WHERE DOES THE RECURRENCE HELP? RETRIEVAL-DISTANCE ATTRIBUTION IN HYBRID SEQUENCE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Attention costs quadratic time in sequence length, and the two standard remedies fail differently. A sliding window bounds cost but cannot see past its edge – a positional limit. A compressed recurrence bounds cost but retains only a lossy summary of the prefix – a capacity limit. Hybrid architectures combine both, but evaluate with aggregate quality metrics that average the two regimes into a single number revealing neither.

We present HALA, a block that computes sliding-window attention and a low-rank gated recurrence in parallel, mixes them with a learned per-head gate, and carries no position-wise feed-forward sublayer. Our contribution is less the architecture than the measurement. We stratify tokens by how far back their antecedent lies and attribute each path's contribution directly. Against an ablation with the recurrence removed, HALA is statistically indistinguishable on antecedents inside the 64-token window and consistently better outside it, by 1.12, 0.30 and 0.90 perplexity across three domains – and this advantage grows monotonically with scale, from 0.97 at 26.7M non-embedding parameters to 3.72 at 74.0M. On associative recall the separation from pure recurrence is categorical: Mamba-2 sits at chance across a full learning-rate sweep where HALA reaches 0.590.

We are equally explicit about the limits. HALA does not lead aggregate perplexity in any domain: Transformer and Jamba are ahead throughout. Against Mamba-2 the comparison is a crossover with a legible mechanism – HALA leads on antecedents inside the window, Mamba-2 on those beyond it, and Mamba-2 carries $3.4\times$ the recurrent state per layer. Length-extrapolation stability is inherited from recurrent state rather than contributed, and all results sit at 11.9–74.0M non-embedding parameters.

1 INTRODUCTION

Attention costs quadratic time in sequence length, and the two standard remedies fail differently. A sliding window bounds cost but cannot see past its edge – a positional limit. A compressed recurrence bounds cost but retains only a lossy summary, and Arora et al. (2024a) show recall accuracy is bounded by state size – a capacity limit. Neither is a matter of degree.

Combining them is now common, at two granularities: Jamba, Griffin and Samba (Lieber et al., 2024; De et al., 2024; Ren et al., 2024) interleave whole layers across depth, while B'MOJO, Hymba and Native Hybrid Attention (Zancato et al., 2024; Dong et al., 2025; Du et al., 2025) compose both inside a layer. We take the second approach and do not claim it as novel.

HALA computes sliding-window attention and a low-rank compressed recurrence in parallel within every block, mixes them with a learned per-head gate, and carries no position-wise feed-forward sublayer. But the question motivating this work is not whether such a block can be built – it is whether the two paths actually divide labour the way the design assumes, and that turns out to be answerable directly.

Stratifying by retrieval distance separates the paths. We identify tokens whose continuation was determined by an earlier occurrence of the same bigram and bucket them by how far back that

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.

antecedent lies. Against an ablation with the recurrent path removed, the full model is indistinguishable inside the 64-token window – 3.24 vs 3.19 perplexity on mixed, with the ablation marginally ahead in two of three domains – and consistently better beyond it, by 1.12, 0.30 and 0.90 across the three corpora. The recurrent path contributes nothing the window already covers and something measurable outside it; aggregate perplexity, which mixes both regimes, shows neither. The effect strengthens with scale, growing monotonically from 0.97 at 26.7M non-embedding parameters to 3.72 at 74.0M.

On associative recall the gap to pure recurrence is categorical. Mamba-2 scores at chance on MQAR across a four-value learning-rate sweep and three seeds; HALA reaches 0.590. Retrieving a token seen exactly once requires addressing a specific earlier position – impossible for fixed-size compressed state, direct for a window.

We are equally explicit about what this does not buy. **HALA does not lead aggregate perplexity in any domain**, trailing Transformer, Jamba and Mamba-2 in all three, with the deficit widening at distance. Its length-extrapolation stability – 0.964–0.969 of its 1k perplexity at 32k, where a Transformer degrades 16.0–44.6× – is a property of recurrent state that Mamba-2 matches, inherited rather than contributed. Its peak memory at long context is the highest we measure. Against Mamba-2 the picture is a crossover with a legible mechanism, not a win: HALA leads near-token accuracy in two of three domains, Mamba-2 leads far tokens in all three, and Mamba-2 carries 3.4× the recurrent state per layer.

Our contributions are: a within-block hybrid using a learned per-head gate over sliding-window attention and an MLA-style low-rank gated recurrence, with no feed-forward sublayer (Section 3); retrieval-distance-stratified evidence that the two paths cover complementary ranges, with the division falling at the window boundary – evidence prior within-block hybrids do not report (Section 5.2); a scaling ladder showing the recurrent path's contribution grows with parameter count across 6.25× (Section 5.3); gate telemetry including a null result, where MQAR routing shows no distance sensitivity while the model still scores 0.590 (Section 5.6); and parameter-matched comparisons across six architectures with two disclosed mismatches reported rather than concealed (Section 4).

All results sit at 11.9–74.0M non-embedding parameters and 0.57B training tokens. The ladder is evidence the central effect is not a small-model artifact; it is not evidence about billion-parameter behaviour, and we do not present it as such.

2 Related Work

Two escapes from quadratic cost, two different losses. Restricting attention to a sliding window bounds cost at $O(NW)$ (Beltagy et al., 2020; Jiang et al., 2023) and gives up everything outside the band. Replacing attention with a recurrence bounds cost at $O(N)$ but compresses the prefix into fixed state: linear attention (Katharopoulos et al., 2020) reorders the attention product, RetNet (Sun et al., 2023) and gated linear attention (Yang et al., 2024) add explicit decay, and selective state-space models (Gu & Dao, 2024; Dao & Gu, 2024) make the recurrence input-dependent. HALA's recurrent path belongs to this family.

The failure modes differ in kind. A window's horizon is *positional* – everything beyond W is invisible regardless of content. A compressed state's limit is *capacity*: Arora et al. (2024a) show recall accuracy scales with state size, so exact retrieval degrades as the prefix grows. That asymmetry is why combining them is worth trying, and why our evaluation stratifies by distance.

Recall as diagnostic. Arora et al. (2024a) isolate associative recall as the capability separating recurrent models from attention and introduce MQAR to measure it. Based (Arora et al., 2024b) responds architecturally – pair linear attention with a short window, so the window absorbs local associations and frees recurrent capacity for distant ones. That is a claim about division of labour, and to our knowledge it has not been tested by measurement; Based reports aggregate recall and throughput, not per-distance attribution. Section 5.2 tests it directly.

Hybrids, by granularity. Jamba (Lieber et al., 2024), Griffin (De et al., 2024) and Samba (Ren et al., 2024) interleave whole attention and recurrent layers across depth, so each layer performs one kind of mixing. A second group composes both inside a layer: B'MOJO (Zancato et al., 2024) fuses

a state-space recurrence with sliding-window attention and a sparse memory; Hymba (Dong et al., 2025) runs attention and Mamba heads in parallel and fuses their outputs; Native Hybrid Attention (Du et al., 2025) keeps long-range context in linear-RNN slots, appends window tokens, and applies one softmax across both.

HALA is in the second group and **we do not claim within-block hybridisation as a contribution.** Three things differ: the paths are combined by a *learned per-head gate* rather than a fixed fusion rule – Qiu et al. (2025) study head-specific gating but gate one mechanism's output rather than routing between two; the recurrent path's keys and values come from a shared low-rank code (DeepSeek-AI, 2024); and the block has no feed-forward sublayer, which is why baselines are matched by solving for *their* FFN width.

What we add beyond the architecture is the measurement. None of the above reports which path handles which retrieval distances – they report aggregate quality, which averages a null effect and a real one into one number revealing neither. We also report what the composition does not buy: Jamba outperforms HALA on every quality metric here.

Implementation. The local path uses FlashAttention-style tiling (Dao et al., 2022; Dao, 2024) on a causal band, with rotary embeddings (Su et al., 2024) on both paths. Training budgets depart from compute-optimal guidance (Hoffmann et al., 2022), as explained in Section 4.

3 Architecture

A HALA layer replaces the transformer block entirely: no position-wise feed-forward sublayer. Each layer computes sliding-window attention and a compressed linear recurrence in parallel, then blends them with a gate learned per head. The whole non-embedding budget goes to mixing and routing.

For input $x_\ell \in \mathbb{R}^{N\times d}$ with H heads and $d_k = d/H$,

$$x_{\ell+1} = x_\ell + W_o\ \mathrm{lerp}\big(o^{\mathrm{lat}}, o^{\mathrm{loc}}, \tilde{g}\big), \qquad \big(o^{\mathrm{loc}}, o^{\mathrm{lat}}, \tilde{g}\big) = f\big(\mathrm{LN}(x_\ell)\big). \tag{1}$$

All projections come from one fused GEMM of shape $d \times (4d + L + H)$, where $L = d/8$ is the latent rank.

Local path. Causal attention over a band of $W = \min(64, N)$ tokens, computed on ℓ_2-normalized queries and keys:

$$o_i^{\mathrm{loc}} = \sum_{i-W<j\le i} \mathrm{softmax}_j\big(s\,\hat{q}_i^\top \hat{k}_j\big)\, v_j, \qquad \hat{q} = q/(\|q\|_2 + 10^{-6}). \tag{2}$$

The scale needs care. Conventional attention divides by $\sqrt{d_k}$ to cancel the norm growth that normalization has already removed – $\hat{q}^\top\hat{k}$ is a cosine in $[-1, 1]$ regardless of d_k – so applying $1/\sqrt{d_k}$ divides twice. At $d_k = 128$ this gives logits of standard deviation 0.0078 against 1.00 for ordinary attention, $128\times$ flatter, under which a perfectly aligned key receives at most $1.09\times$ uniform weight across the window. The path becomes a box average that training cannot sharpen. We set $s = \sqrt{d_k}$, multiplying rather than dividing. This changes the computed function, not a hyperparameter; each result row records which setting applied.

Latent path. Following multi-head latent attention (DeepSeek-AI, 2024), keys and values for all heads come from one shared low-rank code $c_t \in \mathbb{R}^L$ via $[k_t^{\mathrm{lat}} \| v_t^{\mathrm{lat}}] = W_{\mathrm{up}} c_t$ ($W_{\mathrm{up}} \in \mathbb{R}^{2Hd_k\times L}$) – a $d \to L \to 2d$ bottleneck, not a per-head factorization. Rotary embeddings apply to q^{lat} and k^{lat}; v^{lat} is left unrotated.

Each token and head emits a retention coefficient

$$\gamma_{t,h} = \gamma_{\min} + (\gamma_{\max} - \gamma_{\min})\,\sigma\big(\mathrm{clamp}(a_{t,h}, -8, 8)\big) + 10^{-4}, \qquad (\gamma_{\min}, \gamma_{\max}) = (0.88, 0.999). \tag{3}$$

Retention over n tokens is γ^n: 37% at 10^3, 13.5% at 2×10^3, 1.7% at 4096. Half-retention at 4096 needs $\gamma = 0.99983$, outside this parameterization – unlike Mamba-2's unbounded $\exp(A\Delta t)$ (Dao & Gu, 2024). Section 5.4 tests whether this ceiling binds in practice; it does not.

Evaluation is chunked at $S = 64$. With g_i the within-chunk cumulative log-decay, tokens interact inside a chunk through $D_{ij} = \exp(\mathrm{clamp}(g_i - g_j, -20, 0))$ for $j \le i$ – exactly the accumulated decay $\prod_{t=j+1}^{i} \gamma_t$ – and chunks communicate through a running state, giving $O(N)$ cost. Appendix B gives the full intra/inter decomposition. The path output is layer-normalized before blending, bounding its scale per head.

Decay orientation. All results use $g_i - g_j$. An earlier implementation computed $g_j - g_i$; since $\log \gamma < 0$ makes g decreasing, every causal entry was non-negative and the subsequent $\mathrm{clamp}(\cdot, \max = 0)$ pinned it to zero, silently reducing the intra-chunk term to undecayed linear attention. This flag is not recorded per result row; all runs reported here postdate the correction.

Gate. A two-layer MLP emits one logit per head, passed through a sigmoid at temperature $\tau = 4$ with a fixed non-learned per-head spread and training-time noise. The result passes through an affine fixed at construction from layer depth, biasing early layers local and late layers latent (Appendix C). This imposes a depth gradient on an untrained model, so any claim about learned routing must read the pre-shaping signal – all telemetry in Section 5.6 does.

The blend is $\mathrm{lerp}(o^{\mathrm{lat}}, o^{\mathrm{loc}}, \tilde{g}) = (1 - \tilde{g})o^{\mathrm{lat}} + \tilde{g}\, o^{\mathrm{loc}}$: $\tilde{g} \to 1$ is fully local, $\tilde{g} \to 0$ fully latent. Two ablations replace it – `hard`, applying $\lfloor g \rceil$ with a straight-through estimator, and `static`, fixing $\tilde{g} = 0.5$. An auxiliary term pushes gates away from 0.5; introduced for the hard threshold, weaker in motivation under continuous gating, and ablatable (Appendix C).

Per-layer parameter counts by module are in Appendix G.8. Because HALA has no FFN, no closed-form parity condition against FFN-bearing baselines exists; we match empirically (Section 4).

4 Experimental Setup

Data. Three corpora chosen to span different dependency structure: **mixed** (WikiText-103), **code** (`Magicoder-Evol-Instruct-110k`, instruction and response concatenated), and **books** (PG-19). Code is dense in short-range structure – matched delimiters, identifiers repeated within a scope – while long-form prose carries dependencies spanning pages. Since our central measurement stratifies by retrieval distance, varying where those dependencies fall is the point rather than incidental.

Each corpus is capped at 5×10^7 tokens, matching domains on token budget rather than document count. We tokenize with the LLaMA SentencePiece tokenizer ($|V| = 32{,}000$) and pack documents into contiguous 1024-token blocks, discarding the remainder; nothing is ever padded. Dataset identifiers and tokenizer details are in Appendix D.

Parameter parity. All models are 8 layers at $d = 512$, $H = 8$. Matched counts cannot be obtained by construction – HALA has no feed-forward sublayer, Mamba-2 no attention, Jamba both – so a comparison that ignores this measures capacity, not architecture. We match empirically: for each baseline, binary-search one width knob until its non-embedding count matches HALA's 11,852,352, instantiating candidates on the `meta` device so the search costs no memory. Five of six land within 0.2% (Appendix G.9).

Two exceptions need stating. Based cannot be matched at this shape: `num_key_value_heads` must divide H, leaving $\{1, 2, 4, 8\}$, of which only 8 produces a runnable model – the rest fail a forward pass – and that setting sits 55.7% below HALA. Mamba-2's head dimension is reduced from 64 to 32 to reach tolerance; this preserves total recurrent state while increasing head count, so it is not a capacity change, but it is an inductive-bias change on the axis we measure. Jamba is configured dense so active and total parameters coincide.

Recurrent state per layer is 32,768 for HALA, 110,592 for Mamba-2 ($3.4\times$), 139,776 for Based ($4.3\times$). This asymmetry is the mechanism behind the near/far crossover in Section 5.1: HALA spends the difference on an attention window.

Training. 35,000 steps at batch size 16, block size 1024 – 0.57B tokens per run, roughly 48 tokens per non-embedding parameter. This is deliberately overtrained relative to compute-optimal guidance (Hoffmann et al., 2022), because the comparison of interest is between architectures at a fixed budget, not each at its own optimum.

AdamW, weight decay 0.01 excluding biases and normalization parameters, gradient clipping at 1.0, linear warmup over $3{,}500$ steps then cosine decay floored at 10% of peak (Appendix E). Peak learning rate is set per architecture, not shared – a single rate measures which architecture happens to suit it. Based is the exception and is untuned, inheriting HALA's 4×10^{-4}. With its parameter deficit that gives two independent confounds, either sufficient to explain any gap; Based results should be attributed to learning rate before architecture. Three seeds per model–domain cell; all results are mean $\pm$ one standard deviation.

Retrieval-distance metrics. Aggregate perplexity averages over tokens whose antecedent is one position back and tokens whose antecedent is a thousand, so it cannot answer the question this paper asks. A target at position $i+1$ is an associative-recall (AR) token if the bigram (x_i, x_{i+1}) occurred earlier in the same block; its distance $d = i - j$ uses the most recent such occurrence. We bucket

$$\text{near: } d \leq 64, \qquad \text{far: } d > 64, \qquad \text{gt256: } d > 256, \tag{4}$$

setting the near/far boundary at the local window's span – near tokens are those the window reaches, far tokens are those it cannot. The buckets are not a partition: near and far exhaustively partition AR tokens, but `gt256` is nested inside far, so columns read as "distance at least." We report perplexity and top-1 exact-match accuracy over each set; the latter puts natural-text recall in the same units as MQAR, making the transfer question in Section 6 directly answerable. A `gt1024` bucket is structurally empty at a 1024-token block and omitted.

MQAR. Multi-query associative recall (Arora et al., 2024a) presents key–value pairs followed by queries; the model must retrieve each queried key's value. This is a separate training run from the language-modelling sweep, at $d = 512$, 2 layers, 4 heads, sequence length 512, 16 key–value pairs, $16{,}000$ steps. We sweep four learning rates on a held-out seed, then evaluate the best on three disjoint seeds, using the Zoology reference generator so results are comparable to published numbers. MQAR runs on the PyTorch reference kernel path: its $d_k = 128$ geometry is outside the verified set for our fused kernels, which fault at that head dimension (Appendix F).

Efficiency. Batch size 1, contexts 2^{10}–2^{17}, median of five runs after two warmups. Peak VRAM is reported with and without the language-model head, since a 32k-vocabulary logit tensor dominates at long context – roughly 7.8 GiB at 131k – and is identical across architectures. Each row records the kernel that actually executed, resolved after the fact: the sliding-window baseline changes implementation across lengths, and reporting a requested backend would conceal that (Section 5.7).

5 Results

5.1 Language modelling

HALA does not lead aggregate perplexity in any domain (Table 3, Table 4, both in Appendix G.1): 32.33 ± 0.40 on mixed against Jamba's 24.78 ± 0.19, Transformer's 27.07 ± 0.10 and Mamba-2's 27.81 ± 0.37, with the same ordering on code and books. The deficit concentrates at distance – on mixed far tokens HALA reaches 8.31 ± 0.32 perplexity where Transformer and Jamba reach 4.28 and 4.26, and 0.576 ± 0.005 accuracy against 0.682 for both – while on near tokens it sits within a few points of the attention baselines.

Against Mamba-2 specifically the picture is a crossover, not a loss. HALA leads on near-token perplexity and accuracy in mixed (3.24 vs 3.69; 0.766 vs 0.740) and code (1.97 vs 2.05; 0.827 vs 0.814) and ties on books, while Mamba-2 leads on far tokens in all three. The state-size asymmetry of Section 4 – $3.4\times$ in Mamba-2's favour – is the mechanism: larger compressed state buys long range, an explicit window buys short-range precision. Based is reported at a disclosed parameter deficit and an untuned learning rate, and only on mixed at a single seed; we draw no architectural conclusion from it.

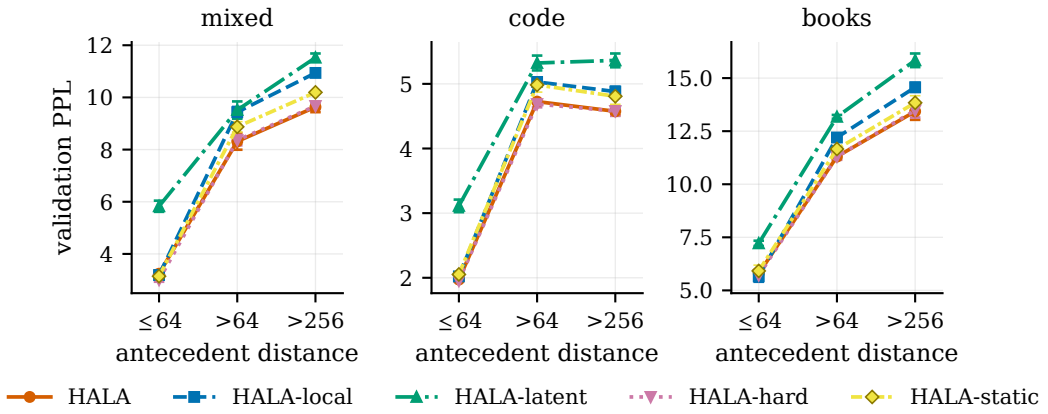

Figure 1: Perplexity against retrieval distance for the routing ablation (HALA, local-only, latent-only, hard-gate, static-gate), per domain. Each panel has its own y-axis; absolute perplexity is not comparable across corpora.

5.2 Component ablation

Figure 1 isolates the block's parts, with per-domain values in Appendix G.2. The ordering is identical in all three domains:

$$\{\text{hard-gate, HALA}\} < \{\text{static-gate, local-only}\} \ll \text{latent-only} \tag{5}$$

Hard and soft gating are statistically indistinguishable at the front; static-gate and local-only are not consistently separable from each other, tying on code (8.61 both) and differing within seed noise on books. Removing the latent path costs 2.31, 0.40 and 1.88 perplexity across the three domains; removing the local path costs 10.83, 2.62 and 11.39; replacing the learned gate with a fixed blend costs 1.36, 0.40 and 1.63. Every component contributes, and the recurrence alone is by a wide margin the weakest configuration.

The two paths cover complementary ranges. Against the local-only ablation, HALA is indistinguishable on antecedents inside the window – 3.24 vs 3.19 on mixed, 1.97 vs 2.02 on code, 5.79 vs 5.66 on books, with the ablation marginally ahead in two of three – and consistently ahead outside it: 8.31 vs 9.43, 4.73 vs 5.03, 11.30 vs 12.20. Accuracy moves the same way (+0.022, +0.010, +0.012 on far; −0.003, +0.006, −0.003 on near). The latent path contributes nothing the window already covers and something measurable beyond it – a dissociation aggregate perplexity, differing by 2.31, 0.40 and 1.88, cannot show.

5.3 Parameter scaling

Figure 4 (Appendix G.3) extends the HALA vs local-only comparison across three widths at fixed token budget on books.

The gap grows monotonically with parameter count: far-token perplexity advantage 0.97 at 26.7M non-embedding parameters, 2.04 at 47.4M, 3.72 at 74.0M; far-token accuracy advantage 0.012, 0.021, 0.029. The latent path's contribution is larger, not smaller, as capacity increases.

Two caveats. The budget is fixed across rungs, so tokens per parameter falls from 48 to 8 and the larger models are undertrained – which is why absolute far-token perplexity *rises* across the ladder for both models (11.35 → 13.07 for HALA). Three points across 6.25× support "the effect is present and growing across this range," not extrapolation to arbitrary scale.

5.4 Length extrapolation

Models trained at 1024 tokens, evaluated to 32,768 unmodified (Figure 5 and per-length table, both in Appendix G.4).

Model	Accuracy
HALA	0.590 ± 0.002
Transformer	$\mathbf{0.999 \pm 0.000}$
Mamba-2	0.000 ± 0.000
Local Attn	0.025 ± 0.001
Based	0.051 ± 0.010

Table 1: MQAR recall accuracy on answer positions.

Attention collapses and the recurrent architectures do not. Transformer's 32k perplexity ratio against its 1k baseline is $16.0\times$ on mixed, $44.6\times$ on code, $18.6\times$ on books, where HALA's is 0.964, 0.966, 0.969 – perplexity improves slightly with context and stays stable to $32\times$ training length. Seed variance is correspondingly asymmetric: Transformer reaches ± 60.6 at 32k on mixed, HALA ± 0.001 on the ratio.

This is a property of the recurrent formulation, not of HALA: Mamba-2 achieves 0.966, 0.958, 0.975. Jamba, interleaving both, degrades but far less than pure attention ($1.9\times$, $2.6\times$, $2.5\times$). We report it as inherited, not as an advantage over SSMs. Raising the decay ceiling post hoc on a trained checkpoint changes perplexity in the third decimal (Appendix G.7); the ceiling is not what bounds extrapolation.

5.5 Associative recall

On MQAR (setup in Section 4; Table 1), HALA reaches 0.590 ± 0.002 where Mamba-2 sits at 0.000 ± 0.000 – chance – across a full four-value learning-rate sweep and three seeds. Based reaches 0.051, sliding-window attention alone 0.025, and Transformer solves the task at 0.999.

The separation from Mamba-2 is structural: retrieving a token seen once requires addressing a specific earlier position, which fixed-size compressed state cannot do in principle and a window can do directly. That this advantage does **not** transfer to natural-language perplexity, where Mamba-2 leads on far tokens in all three domains, is itself informative; Section 6 returns to it.

5.6 What the gate learns

Figure 2 shows pre-shaping gate values by layer and head, reported before the depth affine since that affine imposes a depth gradient on an untrained model.

Trained gates average ≈ 0.65 with head-to-head standard deviation ≈ 0.16, against 0.499 ± 0.073 at initialisation – a $2.2\times$ widening and a shift toward local. Heads do not collapse to a common value; indices permute across seeds under permutation symmetry while the distribution stays stable. Per-domain macro-averages differ by less than a fifth of within-domain spread (0.655, 0.626, 0.639), so we claim no domain-specific routing.

On MQAR the same measurement gives a different answer. Gates converge to 0.064 and 0.112 in layers 0 and 1, routing roughly 90% to the latent path, and are **flat with respect to retrieval distance** – the step at the window boundary is -0.000 and $+0.000$ against head spreads of 0.021 and 0.012 (Figure 7, Appendix G.5). The architecture's central routing prediction is not met on that benchmark. We report it as a null result.

5.7 Efficiency

Table 2 reports time-to-first-token and peak memory at batch size 1.

HALA's cost profile crosses over with context. Slowest of the compiled models at 1024 tokens (15.2 ms against Transformer's 3.0), fastest by 131,072 (127.4 ± 0.1 ms against 831.4 ± 2.1, a $6.5\times$ advantage), with the crossover between 8k and 16k. It also leads the reference Mamba-2 (189.0 ms) and Jamba (338.9 ms) implementations, though both are unfused PyTorch references – not a like-for-like kernel comparison.

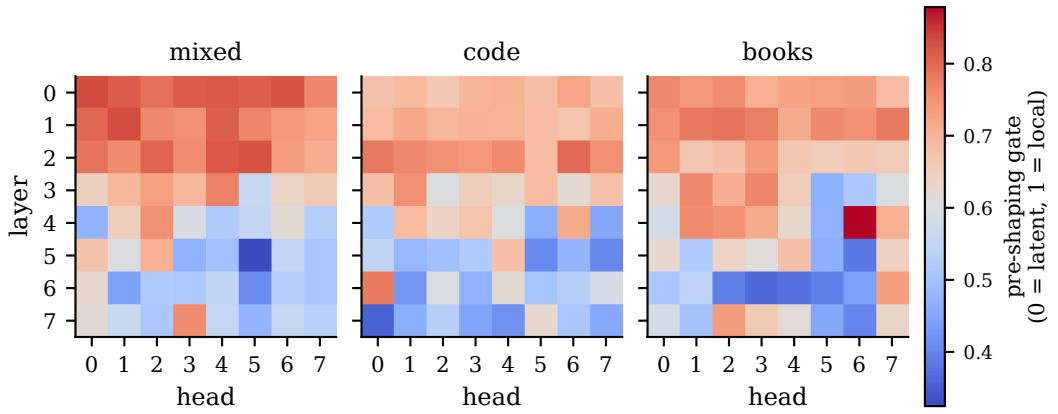

Figure 2: Pre-shaping gate by layer and head, per domain. One colour scale is shared across all panels. The pre-shaping view is plotted because the depth-dependent affine applied at construction produces depth structure in an untrained model, so only this view supports a claim about what was learned.

Model	TTFT@1k (ms)	VRAM@1k (GiB)	TTFT@128k (ms)	VRAM@128k (GiB)
HALA	15.2 ± 0.0	0.30 ± 0.00	$\mathbf{127.4 \pm 0.1}$	8.67 ± 0.00
Transformer	$\mathbf{3.0 \pm 0.0}$	0.40 ± 0.00	831.4 ± 2.1	8.22 ± 0.00
Local Attn	1711.7 ± 28.4	0.34 ± 0.00	2330.9 ± 28.9	8.24 ± 0.00
Jamba-REF	5.0 ± 0.0	0.34 ± 0.00	338.9 ± 1.3	8.22 ± 0.00
Mamba-2-REF	4.3 ± 0.1	$\mathbf{0.28 \pm 0.00}$	189.0 ± 6.6	$\mathbf{8.15 \pm 0.00}$

Table 2: Time-to-first-token and peak VRAM at representative context lengths (endpoints of the sweep in Figure 8), best available implementation per model. The full grid – every context length, eager and compiled – is in the released `hardware_efficiency.csv`.

HALA does not win on memory. 8.67 GiB at 131k against 8.15–8.24 GiB for the baselines, the highest in the table, though the 7.8 GiB logit tensor common to all architectures dominates the totals.

Sliding-window attention is reported eager, because its compiled path is anomalous – 1711.7 ms at 1k, nearly flat to 2330.9 ms at 131k, inconsistent with windowed attention and indicating compilation did not produce a working kernel. Its eager measurements are 10.1 ms at 1k rising to 28.3 ms at 32k under FlexAttention, then 1072.4 ms and 2425.6 ms at 64k and 131k after the kernel exhausts memory and falls back to blocked SDPA. It is the only model not reported compiled, and we flag it rather than presenting a uniform but meaningless column.

6 Discussion

Two paths, two ranges. The clearest result here is a dissociation. Against its own local-only ablation, HALA is indistinguishable on antecedents inside the 64-token window – 3.24 vs 3.19 on mixed, 1.97 vs 2.02 on code, 5.79 vs 5.66 on books, with the ablation marginally ahead in two of three – and consistently ahead outside it, by 1.12, 0.30 and 0.90. The recurrent path contributes nothing the window already covers and something measurable beyond it. The effect also survives scale: the far-token advantage grows monotonically from 0.97 at 26.7M non-embedding parameters to 3.72 at 74.0M.

Against Mamba-2: a crossover with a mechanism. At matched parameters the two win on opposite sides of the same boundary. HALA leads near-token accuracy on mixed (0.766 vs 0.740) and code (0.827 vs 0.814) and ties books; Mamba-2 leads far tokens in all three. State sizes explain the direction – Mamba-2 carries 110,592 recurrent parameters per layer to HALA's 32,768, while HALA spends the difference on an attention window. Larger compressed state buys long range; explicit local attention buys short-range precision.

Why exact recall does not transfer. On MQAR, Mamba-2 scores 0.000 across a four-value learning-rate sweep and three seeds – chance at every setting – while HALA reaches 0.590. The separation is structural: retrieving a token seen once requires addressing a specific earlier position, which fixed-size compressed state cannot do in principle. MQAR isolates that capability and scores nothing else; it also places 58.3% of queries within the 64-token window, so the benchmark is structurally biased toward the path that can address positions directly. Natural text is different – most predictions are supported by many partially informative cues, a lossy summary usually suffices, and the rare token needing exact recall contributes little to an averaged loss. The capability is real and structurally absent from pure SSMs; it is simply not what limits perplexity at this scale.

Our own null supports this. On MQAR the gate converges to 0.064 and 0.112 in layers 0 and 1, routing roughly 90% to the latent path with no sensitivity to retrieval distance – the step at the window boundary is -0.000 and $+0.000$ against head spreads of 0.021 and 0.012 – and HALA still reaches 0.590. The local path carries that result despite the routing, not because of it, which is what one expects if exact retrieval is available or unavailable rather than something to allocate.

What we cannot claim. HALA trails Transformer, Jamba and Mamba-2 on aggregate perplexity in all three domains, with the deficit widening at distance. Its length-extrapolation stability (0.964–0.969 of 1k perplexity at 32k, where Transformer degrades 16.0–$44.6\times$) is a property of recurrent state that Mamba-2 matches at 0.958–0.975, inherited rather than contributed. Its peak memory at 131k is the highest we measure, 8.67 GiB against 8.15–8.24. Based carries a 55.7% parameter deficit and an untuned learning rate – two confounds, either sufficient alone – so we draw no architectural conclusion from it. Hard and soft gating are indistinguishable at convergence (32.09 vs 32.33, 8.19 vs 8.21, 44.73 vs 44.75), and we default to the continuous gate for differentiability under compilation rather than measured superiority. Trained gates spread $2.2\times$ wider than at initialisation but their per-domain macro-averages differ by less than a fifth of within-domain spread, so we claim no domain-adaptive routing. And every result sits at 11.9–74.0M non-embedding parameters and 0.57B tokens: the ladder is evidence against a small-model artifact, not evidence about billion-parameter behaviour.

Relation to layer-wise hybrids. Jamba interleaves whole layers; HALA composes both mechanisms in every block and mixes them per head and per token. The granularity differs, and so does what can be measured – a per-head gate exposes a routing signal a layer-wise arrangement has no analogue for, which is what makes Section 5.6's telemetry, including its null, available at all. Whether within-block composition is preferable is not settled by our results: Jamba outperforms HALA on every quality metric here. Substituting a HALA block for the recurrent layers of a layer-wise hybrid is a natural next experiment we have not run.

7 Conclusion

We set out to test an assumption hybrid architectures make but rarely verify: that a local window and a compressed recurrence divide labour, each covering what the other cannot. Stratifying by retrieval distance, we find the division is real and falls where the design predicts – at the window boundary – and that the recurrent path's contribution grows rather than shrinks across a $6.25\times$ range of parameter counts.

The same discipline produces less flattering results. Aggregate perplexity places HALA behind every baseline tested. Length-extrapolation stability is a property Mamba-2 shares, inherited rather than earned. Against Mamba-2 the picture is a crossover with a legible mechanism rather than a win. And our gate telemetry includes a null: on MQAR the routing shows no sensitivity to retrieval distance while the model still scores 0.590, meaning the local path carries that result despite the gate rather than because of it.

Two directions follow: scale, since our ladder reaches 74M and says nothing about billion-parameter behaviour; and composition granularity, since layer-wise and within-block hybrids are rarely compared under matched conditions. More broadly, the gap between our MQAR and natural-text results seems worth taking seriously – a capability can be structurally absent from an architecture, cleanly demonstrable on a synthetic probe, and still not be what limits perplexity.

REPRODUCIBILITY STATEMENT

The complete experimental pipeline is provided as supplementary material, including model definitions, training and evaluation scripts, kernel implementations, plotting and table-generation code, and the result files from which every table and figure in this paper is derived.

Architecture details sufficient to reimplement HALA appear in Section 3, with the chunked recurrence decomposition, gate construction, and per-layer parameter counts in Appendices B, C, and G.8. Data processing, tokenization, and packing are described in Appendix D; optimization and the learning-rate schedule in Appendix E; the MQAR task configuration in Appendix F. Parameter parity against baselines is solved programmatically rather than by hand; the procedure and per-baseline results, including two cases where matching was not achievable and is disclosed instead, are in Section 4 and Appendix G.9. Column semantics for all released result files are documented in `COLUMNS.md` in the supplementary material.

Derived tables in this paper are generated from raw per-seed data by a verification script that independently recomputes each cell and fails on mismatch; that script is included. Four items are not reproducible from the released artifacts alone and are identified where they appear: Based's efficiency numbers, the sliding-window baseline's compiled path (Section 5.7), the gate-telemetry figures (Section 5.6), and MQAR's accuracy comparison, whose released file is seed-averaged rather than per-seed.

AI USE STATEMENT

Large language models were used substantially in this work, and we describe that use precisely.

Writing. All prose in this paper was drafted with Claude (Anthropic) in an iterative loop: the author specified the argument, scope, and framing of each section; the model produced drafts; the author reviewed, corrected, and directed revisions. Every claim, number, and framing decision was verified by the author against the underlying data before inclusion. Section ordering, the decision to lead with the measurement rather than the architecture, and the scope of every claim were author decisions.

Code. The experimental pipeline – training scripts, evaluation harnesses, Triton kernels, plotting and table-generation code – was written with Claude Code (Anthropic) under the author's direction. The author specified experiments, reviewed implementations, and diagnosed failures. Several correctness bugs found during development were identified by the model during verification passes it was directed to run, including the intra-chunk decay orientation error described in Section 3 and a data-provenance error in which test-fixture output overwrote real result tables.

Research direction. Experimental design, hypotheses, and interpretation were developed in dialogue between the author and the model. The central methodological choice of this paper – stratifying evaluation by retrieval distance to attribute each path's contribution – emerged from that dialogue. The author made all decisions about what to run, what to report, and what the results support. In several instances the author overruled the model's recommendation, including insisting on convergence reruns that overturned a preliminary conclusion the model had endorsed.

Verification. All reported numbers derive from the author's own experimental runs. Tables and figures are generated programmatically from result files, and a verification script independently recomputes each derived table from the raw per-seed data (Appendix G.9 describes the parity methodology; the reconciliation check is in the released code). The author takes full responsibility for the paper's contents.

REFERENCES

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

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall-throughput

tradeoff. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *PMLR*, pp. 1763–1840, 2024b.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *PMLR*, pp. 10041–10071, 2024.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Soham De, Samuel L. Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, Guillaume Desjardins, Arnaud Doucet, David Budden, Yee Whye Teh, Razvan Pascanu, Nando De Freitas, and Caglar Gulcehre. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*, 2024.

DeepSeek-AI. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, Zijia Chen, Ameya Sunil Mahabaleshwarkar, Shih-Yang Liu, Matthijs Van Keirsbilck, Min-Hung Chen, Yoshi Suhara, Yingyan Lin, Jan Kautz, and Pavlo Molchanov. Hymba: A hybrid-head architecture for small language models. In *International Conference on Learning Representations (ICLR)*, 2025.

Jusen Du, Jiaxi Hu, Tao Zhang, Weigao Sun, and Yu Cheng. Native hybrid attention for efficient sequence modeling. *arXiv preprint arXiv:2510.07019*, 2025.

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

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *PMLR*, pp. 5156–5165, 2020.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Haim Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avshalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*, 2024.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. In *Advances in Neural Information Processing Systems*, 2025.

Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. Samba: Simple hybrid state space models for efficient unlimited context language modeling. *arXiv preprint arXiv:2406.07522*, 2024.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Proceedings of the 41st International Conference on Machine Learning*, PMLR, pp. 56501–56523, 2024.

Luca Zancato, Arjun Seshadri, Yonatan Dukler, Aditya Golatkar, Yantao Shen, Benjamin Bowman, Matthew Trager, Alessandro Achille, and Stefano Soatto. B'MOJO: Hybrid state space realizations of foundation models with eidetic and fading memory. In *Advances in Neural Information Processing Systems*, volume 37, pp. 130433–130462, 2024.

A MQAR ROUTING ABLATION: SCOPE DECISION

Under the window-coverage bias described in Section 6, the MQAR ablation ordering *reverses* the real-text ablation ordering: on MQAR the local-only ablation led, the static-gate and hard-gate variants followed, and the learned soft gate trailed, while on natural text HALA beats the local-only ablation on overall perplexity in all three domains. Reporting that reversal without this context risks being read as evidence against the architecture rather than evidence about MQAR's window-coverage confound.

B CHUNKED RECURRENCE

Within a chunk, tokens interact directly:

$$o^{\text{intra}} = \big(QK^\top \odot D\big)V, \qquad D_{ij} = \begin{cases} \exp\big(\operatorname{clamp}(g_i - g_j,\, -20,\, 0)\big) & j \le i, \\ 0 & j > i, \end{cases} \tag{6}$$

so that $D_{ij} = \prod_{t=j+1}^{i} \gamma_t$ is exactly the accumulated decay from key j to query i. Chunks communicate through a running state,

$$h_{c+1} = h_c \exp(g_{S-1}) + \textstyle\sum_j k_j \otimes v_j \exp(g_{S-1} - g_j), \qquad o_c^{\text{inter}} = (Q_c h_c) \odot \exp(g), \tag{7}$$

and the path outputs

$$o^{\text{lat}} = \operatorname{LN}\big(o^{\text{intra}} + o^{\text{inter}}\big). \tag{8}$$

C GATE DETAILS

The gate MLP is

$$z = W_2\, \operatorname{GELU}(W_1\, \operatorname{LN}(x)), \qquad g = \sigma\big(\tau\,(z \odot \rho + \eta)\big), \tag{9}$$

with $W_1 \in \mathbb{R}^{m \times d}$, $W_2 \in \mathbb{R}^{H \times m}$, $m = \max(d/4, 32)$, a fixed non-learned per-head spread $\rho = \operatorname{linspace}(0.85, 1.15, H)$ held as a buffer, and anti-stagnation noise $\eta \sim \mathcal{N}(0, 0.05^2)$ applied during training only.

The depth prior is a fixed affine resolved once per layer from its depth index:

$$\tilde{g}_\ell = \alpha_\ell\, g + \beta_\ell, \qquad (\alpha_\ell, \beta_\ell) = \begin{cases} (0.70,\ 0.30) & \ell < L/3 \\ (0.70,\ 0.00) & \ell > 0.7L \\ (1.00,\ 0.00) & \text{otherwise} \end{cases} \tag{10}$$

biasing early layers toward local and late layers toward latent.

The `hard` gate variant applies $\lfloor g \rceil$ in the forward pass with a straight-through estimator $g + (\lfloor g \rceil - g)_{\text{detach}}$. The auxiliary term

$$\mathcal{L}_{\text{aux}} = 0.05\big(0.25 - \overline{(g - 0.5)^2}\big) \tag{11}$$

pushes gates away from 0.5, with default weight 0.1.

D Dataset details

Mixed uses `Salesforce/wikitext` (`wikitext-103-raw-v1`); code uses `ise-uiuc/Magicoder-Evol-Instruct-110k`, with text formed by concatenating instruction and response fields; books uses `emozilla/pg19`. Tokenization uses `hf-internal-testing/llama-tokenizer`. The tokenizer defines no padding token; we set `pad_token_id = eos_token_id`, though nothing is ever padded in practice – documents are concatenated and packed into contiguous blocks with the trailing remainder discarded, so no pad token is ever scored.

E Training schedule

$$\text{lr}(t) = \text{lr}_{\text{peak}} \cdot \begin{cases} t/T_w & t < T_w \\ \max\Big(0.1,\ \frac{1}{2}\big(1 + \cos(\pi p)\big)\Big), \quad p = \dfrac{t - T_w}{T - T_w} & t \geq T_w \end{cases} \tag{12}$$

Peak learning rate by architecture: HALA 4×10^{-4}, Transformer and local attention 3×10^{-4}, Mamba-2 8×10^{-4}, Jamba 6×10^{-4}. Three seeds per model–domain cell: 42, 64, 128.

F MQAR: additional detail

Vocabulary size 8,192; 8,192 training and 512 validation sequences; batch size 32. Peak learning rate is chosen by a sweep over four values (10^{-4}, 3×10^{-4}, 10^{-3}, 3×10^{-3}) at a fixed sweep seed of 42; final results are reported over three seeds (64, 128, 256), held disjoint from the sweep seed so accuracy is never reported on the same draw that selected its learning rate.

Data comes from the Zoology reference generator (Arora et al., 2024a) rather than a local reimplementation. Format risk is the largest remaining source of error in this experiment: a locally-reimplemented task can be internally consistent and still not comparable to published numbers, so where the reference generator is available we use it verbatim and inherit its format exactly.

The kernel-fault justification, from `hala_decay_recurrence.py`: "Probing only at B=1,H=2 let a d_k=128 / H=4 / B=256 configuration reach training and fault there with an illegal memory access instead of failing pre-flight and falling back cleanly." MQAR's shape ($d_k = 128$, $H = 4$) is exactly this configuration, so runs are forced onto the PyTorch reference kernel path at process start; MQAR measures recall accuracy, not throughput, so the reference path costs nothing we care about here.

G Supplementary results: relocated tables and figures

Items below are relocated from Section 5, not new content; each subsection here corresponds to the identically-named subsection of the main results. Nothing here is deleted from the paper – only moved to keep the body within length.

G.1 LANGUAGE MODELLING

Domain	Model	PPL_{all}	PPL_{near}	PPL_{far}	$\text{PPL}_{>256}$
mixed	HALA	32.33 ± 0.40	3.24 ± 0.13	8.31 ± 0.32	9.63 ± 0.20
	Transformer	27.07 ± 0.10	$\mathbf{2.40 \pm 0.04}$	4.28 ± 0.09	5.83 ± 0.06
	Mamba-2	27.81 ± 0.37	3.69 ± 0.08	6.26 ± 0.18	8.09 ± 0.14
	Jamba	$\mathbf{24.78 \pm 0.19}$	2.42 ± 0.03	$\mathbf{4.26 \pm 0.04}$	$\mathbf{5.75 \pm 0.03}$
	Based	69.14	8.41	12.88	15.58
code	HALA	8.21 ± 0.20	1.97 ± 0.01	4.73 ± 0.00	4.58 ± 0.06
	Transformer	6.99 ± 0.15	1.76 ± 0.02	2.57 ± 0.04	2.97 ± 0.05
	Mamba-2	7.28 ± 0.13	2.05 ± 0.05	3.28 ± 0.14	3.80 ± 0.11
	Jamba	$\mathbf{6.67 \pm 0.12}$	$\mathbf{1.75 \pm 0.01}$	$\mathbf{2.54 \pm 0.02}$	$\mathbf{2.92 \pm 0.05}$
books	HALA	44.75 ± 0.32	5.79 ± 0.16	11.30 ± 0.14	13.43 ± 0.40
	Transformer	39.87 ± 0.82	$\mathbf{4.28 \pm 0.11}$	$\mathbf{8.45 \pm 0.03}$	$\mathbf{10.85 \pm 0.29}$
	Mamba-2	40.57 ± 0.40	5.83 ± 0.14	10.66 ± 0.19	12.75 ± 0.46
	Jamba	$\mathbf{39.09 \pm 0.47}$	4.92 ± 0.11	9.44 ± 0.27	11.63 ± 0.48

Table 3: Validation perplexity against baselines, per domain. Cells are mean $\pm$ std over seeds. Distance buckets are nested: *far* is > 64 tokens and > 256 is a subset of it.

Domain	Model	Acc_{near}	Acc_{far}	$\text{Acc}_{>256}$
mixed	HALA	0.766 ± 0.006	0.576 ± 0.005	0.535 ± 0.002
	Transformer	$\mathbf{0.812 \pm 0.003}$	$\mathbf{0.682 \pm 0.004}$	0.610 ± 0.001
	Mamba-2	0.740 ± 0.002	0.626 ± 0.003	0.565 ± 0.002
	Jamba	0.809 ± 0.003	0.682 ± 0.001	$\mathbf{0.611 \pm 0.002}$
	Based	0.594	0.501	0.453
code	HALA	0.827 ± 0.002	0.625 ± 0.002	0.625 ± 0.002
	Transformer	0.849 ± 0.003	0.750 ± 0.006	0.709 ± 0.006
	Mamba-2	0.814 ± 0.006	0.700 ± 0.011	0.665 ± 0.006
	Jamba	$\mathbf{0.850 \pm 0.002}$	$\mathbf{0.755 \pm 0.004}$	$\mathbf{0.716 \pm 0.005}$
books	HALA	0.669 ± 0.008	0.528 ± 0.004	0.479 ± 0.007
	Transformer	$\mathbf{0.716 \pm 0.001}$	$\mathbf{0.572 \pm 0.004}$	$\mathbf{0.508 \pm 0.006}$
	Mamba-2	0.667 ± 0.003	0.543 ± 0.003	0.490 ± 0.005
	Jamba	0.696 ± 0.006	0.558 ± 0.004	0.501 ± 0.005

Table 4: Top-1 accuracy on AR positions against baselines, per domain.

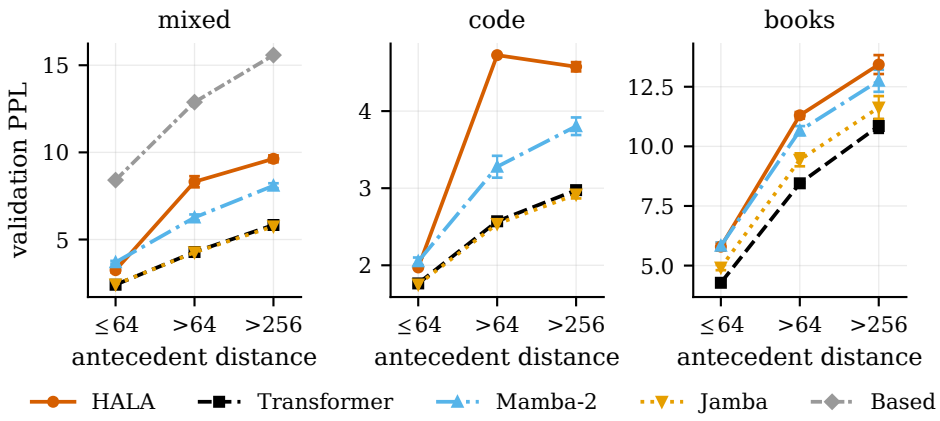

Figure 3: Perplexity against the retrieval distance of the AR token's antecedent, per domain. Each panel has its own y-axis; absolute perplexity is not comparable across corpora.

G.2 COMPONENT ABLATION

Domain	Model	PPL_{all}	PPL_{near}	PPL_{far}	$\text{PPL}_{>256}$
mixed	HALA	32.33 ± 0.40	3.24 ± 0.13	$\mathbf{8.31 \pm 0.32}$	$\mathbf{9.63 \pm 0.20}$
	HALA-local	34.64 ± 0.45	3.19 ± 0.12	9.43 ± 0.20	10.94 ± 0.08
	HALA-latent	43.16 ± 0.35	5.83 ± 0.21	9.52 ± 0.33	11.52 ± 0.16
	HALA-hard	$\mathbf{32.09 \pm 0.18}$	$\mathbf{3.01 \pm 0.07}$	8.36 ± 0.20	9.67 ± 0.10
	HALA-static	33.69 ± 0.23	3.14 ± 0.15	8.87 ± 0.20	10.19 ± 0.06
code	HALA	8.21 ± 0.20	1.97 ± 0.01	4.73 ± 0.00	$\mathbf{4.58 \pm 0.06}$
	HALA-local	8.61 ± 0.24	2.02 ± 0.06	5.03 ± 0.04	4.88 ± 0.08
	HALA-latent	10.83 ± 0.34	3.11 ± 0.10	5.32 ± 0.11	5.36 ± 0.11
	HALA-hard	$\mathbf{8.19 \pm 0.13}$	$\mathbf{1.95 \pm 0.02}$	$\mathbf{4.69 \pm 0.05}$	4.58 ± 0.07
	HALA-static	8.61 ± 0.14	2.05 ± 0.02	4.98 ± 0.11	4.81 ± 0.12
books	HALA	44.75 ± 0.32	5.79 ± 0.16	$\mathbf{11.30 \pm 0.14}$	$\mathbf{13.43 \pm 0.40}$
	HALA-local	46.63 ± 0.91	$\mathbf{5.66 \pm 0.25}$	12.20 ± 0.06	14.57 ± 0.21
	HALA-latent	56.14 ± 1.10	7.22 ± 0.12	13.17 ± 0.09	15.84 ± 0.32
	HALA-hard	$\mathbf{44.73 \pm 0.58}$	5.71 ± 0.25	11.30 ± 0.15	13.48 ± 0.34
	HALA-static	46.38 ± 0.72	5.92 ± 0.27	11.66 ± 0.08	13.84 ± 0.33

Table 5: Routing ablation, validation perplexity per domain. Table rows abbreviate the ablations named in prose: HALA-local = local-only, HALA-latent = latent-only, HALA-hard = hard-gate, HALA-static = static-gate.

Domain	Model	Acc_{near}	Acc_{far}	$\text{Acc}_{>256}$
mixed	HALA	0.766 ± 0.006	$\mathbf{0.576 \pm 0.005}$	$\mathbf{0.535 \pm 0.002}$
	HALA-local	0.769 ± 0.005	0.554 ± 0.003	0.517 ± 0.002
	HALA-latent	0.666 ± 0.006	0.559 ± 0.004	0.509 ± 0.000
	HALA-hard	$\mathbf{0.777 \pm 0.002}$	0.575 ± 0.004	0.533 ± 0.001
	HALA-static	0.771 ± 0.006	0.566 ± 0.002	0.528 ± 0.001
code	HALA	0.827 ± 0.002	$\mathbf{0.625 \pm 0.002}$	$\mathbf{0.625 \pm 0.002}$
	HALA-local	0.821 ± 0.006	0.615 ± 0.004	0.616 ± 0.002
	HALA-latent	0.730 ± 0.008	0.597 ± 0.005	0.591 ± 0.004
	HALA-hard	$\mathbf{0.827 \pm 0.002}$	0.625 ± 0.004	0.625 ± 0.004
	HALA-static	0.818 ± 0.003	0.617 ± 0.006	0.619 ± 0.005
books	HALA	0.669 ± 0.008	$\mathbf{0.528 \pm 0.004}$	$\mathbf{0.479 \pm 0.007}$
	HALA-local	$\mathbf{0.672 \pm 0.005}$	0.516 ± 0.004	0.466 ± 0.005
	HALA-latent	0.636 ± 0.004	0.511 ± 0.002	0.457 ± 0.005
	HALA-hard	0.671 ± 0.010	0.526 ± 0.003	0.476 ± 0.004
	HALA-static	0.665 ± 0.006	0.524 ± 0.004	0.474 ± 0.005

Table 6: Routing ablation, top-1 accuracy on AR positions per domain.

The hard-gate ablation (HALA-hard) reaches 32.09 ± 0.18, 8.19 ± 0.13 and 44.73 ± 0.58 against HALA's 32.33 ± 0.40, 8.21 ± 0.20 and 44.75 ± 0.32 – differences of 0.24, 0.02 and 0.02 against seed standard deviations of comparable or larger magnitude. We default to the continuous gate because it keeps the forward pass differentiable and reproducible under compilation, not because it measurably outperforms the discrete one at this budget.

G.3 PARAMETER SCALING

d_{model}	Non-emb. (M)	Model	PPL_{near}	PPL_{far}	Acc_{near}	Acc_{far}
768	26.66	HALA	5.50 ± 0.29	$\mathbf{11.35 \pm 0.07}$	$\mathbf{0.674 \pm 0.005}$	$\mathbf{0.526 \pm 0.004}$
	26.66	HALA-local	$\mathbf{5.48 \pm 0.14}$	12.32 ± 0.04	0.672 ± 0.002	0.514 ± 0.003
1024	47.38	HALA	$\mathbf{5.34 \pm 0.20}$	$\mathbf{11.84 \pm 0.05}$	$\mathbf{0.673 \pm 0.003}$	$\mathbf{0.520 \pm 0.003}$
	47.38	HALA-local	5.72 ± 0.08	13.88 ± 0.10	0.662 ± 0.005	0.499 ± 0.004
1280	74.02	HALA	$\mathbf{5.64 \pm 0.11}$	$\mathbf{13.07 \pm 0.04}$	$\mathbf{0.666 \pm 0.002}$	$\mathbf{0.510 \pm 0.003}$
	74.02	HALA-local	6.50 ± 0.22	16.79 ± 0.34	0.643 ± 0.002	0.481 ± 0.003

Table 7: Parameter scaling ladder on `books`. Depth (8 layers) and head dimension ($d_k = 64$) are held fixed, so the verified kernel path is unchanged at every rung. **The token budget is fixed, not scaled**: every rung sees the same 0.57B tokens, so the smaller models are progressively more overtrained relative to Chinchilla (21.5, 12.1 and 7.75 tokens per non-embedding parameter). The variable under test is parameter count, not token budget.

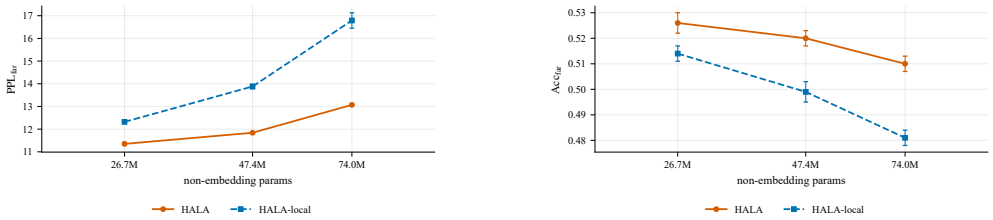

Figure 4: PPL_{far} (left) and top-1 accuracy on far AR positions (right) versus non-embedding parameters, on `books`, log-x: HALA against the local-only ablation, error bars over seeds. The gap between the two curves is the near/far dissociation this ladder tests; its exact paired value (mean, std, per-rung t) is in Table 7, not plotted separately. The rungs support the claim that the effect is *present* across this range; they do not establish how it scales.

G.4 LENGTH EXTRAPOLATION

Domain	Model	1k	4k	8k	32k
mixed	HALA	31.55 ± 0.10	30.69 ± 0.10	30.53 ± 0.10	30.43 ± 0.10
	Transformer	26.43 ± 0.20	105.95 ± 9.10	200.97 ± 10.11	422.45 ± 60.59
	Mamba-2	27.03 ± 0.25	$\mathbf{26.31 \pm 0.26}$	$\mathbf{26.19 \pm 0.26}$	$\mathbf{26.11 \pm 0.26}$
	Jamba	$\mathbf{24.17 \pm 0.03}$	28.83 ± 1.21	34.10 ± 0.75	45.51 ± 0.50
	Based	62.70	108.05	157.02	297.98
	HALA-local	33.82 ± 0.20	33.11 ± 0.20	32.99 ± 0.19	32.91 ± 0.19
	HALA-latent	42.18 ± 0.26	41.03 ± 0.27	40.84 ± 0.27	40.70 ± 0.27
	HALA-hard	31.31 ± 0.09	30.45 ± 0.11	30.30 ± 0.11	30.19 ± 0.12
	HALA-static	32.90 ± 0.17	32.03 ± 0.18	31.88 ± 0.18	31.78 ± 0.18
code	HALA	8.22 ± 0.03	8.00 ± 0.03	7.97 ± 0.03	7.94 ± 0.03
	Transformer	6.99 ± 0.02	52.56 ± 5.73	131.07 ± 10.62	311.54 ± 3.37
	Mamba-2	7.30 ± 0.06	$\mathbf{7.06 \pm 0.07}$	$\mathbf{7.03 \pm 0.07}$	$\mathbf{7.00 \pm 0.07}$
	Jamba	$\mathbf{6.66 \pm 0.02}$	9.38 ± 0.30	12.11 ± 0.51	17.49 ± 0.88
	HALA-local	8.60 ± 0.04	8.40 ± 0.04	8.37 ± 0.04	8.34 ± 0.04
	HALA-latent	10.84 ± 0.07	10.59 ± 0.08	10.55 ± 0.08	10.52 ± 0.08
	HALA-hard	8.19 ± 0.05	7.98 ± 0.05	7.94 ± 0.05	7.91 ± 0.05
	HALA-static	8.62 ± 0.05	8.42 ± 0.07	8.39 ± 0.07	8.36 ± 0.07
books	HALA	32.15 ± 0.50	31.38 ± 0.52	31.25 ± 0.53	31.15 ± 0.54
	Transformer	28.03 ± 0.10	206.35 ± 101.45	380.07 ± 189.47	521.16 ± 60.15
	Mamba-2	29.20 ± 0.40	$\mathbf{28.63 \pm 0.41}$	$\mathbf{28.54 \pm 0.41}$	$\mathbf{28.47 \pm 0.41}$
	Jamba	$\mathbf{27.65 \pm 0.34}$	64.69 ± 44.31	70.67 ± 49.63	68.26 ± 41.97
	HALA-local	33.11 ± 0.19	32.48 ± 0.19	32.38 ± 0.19	32.30 ± 0.19
	HALA-latent	40.47 ± 0.06	39.53 ± 0.08	39.38 ± 0.09	39.27 ± 0.09
	HALA-hard	32.12 ± 0.40	31.33 ± 0.41	31.20 ± 0.42	31.11 ± 0.42
	HALA-static	33.45 ± 0.06	32.65 ± 0.06	32.51 ± 0.06	32.41 ± 0.06

Table 8: Validation perplexity at each context length. Models were trained at a 1024-token context, so every column beyond it is extrapolation. Based is reported for `mixed` only ($n = 1$): no `code`/`books` checkpoints exist for Based, and evaluating those domains would require new training rather than an eval pass – real cost for a secondary baseline, not undertaken here. Consistent with Based's parameter-mismatch disclosure elsewhere in this paper.

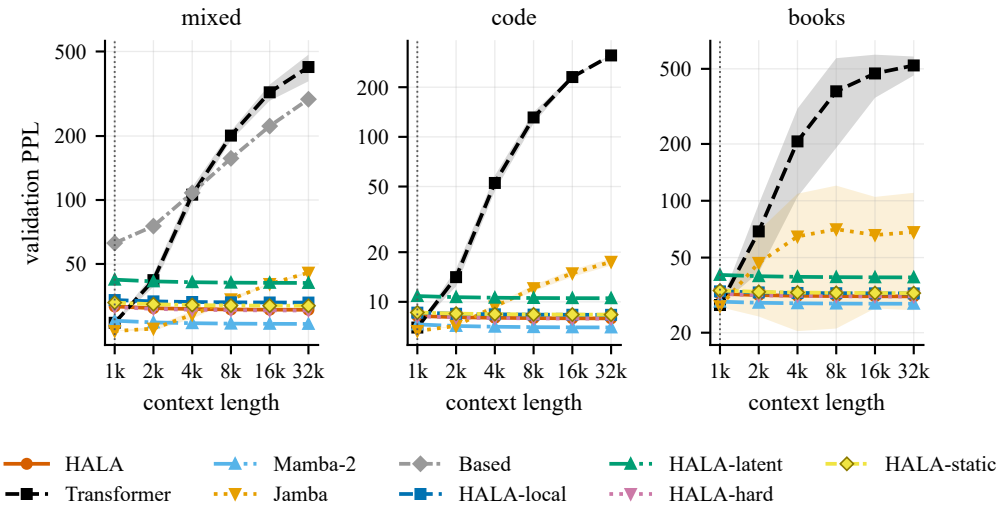

Figure 5: Perplexity versus context length, per domain. The dotted line marks the 1024-token training context.

G.5 ASSOCIATIVE RECALL

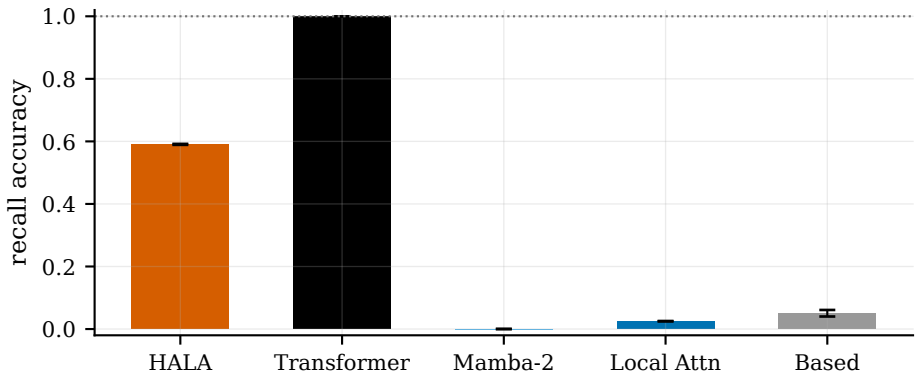

Figure 6: MQAR recall accuracy against competing architectures, matched parameters, one bar per model with seed error bars. HALA's routing ablations are not shown here; see Appendix A for why.

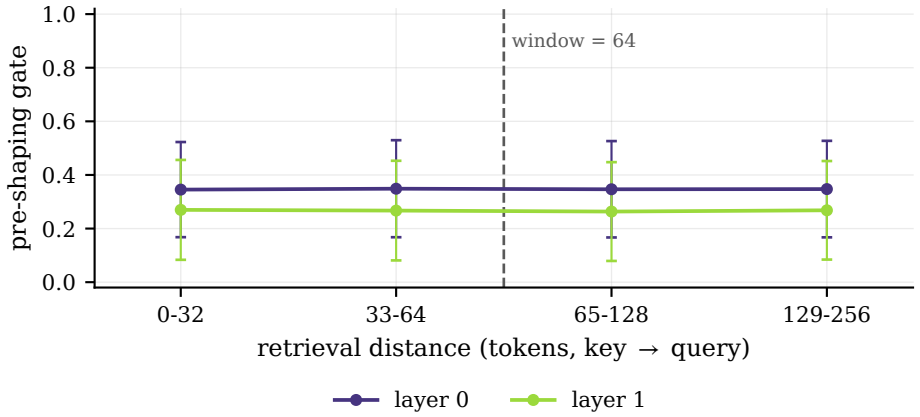

Figure 7: Pre-shaping gate against ground-truth retrieval distance on MQAR – the only measurement here where the distance from each key to its query is known exactly. The architecture predicts a step at the 64-token window boundary (dashed): inside it the local path can retrieve the key, outside it only the latent path can. **No such step appears.** No layer separates the two sides by more than its own head spread, so the gate is flat with respect to retrieval distance. That null is why the gate routes roughly 90% of signal into the latent path uniformly, and in turn why the MQAR routing ablation (not shown here; see Appendix A) reverses the real-text ablation ordering: on MQAR specifically the latent path is at chance.

G.6 EFFICIENCY

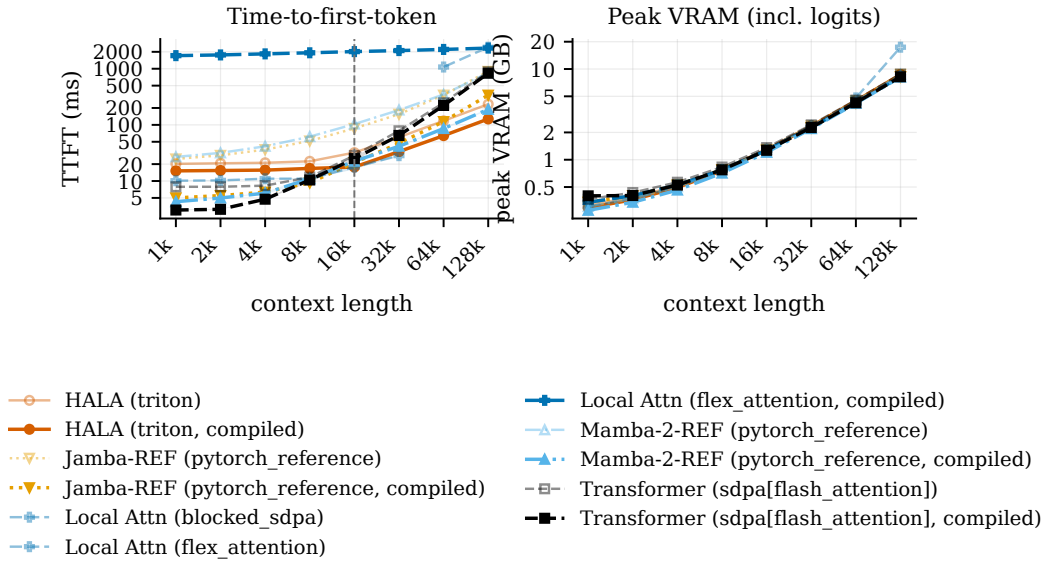

Figure 8: Time-to-first-token and peak VRAM versus context length, log–log, best available implementation per model. The kernel backend is named in the legend. The y-axis is labelled GB; values are GiB, matching Table 2 and the text.

G.7 DECAY-CEILING ABLATION

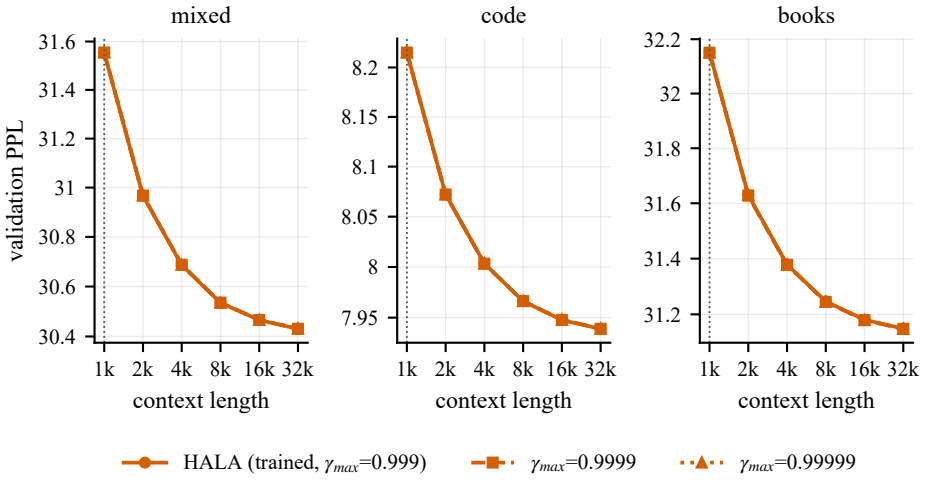

Figure 9: HALA's validation perplexity at longer context, the trained checkpoint re-evaluated at decay ceiling $\gamma_{max} = 0.99999$ instead of its trained 0.999, per domain. Same weights throughout – only the decay squash range read at evaluation time differs, since the decay is produced by a projection through a sigmoid and a wider squash range is a valid reinterpretation of the same weights. This measures headroom in the parameterisation on an already-trained checkpoint, not what a model trained at a higher ceiling would do. A $\gamma_{max} = 0.9999$ intermediate point was not run in this pass.

G.8 PARAMETERS

Per layer, with $m = \max(d/4, 32)$:

Module	Shape	Parameters
Fused QKV	$d \to 4d + L + H$	$d(4d + L + H)$
KV up-projection	$L \to 2d$	$2Ld$
Output projection	$d \to d$	$d^2 + d$
Block norm	–	$2d$
Recurrence norm	–	$2d_k$
Gate MLP	$d \to m \to H$	$m(d + 1) + H(m + 1)$

Table 9: Per-layer parameter count by module. Rotary tables and per-head scales are buffers and cost nothing. Because HALA has no FFN, no closed-form parity condition against FFN-bearing baselines exists; we match parameters empirically instead (Section 4).

G.9 PARAMETER-PARITY SOLVE

Table 10 lists the knob solved per baseline, its solved value, and the resulting error against HALA's 11,852,352 non-embedding parameters.

Model	Knob	Solved value	Error
HALA	–	reference	–
Transformer	FFN hidden	420 ($0.82d$)	0.12%
Local attention	FFN hidden	420	0.12%
Mamba-2	d_{inner}	864 (expand 1.688), head dim 32	0.16%
Jamba	FFN hidden	108	0.08%
Based	kv heads	8 (published feature dim 16)	−55.7%

Table 10: Parameter-parity solve per baseline, matched against HALA's 11,852,352 non-embedding parameters.