coordinate and RMS-normalising the injection path, we re-run the verification of Section 7.3 across five architectures (L from 3 to 8, m from 1 to 3, Λ early, spread and single-slice), three independent random labellings each: 2,646 labels checked, **zero mismatches**. As elsewhere the harness has no tolerance and no subsampling, so one mismatch would falsify the proposition.

The rank cap (Proposition 24). Asking how well a random target $M \in \mathbb{R}^{K\times d}$ ($K = 64$, $d = 16$) is realisable as $E_0 A^\top$, the relative error falls monotonically with p' and reaches 2×10^{-15} — floating-point zero — exactly at $p' = d$, remaining there for every larger p'. Nominal parameters Kp' therefore exceed usable parameters $K \min(p', d)$ by the factor p'/d beyond the cap.

H Code and Reproduction

Table 18 maps each claim to the script that tests it, the output it writes, and where the number appears; Table 19 lists every source file in the supplement.

H.1 Commands

Python 3 and `numpy` are the only requirements for the CPU experiments; `matplotlib` is needed for the figures. Run from `code/` in the order of Table 18; `exp3c.py` must follow `exp3ab.py` and `exp3d_sweep.py`, whose float64 failures it re-runs. `exp3d_sweep.py --dry` confirms the "435 configurations, 75,298 labels" count in under a second. `make_figs.py`, `make_fig_body.py` and `make_figs_gpu.py` regenerate every figure from `code/results/*.json` and `code/results_gpu/*.csv`.

H.2 Files in the supplement

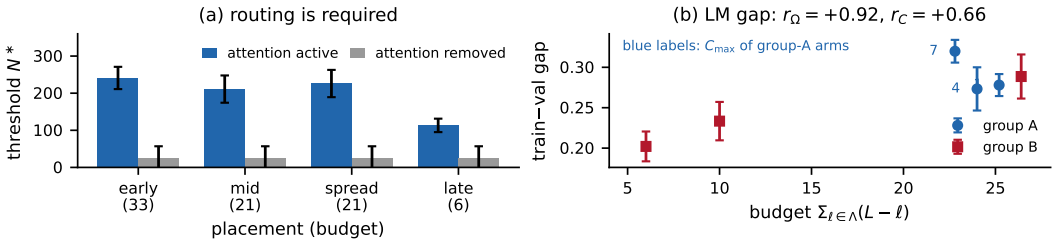


Figure 5: **(a)** Attention ablation in the frozen-backbone harness: with attention the threshold N^* orders with the slice budget; without it all four placements return the identical per-seed vector (Table 15). **(b)** Language-modelling version of the controlled design (Table 16): train–validation gap against the slice budget, groups A (fixed budget) and B (fixed $C_{\max} = 1$); error bars are one standard deviation over four seeds. Generated by `make_figs_gpu.py` from the CSVs in the supplement.

Table 11: GPU verification of Theorem 8 at scale (Section 7.5), one axis pushed at a time. N is the full shattered-set size; "checked" is the number of labels actually verified, a uniformly random subset when N is too large to enumerate. All in float64 on one A6000.

axis	L	T	K	p	$\lvert\Lambda\rvert$	ρ	N	checked	mism.	peak GB
key	8	20	1,043	3	8	40	147,456	20,000	0	0.17
key	8	20	4,115	3	8	40	589,824	20,000	0	0.17
key	8	20	16,403	3	8	40	2,359,296	20,000	0	0.18
key	8	20	65,555	3	8	40	9,437,184	20,000	0	0.19
key	8	20	131,091	3	8	40	18,874,368	20,000	0	0.22
ctx	7	68	273	3	7	49	43,008	20,000	0	1.49
ctx	6	132	271	3	6	48	37,632	20,000	0	5.28
ctx	5	260	269	3	5	45	30,720	20,000	0	6.37
ctx	5	516	269	3	5	50	34,560	20,000	0	6.27
ctx	4	1028	267	3	4	44	25,600	20,000	0	6.18
width	6	20	527	3	6	30	43,008	20,000	0	0.17
width	6	20	528	6	6	30	86,016	20,000	0	0.18
width	6	20	530	12	6	30	172,032	20,000	0	0.29
width	6	20	534	24	6	30	344,064	20,000	0	0.56
width	6	20	542	48	6	30	688,128	20,000	0	1.09
depth	8	12	525	3	2	8	4,608	4,608	0	0.08
depth	12	12	530	3	3	12	9,216	9,216	0	0.08
depth	16	12	535	3	4	16	15,360	15,360	0	0.09
depth	24	12	545	3	6	24	32,256	20,000	0	0.10
depth	32	12	555	3	8	32	55,296	20,000	0	0.11
placement	10	12	1,041	3	4	40	104,448	20,000	0	0.08
placement	10	12	1,041	3	4	16	30,720	20,000	0	0.08
placement	10	12	1,042	3	5	40	92,160	20,000	0	0.08
placement	10	12	1,047	3	10	40	168,960	20,000	0	0.08
total, 24 configurations								**449,184**	**0**	

Table 12: Longest chain the witness realises exactly at b mantissa bits (Theorem 10). Each cell is measured; the fitted slope is compared with the parameter-free prediction $1/\log_2(2T_c)$.

T_c	8	10	12	14	16	18	20	24	28	32	36	40	44	48	52	slope	pred.
8	2	3	3	4	4	4	5	6	7	8	9	10	11	12	14	0.2550	0.2500
16	1	2	2	3	3	3	4	5	5	6	7	8	9	9	10	0.1992	0.2000
32	1	2	2	2	2	3	3	4	4	5	6	6	7	8	9	0.1686	0.1667
64	1	1	1	2	2	2	3	3	4	4	5	5	6	7	7	0.1437	0.1429

Table 13: Train–validation gap at fixed table budget, final step, from an earlier four-placement run (two seeds) that is confounded by $C_{\max}$ and is superseded for the placement question by Table 16; it is kept for the cross-mechanism arms (Section F.2). Only the 2-gram rows at $n = 32768$ are in a regime where the model generalises (validation accuracy 0.11–0.12 against a chance level of 0.0005); elsewhere both tasks memorise and the gap is uninformative.

task	config	Φ slice	n	train	val	gap
ngram	early	5,947,392	32768	5.4152	5.7555	0.3404
	mid	3,784,704	32768	5.4141	5.7018	0.2877
	spread	3,784,704	32768	5.4244	5.6792	0.2548
	late	1,081,344	32768	5.4769	5.6801	0.2032
	input0	6,488,064	32768	5.4514	5.7060	0.2547
	statekey	5,947,392	32768	5.4286	5.7209	0.2923
	none	0	32768	5.4408	5.6717	0.2309
recall	early	5,947,392	32768	0.8704	7.8689	6.9986
	mid	3,784,704	32768	1.1410	7.8071	6.6662
	spread	3,784,704	32768	1.1922	7.7676	6.5754
	late	1,081,344	32768	2.9938	7.6473	4.6535
	input0	6,488,064	32768	0.9731	7.8546	6.8816
	statekey	5,947,392	32768	1.8945	7.7627	5.8682
	none	0	32768	1.2442	7.8099	6.5657

Table 14: All thirteen arms of the controlled placement experiment (Table 2), frozen backbone, table trainable only, five seeds, $\sqrt{2}$ ladder. *budget* is $\sum_{\ell\in\Lambda}(L-\ell)$ with layers indexed from 0, which is $\Omega_\Lambda/(Kp)$ of (5) after re-indexing; the table budget W_{lut} is constant within groups A and B and grows with $|\Lambda|$ in group C. The two $\{0,1,11\}$ rows and the two $\{0,1,2\}$ rows are the same placement measured twice.

group	Λ	budget	$C_{\max}$	N^*	per-seed
C	$\{0,1,2,3,4\}$	50	8	362.0	362, 362, 362, 362, 362
C	$\{0,1,2,3\}$	42	9	298.4	256, 362, 362, 256, 256
C	$\{0,1,2\}$	33	10	241.0	256, 256, 256, 181, 256
—	$\{0,1,2\}$ (early)	33	10	241.0	256, 256, 256, 181, 256
—	$\{4,5,6\}$ (mid)	21	6	211.0	256, 181, 181, 181, 256
—	$\{0,5,10\}$ (spread)	21	2	226.0	256, 256, 181, 181, 256
—	$\{9,10,11\}$ (late)	6	1	113.2	91, 128, 91, 128, 128
A	$\{3,4,5\}$	24	7	241.0	256, 256, 256, 181, 256
A	$\{0,4,8\}$	24	4	226.0	256, 256, 181, 181, 256
A	$\{0,1,11\}$	24	1	181.0	181, 181, 181, 181, 181
B	$\{0,1,11\}$	24	1	211.0	181, 256, 181, 181, 256
B	$\{5,10,11\}$	10	1	159.8	181, 128, 128, 181, 181
B	$\{9,10,11\}$	6	1	100.4	91, 128, 64, 91, 128

Table 15: Attention ablation inside a single harness: same model, same parameter count, attention not executed. Five seeds. The per-seed vectors are *identical* across all four placements.

placement	attention active	attention removed	per-seed (removed)
early	241.0	25.6	0, 0, 0, 64, 64
mid	211.0	25.6	0, 0, 0, 64, 64
spread	226.0	25.6	0, 0, 0, 64, 64
late	113.2	25.6	0, 0, 0, 64, 64

Table 16: The controlled design on the language-modelling objective, four seeds, $n = 32768$. Group A is *not* flat here: the largest-$C_{\max}$ arm exceeds the other two group-A arms by 16%, about four times the anchor noise of 3.7%. We report the discrepancy with the memorisation result of Table 14 rather than resolving it.

Λ	budget	$C_{\max}$	gap (mean $\pm$ s.d.)
$\{3, 4, 5\}$	24	7	0.3199 ± 0.0141
$\{0, 4, 8\}$	24	4	0.2732 ± 0.0268
$\{0, 1, 11\}$	24	1	0.2780 ± 0.0136
$\{0, 1, 11\}$	24	1	0.2886 ± 0.0273
$\{5, 10, 11\}$	10	1	0.2334 ± 0.0238
$\{9, 10, 11\}$	6	1	0.2021 ± 0.0184
$r(\text{gap}, \Omega_\Lambda) = +0.916$;	$r(\text{gap}, C_{\max}) = +0.659$;	anchor noise 3.7%	

Table 17: Distinct normaliser values under the two placements of RMSNorm ($L = 4$, $K = 6$, $p = 5$, $\Lambda = \{0, 2\}$).

N	T	injection path	bound $K\lvert\Lambda\rvert$	residual stream ($= NTL$)
2	4	12	12	32
4	4	12	12	64
8	4	12	12	128
8	8	12	12	256
8	16	12	12	512
16	16	12	12	1024

Table 18: Claim to script to output. All scripts are run from inside `code/`; shipped outputs are in `code/results/`. Runtimes are wall-clock on a single CPU core.

Claim	Script	Output	Reported in	Runtime
Lemma 4, Theorem 7	`exp1_degree.py`	`exp1_degree.json`	Section 7.1, Table 3	3 s
Proposition 5	`exp2_branching.py`	`exp2_branching.json`	Section 7.2, Figure 2b	175 s
Theorem 8, fixed grid	`exp3ab.py`	`exp3ab.json`	Table 6	7 s
Remark 9, precision grid	`exp3ab.py`	`exp3ab.json`	Figure 3a	(same run)
Theorem 8, broad sweep	`exp3d_sweep.py`	`exp3d_sweep.json`	Table 7	61 s
Remark 9, exact re-run	`exp3c.py`	`exp3c.json`	Table 8	56 s
Corollary 14, ceiling	`exp4_allocation.py`	`exp4_allocation.json`	Figure 4a	78 s
Section 7.4, realised	`exp4_allocation.py`	`exp4_allocation.json`	Figure 4bc, Table 9	(same run)
Proposition 32	`exp5_attention.py`	`exp5_attention.json`	Figure 3b, Table 10	58 s
Proposition 22, Proposition 35	`exp8_znorm.py`	`exp8_znorm.json`	Table 17, Section G	6 s
Proposition 24	`exp9_scope.py`	`exp9_scope.json`	Section G	1 s
Model class of Definition 1	`ple_tf.py`	—	—	—
Witness of Theorem 8	`construction.py`	—	—	—
All figures	`make_fig*.py`	`figs/*.pdf`	—	2 s

Table 19: Source files in the supplement (`code/`). Listings are omitted here for space; every number in the paper is produced by these files.

File	Content
`ple_tf.py`	reference model class of Definition 1
`construction.py`	closed-form witness of Theorem 8 (float64 and exact `Decimal`)
`worked_example.py`	chain trace of Table 4
`exp1_degree.py`	exact degree propagation (Lemma 4)
`exp2_branching.py`	branch-flip counting (Proposition 5)
`exp3_lib.py`, `exp3ab.py`	verification harness, fixed grid, precision grid
`exp3d_sweep.py`, `exp3c.py`	declared sweep; exact re-run of float64 failures
`exp4_allocation.py`	allocation ceiling, residual control, Jacobian rank
`exp5_attention.py`	support cap without attention (Proposition 32)
`exp6_orthogonal.py`	orthogonal placement design (CPU)
`exp8_znorm.py`, `exp9_scope.py`	normalisation and rank-cap checks
`make_figs*.py`, `make_gpu_tables.py`	figures and GPU tables from raw outputs
`gpu_code/`	torch witness, agreement test, G1–G4 GPU runs (Section F)