TOWARDS LONG-TERM MEMORY WITH SMALL LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

As personal AI assistants proliferate, they are expected to become capable of complex, long-horizon interactions while preserving the privacy of increasingly personal information. A central challenge in achieving this form of personalized intelligence is long-term memory. An assistant cannot genuinely understand a person from isolated sessions, it needs a continuous thread of experience that accumulates over time, where someone has been, whom they have spoken to, what they have been working on, and the context connecting these experiences. This requirement for persistent memory is tightly coupled to privacy. Rather than transmitting an ever growing record of personal experience to a remote server, data center, or other external system, memory can remain private by being stored and processed locally on the device where the assistant operates. Such a design, however, places a fundamental constraint on the underlying model, mainly the assistant's memory must be small enough to run on a phone, a pair of glasses, or a robot while retaining the capacity to maintain and recall over a persistent stream of experience. Yet every long-term memory result we are aware of is reported at model scales an order of magnitude beyond that budget, and whether a small model can carry such an ability at all is, to the best of our knowledge, underexplored. In this paper we propose an approach to long-term memory built purely on small language models, in which memory lives in intrinsic recurrent state rather than in a growing attention window. We show that small models combined with linear attention and the right training strategy offer strong recall, and that because a bounded state has no key-value cache to grow, such a model can ingest conversations far longer than a large transformer can admit at all. We evaluate against open-source counterparts such as Gemma-3-27B-it, Qwen2.5-32B-Instruct and Qwen3-30B-A3B, on the BEAM long-term memory benchmark and find our approach competitive with models an order of magnitude larger.

1 INTRODUCTION

Personal assistants are increasingly expected to support activities that unfold across a person's life: booking travel, managing a calendar, filing taxes, noticing a health trend, or resuming a project weeks after it was last discussed. What makes such an assistant useful is not only fluency within a conversation, but a persistent understanding of the person's habits, preferences, ongoing projects, and past interactions. That knowledge must survive across sessions.

Today, long-term memory is typically maintained on remote infrastructure: interactions are logged, a retrieval system selects relevant records, and the model consumes them as context. That arrangement becomes a poor fit as assistants migrate from remote services into phones, glasses, and other wearables: each recall that still reaches back to the cloud crosses the network, increasing latency and exposing a growing personal record to infrastructure outside the user's control. Co-locating memory with inference keeps recall local and personal data on the device, but forces both the model and its memory mechanism into a far smaller power, memory, and compute regime.

Yet, to our knowledge, long-term conversational memory has been evaluated primarily at 27B parameters or above; BEAM, for example, reports results for Qwen2.5-32B, Llama-4-Maverick, and Gemini-class models (Tavakoli et al., 2026). The small-model regime, where personal assistants can actually run on-device, remains largely unmeasured. Long-term memory may lie beyond the capac-

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.

ity of small models, or their failures may expose specific limitations that better memory mechanisms or training can address. We distinguish these possibilities by measuring memory at small-model scale, designing a memory mechanism for that regime, and evaluating it on benchmarks that require genuine long-term memory rather than short-context recall. Gated linear attention is a natural fit for this regime because it stores the past in a fixed-size recurrent state, avoiding both a growing context window and an external index (§2). Its long-horizon recall depends on whether that state can recover a fact written forty thousand tokens earlier.

Contributions. We recast bounded-state recall as *sparse recovery rather than lookup*: unrolling a multi-timescale layer shows its M outputs to be M evaluations of one *memory polynomial* whose coefficients are the relevance of each past token (§5), a framing we are aware of no counterpart to in a multi-timescale literature that seeks approximate reconstruction of a general history rather than exact recovery of a few items. Holding backbone, data and compute fixed and varying only the *shape* of the state, an arm with 64 heads beats one holding twice the state across half as many timescales by 30% and 31% at 500K and 1M context lengths. We measure long-term conversational memory at **on-device scale**, a regime we believe unreported, since BEAM and its predecessors evaluate at 27B parameters and above (Tavakoli et al., 2026): our 2.9B recurrent checkpoint is more precise than 27–32B baselines at every corpus length and wins information extraction at all four lengths out to 10M. Finally, recall can be trained into the state : reinforcement learning against a verifiable recall reward lifts the fact-locating abilities: contradiction resolution $0.025 \rightarrow 0.212$, information extraction $0.133 \rightarrow 0.324$ at 100K.

2 Background

2.1 Why gated linear attention

Reprocessing a year of conversation at every turn makes the cost of inference scale with everything that has ever happened, rather than with what matters now. Nor can memory be reduced to an external retrieval system attached to a model that does not remember: the retrieval layer brings its own storage, latency, and failure modes, while the model itself learns nothing about how to retain information. A practical on-device assistant therefore needs memory that is persistent, selective, and internal to the model's operation. Gated linear attention is the one widely used family whose memory is *intrinsic state*: a fixed-size matrix $\boldsymbol{S}_t$ carried forward and updated in place, with read cost independent of how much history it summarises. That is the right shape for a device, and it has a second consequence that matters for evaluation, with no key-value cache to grow, such a model can be shown conversations that a transformer of any size cannot admit at all. Its weakness is equally clear, and is why the family is not already the default: a bounded state is a lossy state, especially over long contexts, retaining the right information becomes increasingly difficult.

2.2 Gated linear attention as a recurrent associative memory

We study Gated DeltaNet (GDN) layers, a GDN layer maintains a matrix state $\boldsymbol{S}_t \in \mathbb{R}^{d_k \times d_v}$, updated at each token by

$$\boldsymbol{S}_t = \Big(\mathrm{Diag}(\boldsymbol{\alpha}_t) - \beta_t \boldsymbol{k}_t^\top \boldsymbol{k}_t\Big)\boldsymbol{S}_{t-1} \;+\; \beta_t \boldsymbol{k}_t^\top \boldsymbol{t}_t, \qquad \boldsymbol{o}_t = \boldsymbol{q}_t \boldsymbol{S}_t. \tag{1}$$

The bracket transforms what is already stored: $\mathrm{Diag}(\boldsymbol{\alpha}_t)$ is the *gate*, which shrinks every stored trace, and $-\beta_t \boldsymbol{k}_t^\top \boldsymbol{k}_t$ is the *delta term*, which erases the value previously held at the address $\boldsymbol{k}_t$ before the new one is written. The final term is the write itself, an outer product of the current key and value. Reading is a single multiplication by the query.

The design problem, and three limitations. A model reads a hundred thousand tokens of conversation and is then asked about something stated once, forty thousand tokens ago. Softmax attention answers by keeping everything exact, at $\Theta(T)$ memory and $\Theta(T^2)$ compute. A gated linear attention keeps instead a fixed-size associative memory $\boldsymbol{S}_t \in \mathbb{R}^{d_k \times d_v}$ at $\Theta(d_k d_v)$ memory and $\Theta(T)$ compute, independent of context length. The difficulty appears the moment the recurrence is unrolled: with a scalar retention $\bar{\alpha}$, a write made Δ tokens ago survives with weight $\bar{\alpha}^{\Delta} = e^{-\Delta/\tau}$ (§2.3), so **a single state has exactly one exponential memory kernel, with exactly one time constant.** Three limitations follow, and they are not the same limitation.

Extinction. The kernel is exponential, so a trace is not merely weakened with depth, it is extinguished. Beyond a few multiples of τ the contribution falls below the interference floor and nothing downstream can recover it. The memory has a horizon, and it is short.

No resolution across timescales. Worse than the horizon is that its position cannot be chosen to suit the task. A τ long enough to retain an early fact is also long enough to retain every intervening distractor; one short enough to discard the distractors discards the fact as well.

Capacity. Even inside the horizon the state is a sum of outer products, so a probe returns its target plus $\sum_{i\neq j}\langle \boldsymbol{k}_i, \boldsymbol{k}_j\rangle \boldsymbol{t}_i$. Interference grows with the number of stored items and with key coherence, and $\mathrm{rank}(\boldsymbol{S}) \leq d_k$ caps the number of cleanly separable traces (Appendix E).

The first two are properties of *time*; the third is a property of the *key space*. The literature on linear attention has examined the third closely, it is the standard explanation for weak in-context retrieval, and it motivates larger head dimensions and the delta rule's erase and has had much less to say about the first two. Yet no amount of d_k buys a second time constant, which is what §2.3 and §3 take up.

2.3 A SINGLE STATE CAN ONLY BE TUNED TO ONE DEPTH

Set the delta term aside for a moment. The gate $\boldsymbol{\alpha}_t \in \mathbb{R}^{d_k}$ enters as $\mathrm{Diag}(\boldsymbol{\alpha}_t)$, acting on the left, so row i of the state evolves entirely on its own:

$$\boldsymbol{S}_t[i,:] = \alpha_{t,i}\, \boldsymbol{S}_{t-1}[i,:] + \beta_t\, k_{t,i}\, \boldsymbol{t}_t. \tag{2}$$

Without the delta term the state is therefore d_k independent scalar recursions, one per key coordinate, each with its own retention. We fix one such coordinate and write $\bar{\alpha}$ for its retention, assumed constant in t. Nothing is approximated by working coordinate-wise; the assumption is the constancy in t, to which we return below. Writing the recursion for that coordinate as $\boldsymbol{S}_t = \bar{\alpha}\, \boldsymbol{S}_{t-1} + \beta_t \boldsymbol{k}_t^\top \boldsymbol{t}_t$, we put it in its final form in two steps.

Step 1 — unroll, indexed by absolute position. Iterating from $\boldsymbol{S}_0 = 0$, the write made at position u is multiplied by one factor of $\bar{\alpha}$ for each of the $t-u$ tokens seen since:

$$\boldsymbol{S}_t = \sum_{\substack{u=1 \\ \text{(oldest write)}}}^{\substack{t \\ \text{(newest write)}}} \bar{\alpha}^{\,t-u}\, \beta_u\, \boldsymbol{k}_u^\top \boldsymbol{t}_u. \tag{3}$$

Step 2 — re-index by depth. Substitute $\Delta = t - u$, so that $u = t - \Delta$ and the summation runs backwards from the current token: $u = t$ becomes $\Delta = 0$ (the most recent write) and $u = 1$ becomes $\Delta = t - 1$ (the oldest). Then

$$\boldsymbol{S}_t = \sum_{\substack{\Delta=0 \\ \text{(current token)}}}^{\substack{t-1 \\ \text{(start of context)}}} \bar{\alpha}^{\,\Delta}\, \beta_{t-\Delta}\, \boldsymbol{k}_{t-\Delta}^\top \boldsymbol{t}_{t-\Delta}. \tag{4}$$

This is the form used throughout: the exponent is now exactly the depth of the write, and Δ indexes how far back we are looking. A write at depth Δ has surviving weight

$$\bar{\alpha}^{\Delta} = e^{-\Delta/\tau}, \qquad \tau = -1/\log\bar{\alpha}. \tag{5}$$

We use $\bar{\alpha}$ throughout for the per-token retention and τ for the timescale it implies. **A single state therefore has one exponential memory kernel, with one time constant τ.** We now show that this forces a choice which cannot be made well.

The dilemma, quantified. A read does not see the target write alone; it sees every write still retained. Take the target at depth Δ and treat the remaining writes as a background of comparable magnitude, whose factor μ scales signal and background alike and so drops out of the ratio below. The retained background mass is

$$B(\tau) = \sum_{\Delta'=0}^{\infty} e^{-\Delta'/\tau} = \frac{1}{1-e^{-1/\tau}} \;\approx\; \tau \qquad (\tau \gg 1), \tag{6}$$

so the target, which itself survives with weight only $e^{-\Delta/\tau}$, is read against a background of total weight $\approx \tau$. What a downstream reader has to work with is the quotient of the two,

$$R(\tau) \;=\; \frac{e^{-\Delta/\tau}}{\tau} \tag{7}$$

which we call the *signal-to-background ratio*: how far the target stands out from the rest of what the state retains. Larger R is a cleaner read; R near zero means the target is buried.

Proposition 1 (Single-state dilemma)**.** *$R(\tau) = e^{-\Delta/\tau}/\tau$ is maximised at $\tau = \Delta$, where $R_{\max} = 1/(e\Delta)$, and for a retention tuned to Δ_0 but a target at Δ,*

$$\frac{R(\Delta_0)}{R_{\max}(\Delta)} = \frac{\Delta}{\Delta_0}\, e^{\,1-\Delta/\Delta_0}. \tag{8}$$

Proof. $\log R = -\Delta/\tau - \log\tau$ has derivative $\Delta/\tau^2 - 1/\tau$, vanishing only at $\tau = \Delta$, where the second derivative is $-1/\Delta^2 < 0$; substituting gives $R_{\max}$, and equation 8 is $R(\Delta_0) = e^{-\Delta/\Delta_0}/\Delta_0$ divided by it. □

Two consequences make the case for the architecture. First, the optimum is a *point*: $\tau = \Delta$ serves one depth and its quality $1/(e\Delta)$ already decays with it, so a state cannot be tuned for a conversation whose evidence is scattered, what it should optimise depends on where the evidence happens to lie. Second, and decisively for what follows, equation 8 depends on the *ratio* Δ/Δ_0 and not the difference: a retention an order of magnitude too long costs about a factor of ten, one an order of magnitude too short costs e^{-10}. Mistuning is forgiving upward and catastrophic downward, which is why a lone state drifts toward long retentions and a diffuse read. A single exponential memory therefore cannot give sharp, selective access at many depths at once.

What the fix must look like. Because the penalty equation 8 depends on the *ratio* of a retention to the depth being read, a head with timescale τ serves a band of depths whose width in tokens scales with τ but whose width *on a logarithmic axis* is the same for every head. Covering a range of depths without gaps is then a tiling problem with equal tiles, and equal tiles must be laid at equal spacing, constant *ratio* in τ. Two things follow, and §3 makes both precise: the ratio between neighbouring timescales fixes the quality of coverage, and the number of heads required grows with the *logarithm* of the range to be covered, which is what lets a handful of heads span five orders of magnitude. Appendix A.5 describes the argument.

3 TEMPORAL HEADS

The preceding section §2.3 leaves a concrete specification: several retentions, spaced by constant ratio, enough of them to cover the depths of interest. Temporal heads are that specification. We fix M retentions on a geometric grid,

$$\tau_m = \tau_1 \rho^{\,m-1}, \quad \rho > 1, \qquad \bar{\alpha}_m = e^{-1/\tau_m}, \qquad m = 1, \dots, M, \tag{9}$$

and give head m its own state evolving as

$$\boldsymbol{S}_t^{(m)} = \Big(\mathrm{Diag}(\bar{\alpha}_m \boldsymbol{g}_t) - \beta_t^{(m)} \boldsymbol{k}_t^\top \boldsymbol{k}_t \Big) \boldsymbol{S}_{t-1}^{(m)} + \beta_t^{(m)} \boldsymbol{k}_t^\top \boldsymbol{t}_t. \tag{10}$$

We refer to equation 9 itself, and to the M heads that occupy it, as *the grid*. The name is the claim: what matters about the collection is not that there are M states but *where their retentions sit*, and §5 makes that literal, the $\bar{\alpha}_m$ turn out to be evaluation points, and two heads sharing a retention are worth one. Each head is a state of the kind analysed in §2.3 and inherits its single kernel; what changes is that a read now sees M of them at once, and by Proposition 1 the head nearest in *ratio* to a given depth is the one that serves it well. With ρ fixed, the grid guarantees that some head is always within a factor $\sqrt{\rho}$ of any depth in $[\tau_1, \tau_M]$, so the worst-case mistuning penalty equation 8 is bounded across the whole band rather than at one point. The span is $\tau_M/\tau_1 = \rho^{M-1}$: range and resolution trade against each other through ρ at fixed M.

The effective memory kernel of the grid. It is worth making precise what the M states buy, since the layer's output is a single vector and the heads are not separately visible outside it. Write the output projection block-wise over heads, $W_O = [\,W_O^{(1)} \;\cdots\; W_O^{(M)}\,]$, so that the layer emits $y = \sum_m W_O^{(m)} \boldsymbol{o}^{(m)}$. Substituting equation 13,

$$y = \sum_{\Delta=0}^{T-1} w_\Delta \, \mathcal{K}(\Delta)\, \boldsymbol{u}_\Delta, \qquad \mathcal{K}(\Delta) := \sum_{m=1}^{M} e^{-\Delta/\tau_m} \, W_O^{(m)}, \tag{11}$$

so the grid acts on the past through a single matrix-valued *memory kernel* $\mathcal{K}$.

To speak of this kernel as a function, whether it decays monotonically, how often it changes sign, we need a scalar, since those are not properties of a matrix. The natural scalar is the one a consumer of this layer actually sees. A downstream unit reads a single linear functional $\langle \boldsymbol{r}, y \rangle$ of the output, where $\boldsymbol{r} \in \mathbb{R}^{d_{\text{model}}}$ is a row of the next layer's weight matrix; and a write contributes to that unit through the direction $\hat{\boldsymbol{u}}$ its value points along. For that pair,

$$\langle \boldsymbol{r}, \mathcal{K}(\Delta)\hat{\boldsymbol{u}} \rangle = \sum_{m=1}^{M} c_m \, e^{-\Delta/\tau_m} =: k(\Delta), \qquad c_m = \langle \boldsymbol{r}, W_O^{(m)} \hat{\boldsymbol{u}} \rangle \in \mathbb{R}. \tag{12}$$

So $\boldsymbol{r}$ is not an analytical device: each choice of $\boldsymbol{r}$ is a different downstream reader, and k is the memory kernel that reader experiences. The structure worth noticing is that the *rates* τ_m are the same in equation 12 for every $\boldsymbol{r}$ and $\hat{\boldsymbol{u}}$, while the coefficients c_m are not. The grid supplies a fixed basis of M timescales, shared across the whole layer; each reader chooses its own weighting of that basis, and the weights are learned and *signed*. In our implementation $\bar{\alpha}_m$ enters through a per-head parameter A initialised as $A_{\log} = \text{linspace}(\log 1, \log 16, M)$.

4 Results

All numbers are BEAM judged scores from `gpt-4.1-mini` at temperature 0, averaged over the nine abilities other than abstention, which is excluded throughout because a truncated model passes it by default (Appendix F). Three results follow. *Shape beats size.* Three arms share one 2.9B backbone, one datamix and 129,375 steps and differ only in head count h and per-head width d_k, which fix the state per layer at $h \times d_k^2$; the arm with most heads (th64, 1.05M per layer) beats the one holding twice the state across half as many timescales (th32/d256, 2.10M) by 30% and 31% at 500K and 1M, winning 8 of 9 and 7 of 9 abilities, and alone holds its score as the corpus grows, losing 3% from 100K to 1M where the dense baseline loses 24% (Figure 1). Information extraction, the ability that most directly asks the model to locate one fact at one depth, rises monotonically with head count at both long lengths. At 100K, where the corpus sits close enough to the query that mistuning costs little, no such trend exists, which is what the coverage argument predicts. *Post-training redistributes recall rather than raising it.* Adding supervised fine-tuning and 160 steps of RL under a reward that pays only for the gold string appearing somewhere in the response (Appendix C) lifts every ability that turns on locating a specific fact — contradiction resolution $0.025 \rightarrow 0.212$ at 100K, information extraction $0.133 \rightarrow 0.324$, knowledge update roughly doubling at 100K and 500K — while preference following and summarization fall at nearly every length, for a nine-ability net of $+0.032$, $+0.027$, -0.003 and $+0.042$ (Figure 2). *Against much larger baselines the failure is recall, not addressing.* Our 2.9B model does not lead Table 1 on average, but it wins information extraction at all four corpus lengths (0.324, 0.189, 0.197, 0.100 against best baselines of 0.195, 0.120, 0.108, 0.050), event ordering from 500K upward and knowledge update at 500K and 1M, the three abilities that ask where in the history a fact sits, while losing those scored on the whole response. Its precision exceeds every baseline at every length (Appendix H), and the average gap closes with length: 0.095 behind Gemma-3-27B at 100K, ahead of it at 10M (0.102 against 0.092) with four of nine abilities outright. Appendix G repeats the comparison with the Qwen windows YaRN-extended; the ordering does not change.

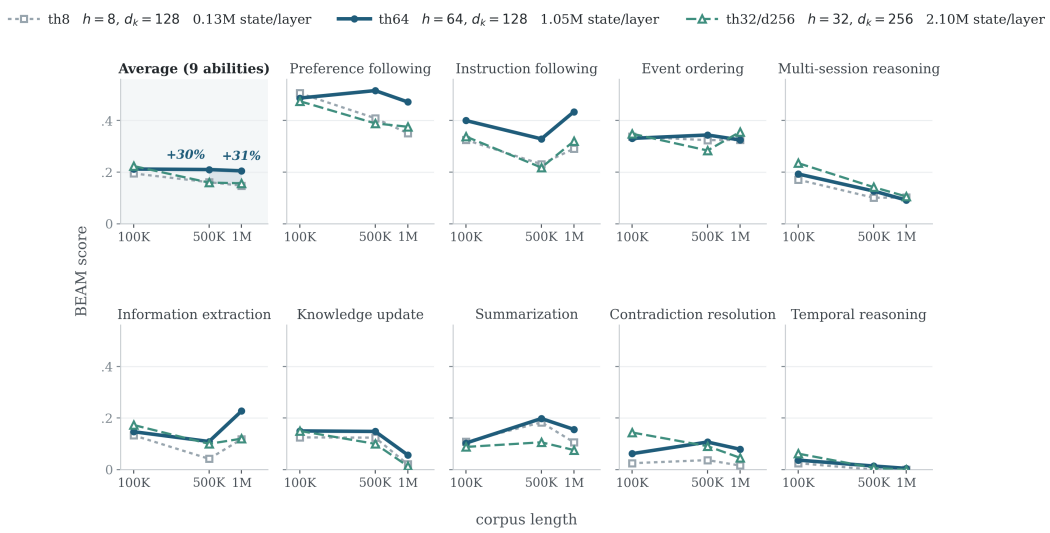


Figure 1: Recurrent-state shape across pretraining arms, by ability and corpus length. All three arms share one 2.9B backbone, one datamix and 129,375 steps, and differ only in head count h and per-head width d_k; state per layer is $h \times d_k^2$. Scores are BEAM judged scores over the nine abilities other than abstention (Appendix F). One panel per ability, ordered by mean score at 100K, with the nine-ability average shaded at the left. **More state is not what helps.** At 100K the ranking follows state size and th64 is last by 5%; at 500K and 1M it inverts, th64 (1.05M per layer) leading by 30% and 31% over the better of the other two and **beating th32/d256 (**2.10**M per layer) while holding half as much state**. The margin is broad, not an average dragged by one column: best of the three on 8 of 9 abilities at 500K and 7 of 9 at 1M, against 2 of 9 at 100K. th64 also loses only 3% from 100K to 1M where th8 loses 24%.

5 Why the grid: recall as sparse recovery

5.1 The grid evaluates the memory polynomial

Specialise equation 10 to a scalar retention and read with $\boldsymbol{q}$. Writing $\Delta = t - u$ for depth, and separating the scalar relevance from the value,

$$\boldsymbol{o}^{(m)} = \sum_{\Delta=0}^{T-1} w_\Delta \, \boldsymbol{u}_\Delta \, \bar{\alpha}_m^\Delta, \qquad w_\Delta := \beta_{t-\Delta} \, \langle \boldsymbol{k}_{t-\Delta}, \boldsymbol{q} \rangle \in \mathbb{R}, \quad \boldsymbol{u}_\Delta := \boldsymbol{t}_{t-\Delta}. \tag{13}$$

The scalar sequence (w_Δ) is the *relevance profile* of the context under this probe: how much the token Δ steps back bears on the question. A head's reading $\boldsymbol{o}^{(m)}$ is a vector, and what follows is about polynomials, so we work one component at a time: fix any $\boldsymbol{e} \in \mathbb{R}^{d_v}$ in the head's *value* space, distinct from the readout direction $\boldsymbol{r} \in \mathbb{R}^{d_{\text{model}}}$ of equation 12, which lives downstream of the output projection and set $c_\Delta := w_\Delta \langle \boldsymbol{e}, \boldsymbol{u}_\Delta \rangle$. Then

$$\varphi_m := \langle \boldsymbol{e}, \boldsymbol{o}^{(m)} \rangle = \sum_{\Delta=0}^{T-1} c_\Delta \, \bar{\alpha}_m^\Delta = P(\bar{\alpha}_m), \qquad P(z) := \sum_\Delta c_\Delta z^\Delta. \tag{14}$$

We call P the *memory polynomial* of the probe $\boldsymbol{q}$. Its degree is $T - 1$, one less than the context length, and its coefficient at z^Δ is the relevance of the token Δ steps back; everything the layer can report about the past, under this probe and along this component, is a property of P.

It is worth being exact about what P belongs to. It is *not* a property of the stored state alone: c_Δ carries the key–query match $\langle \boldsymbol{k}_{t-\Delta}, \boldsymbol{q} \rangle$, so a different question presents a different memory polynomial. The grid is fixed by the architecture; the polynomial is composed afresh at every probe.

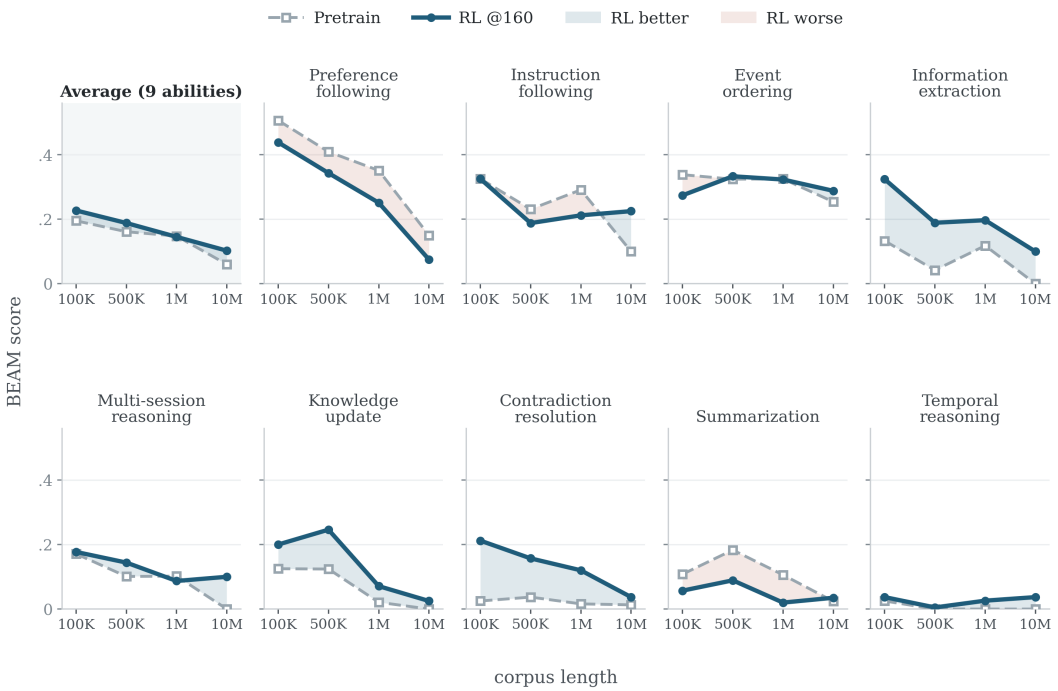


Figure 2: Pretraining against post-training, per ability. Both arms are the same 2.9B Gated DeltaNet checkpoint at step 129,375; the post-trained arm adds supervised fine-tuning and 160 steps of single-turn RL under the recall reward of Appendix C. Shading marks the gap and its sign. Abstention is excluded (Appendix F); abilities are ordered by mean score at 100K. The abilities that require locating a specific fact, contradiction resolution, information extraction, knowledge update, improve at every length, while those scored on the whole response, preference following, summarization, fall at nearly every length. On the nine-ability average the net is $+0.032$, $+0.027$, -0.003 and $+0.042$.

5.2 The readings are samples of a transform

Equation equation 14 can be read in two equivalent ways, and the second is the one that connects this construction to an existing model of memory.

As a generating function. The memory polynomial $P(z) = \sum_{\Delta} c_{\Delta} z^{\Delta}$ is the ordinary generating function of the relevance profile, and head m reports its value at the single point $z = \bar{\alpha}_m$.

As a transform. Substituting $\bar{\alpha}_m = e^{-1/\tau_m}$ and writing $s_m := 1/\tau_m$ for the head's *rate*,

$$\varphi_m = \sum_{\Delta=0}^{T-1} c_{\Delta}\, e^{-s_m \Delta} =: \hat{c}(s_m). \tag{15}$$

The map $\hat{c}(s) = \sum_{\Delta} c_{\Delta} e^{-s\Delta}$ is the *discrete Laplace transform* of the relevance profile — a Dirichlet series, and the same object as P under the change of variable $z = e^{-s}$. Its continuous-time counterpart is the familiar $\hat{c}(s) = \int_0^{\infty} c(\Delta) e^{-s\Delta}\, d\Delta$; we have a sum because depth is counted in tokens.

So the grid does not summarise the past and it does not search it. It **samples one transform of it, at M points $s_1, \ldots, s_M$ chosen by the architecture**. Reading out is inverting that transform from those samples.

Table 1: BEAM at native context windows, by corpus length and ability. Judged by `gpt-4.1-mini` at temperature 0 under the benchmark's own prompt. The abstention ability is reported in Appendix F: under truncation, "the answer is not here" remains correct without exercising any memory, so it inflates short-window scores rather than measuring memory. All averages are over the nine remaining abilities. Appendix G repeats the comparison with the Qwen windows YaRN-extended.

Length	Ability	Ours (RL, single-turn)	Gemma-3 27B-it	Qwen2.5 32B	Qwen3 30B-A3B
100K	Contradiction resolution	0.212	0.297	**0.506**	0.150
	Event ordering	0.274	0.228	0.184	**0.289**
	Information extraction	**0.324**	0.195	0.103	0.076
	Instruction following	0.325	0.500	**0.512**	0.406
	Knowledge update	0.200	**0.300**	0.125	0.125
	Multi-session reasoning	0.177	**0.416**	0.210	0.184
	Preference following	0.438	**0.675**	0.656	0.394
	Summarization	0.057	**0.181**	0.130	0.076
	Temporal reasoning	0.037	**0.106**	0.037	0.013
	Average	0.227	**0.322**	0.274	0.190
500K	Contradiction resolution	0.157	0.327	**0.396**	0.148
	Event ordering	**0.333**	0.233	0.222	0.267
	Information extraction	**0.189**	0.101	0.080	0.120
	Instruction following	0.188	**0.385**	0.354	0.274
	Knowledge update	**0.246**	0.093	0.107	0.138
	Multi-session reasoning	0.144	**0.164**	0.124	0.102
	Preference following	0.343	0.584	**0.637**	0.422
	Summarization	0.089	**0.254**	0.179	0.131
	Temporal reasoning	0.005	0.081	0.060	**0.102**
	Average	0.188	**0.247**	0.240	0.189
1M	Contradiction resolution	0.120	0.275	**0.411**	0.138
	Event ordering	**0.323**	0.212	0.193	0.257
	Information extraction	**0.197**	0.102	0.079	0.108
	Instruction following	0.212	0.349	**0.432**	0.286
	Knowledge update	**0.071**	0.029	0.000	0.000
	Multi-session reasoning	0.087	**0.213**	0.193	0.108
	Preference following	0.251	0.464	**0.608**	0.342
	Summarization	0.020	**0.162**	0.091	0.072
	Temporal reasoning	0.026	**0.033**	0.029	0.021
	Average	0.145	0.204	**0.226**	0.148
10M	Contradiction resolution	0.037	0.287	**0.412**	0.237
	Event ordering	**0.288**	0.164	0.209	0.244
	Information extraction	**0.100**	0.013	0.000	0.050
	Instruction following	0.225	0.237	**0.375**	0.250
	Knowledge update	0.025	0.025	**0.075**	0.025
	Multi-session reasoning	**0.100**	0.000	0.000	0.005
	Preference following	0.075	0.050	**0.354**	0.142
	Summarization	0.035	0.056	0.048	**0.114**
	Temporal reasoning	**0.037**	0.000	0.000	0.000
	Average	0.102	0.092	**0.164**	0.119

6 Related work

Scale-invariant temporal memory. Shankar and Howard Shankar & Howard (2012); **?**; 2013); Pastalkova et al. (2008); Eichenbaum (2014); MacDonald et al. (2011) represent time with a bank of leaky integrators at logarithmically spaced rates, holding the Laplace transform of the stimulus history; a finite-difference approximation to the Post–Widder inverse across neighbouring rates yields time cells whose blur grows with their latency, so resolution is scale-invariant. The line continues through unified treatments of time, space and sequence Howard et al. (2014); Kraus et al. (2013); Tsao et al. (2018), SITH and DeepSITH Shankar & Howard (2018); Jacques et al. (2021); Tiganj et al. (2015); Voelker et al. (2019); Voelker & Eliasmith (2018), and rescaling-invariant CNNs Jacques et al. (2022); HiPPO Gu et al. (2020; 2021); Orvieto et al. (2023) and S4 Gu et al. (2022); Gupta et al. (2022); Smith et al. (2023) arrive nearby from approximation theory. Our grid is the discrete counterpart of their first stage — head m holds $\hat{c}(s_m)$, equation 15 — and spaces logarithmically for the same underlying reason Koutnik et al. (2014); Neil et al. (2016); Bai et al. (2018), though we reach it from the mistuning ratio equation 8 rather than from finite-difference accuracy. Three differences set the claims apart: our signal is query-dependent rather than fixed history, since c_Δ carries the key match; we assume sparsity (Appendix A.3) and recover finitely many items exactly, obtaining a count where they obtain a resolution; and we ask what *any* read-out could extract

rather than fixing one. A structural echo is worth recording: their inversion is a signed combination across neighbouring rates, and the output projection W_O of equation 11 is exactly a learned signed combination across heads. Fixed-size state replaces the growing cache: Mamba makes transitions input-selective Dao & Gu (2024), Gated Linear Attention adds learned decay Yang et al. (2024b); Katharopoulos et al. (2020), DeltaNet overwrites an addressed association before writing Yang et al. (2024a); Schlag et al. (2021), Gated DeltaNet combines targeted revision with broad forgetting Yang et al. (2025), and the line continues in Mamba-3 Lahoti et al. (2026); Fu et al. (2023); Lieber et al. (2024); Glorioso et al. (2024) and Gated DeltaNet-2 Hatamizadeh et al. (2026); Beck et al. (2024); Peng et al. (2023); Sun et al. (2023); Ma et al. (2023). All improve what is written, erased and retained; none says how a limited set of modes should *cover time*. That is the axis we isolate, and the limit it sets is distinct from one set by state rank or key coherence.

Memory beyond the attention window. Transformer-XL reuses hidden states across segments Dai et al. (2019); Rae et al. (2020); Bulatov et al. (2022); Hutchins et al. (2022); Bulatov et al. (2023); Munkhdalai et al. (2024); Chevalier et al. (2023), Memorizing Transformers retrieve from an external store Wu et al. (2022); Weston et al. (2015); Graves et al. (2016); Packer et al. (2023), test-time training treats a learned model as the hidden state Sun et al. (2024); Gandelsman et al. (2022), and Titans learns a long-term memory during inference Behrouz et al. (2025). Our setting is stricter: what survives in bounded intrinsic state whose update cost does not grow with elapsed history Xiao et al. (2024).

Recurrent models at device scale, and how they are evaluated. RecurrentGemma De et al. (2024b;a) and RWKV-edge Choe et al. (2024) show the budget is workable, but neither organises state as an explicit geometric spectrum nor measures persistent recall from it. On the evaluation side, LoCoMo introduced long conversational histories Maharana et al. (2024); Zhong et al. (2024); Xu et al. (2022), LongMemEval separated extraction, multi-session and temporal reasoning, knowledge update and abstention Wu et al. (2025), BEAM — used here — extends past a million tokens Tavakoli et al. (2026), and MemGAS Xu et al. (2026); Bai et al. (2024a;b); Hsieh et al. (2024); Zhang et al. (2024) and LoCoMo-Plus loc (2026); Kuratov et al. (2024); Yen et al. (2025) broaden the target. Together they separate the question we ask from the one retrieval answers: holding a fact in bounded state is not the same problem as fetching it from a store Lewis et al. (2020); Borgeaud et al. (2022); Khandelwal et al. (2020).

7 Conclusion

Recall from a gated linear-attention state is not just a lookup that a bigger state would make more reliable. It can also be viewed as sparse recovery from M evaluations of one polynomial, and the resource that binds is M, how many distinct decay timescales a layer carries, not d_k. It predicts that state spent on more timescales should outperform the same state spent on wider heads, and it does, by 30% and 31% at 500K and 1M against an arm holding twice as much. It also says where the failure should show, and at on-device scale it shows there: a 2.9B recurrent model reads more precisely than 27–32B transformers. The practical consequence is that the cheapest lever on long-horizon memory is not just more state but a better-spread spectrum of retentions, a change to head count and initialisation. What we have not done is separate spacing from count: our arms vary how many timescales a layer holds, not how they are arranged.

Ethics statement

We have read the ICLR Code of Ethics and this work adheres to it. All evaluation uses public benchmarks and public checkpoints — BEAM, LoCoMo, LongMemEval and HotpotQA — and involves no human subjects, no personally identifying data, and no new data collection or release. The work does not raise any specific ethical concerns.

Reproducibility statement

Most of what is needed to reproduce this work is in the appendices: the measured model's layer inventory, parameter shapes and the units in which τ is measured (Appendix B); the post-training

objective, grader weights, verbatim prompt and data statistics, taken from the launched job's own recorded configuration (Appendix C); and every theoretical claim stated with its assumptions and proved (Appendices A, A.5 and A.3). Evaluation uses the public BEAM benchmark under its own prompt with `gpt-4.1-mini` as judge at temperature 0, with the abstention, YaRN and token-level protocols in Appendices F, G and H. We will release the model architecture and training code upon acceptance.

AI USE STATEMENT

Large language model assistance was used in refining the text for this paper. It was not used to generate research ideas, to design or run experiments, or to produce experimental numbers: every quantity reported here is read from evaluation output or from the launched jobs' own logs and configurations. The authors verified all numbers, derivations and claims, and take full responsibility for the content of the paper, including any errors.

REFERENCES

LoCoMo-plus: Beyond-factual cognitive memory evaluation framework for LLM agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, 2026.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018. arXiv:1803.01271.

Yushi Bai et al. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024a.

Yushi Bai et al. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks, 2024b. arXiv:2412.15204.

Maximilian Beck et al. xLSTM: Extended long short-term memory. In *Advances in Neural Information Processing Systems*, 2024.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. In *Advances in Neural Information Processing Systems*, 2025.

Sebastian Borgeaud et al. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Aydar Bulatov, Yuri Kuratov, and Mikhail S. Burtsev. Recurrent memory transformer. In *Advances in Neural Information Processing Systems*, 2022.

Aydar Bulatov, Yuri Kuratov, and Mikhail Burtsev. Scaling transformer to 1m tokens and beyond with RMT, 2023. arXiv:2304.11062.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Wonkyo Choe et al. RWKV-edge: Deeply compressed RWKV for resource-constrained devices, 2024. arXiv:2412.10856.

Zihang Dai et al. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

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*, 2024.

Soham De et al. Griffin: Mixing gated linear recurrences with local attention for efficient language models, 2024a. arXiv:2402.19427.

Soham De et al. RecurrentGemma: Moving past transformers for efficient open language models, 2024b. arXiv:2404.07839.

Howard Eichenbaum. Time cells in the hippocampus: A new dimension for mapping memories. *Nature Reviews Neuroscience*, 15:732–744, 2014.

Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Re. Hungry hungry hippos: Towards language modeling with state space models. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei A. Efros. Test-time training with masked autoencoders. In *Advances in Neural Information Processing Systems*, 2022.

Paolo Glorioso et al. Zamba: A compact 7b SSM hybrid model, 2024. arXiv:2405.16712.

Alex Graves et al. Hybrid computing using a neural network with dynamic external memories. *Nature*, 540:471–477, 2016.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2023. arXiv:2312.00752.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Re. HiPPO: Recurrent memory with optimal polynomial projections. In *Advances in Neural Information Processing Systems*, 2020.

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Re. Combining recurrent, convolutional, and continuous-time models with the linear state space layer. In *Advances in Neural Information Processing Systems*, 2021.

Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. In *Advances in Neural Information Processing Systems*, 2022.

Ali Hatamizadeh, Yejin Choi, and Jan Kautz. Gated DeltaNet-2: Decoupling erase and write in linear attention, 2026. arXiv:2605.22791.

Marc W. Howard et al. A unified mathematical framework for coding time, space, and sequences in the hippocampal region. *Journal of Neuroscience*, 34:4692–4707, 2014.

Cheng-Ping Hsieh et al. RULER: What's the real context size of your long-context language models?, 2024. arXiv:2404.06654.

DeLesley Hutchins, Imanol Schlag, Yuhuai Wu, Ethan Dyer, and Behnam Neyshabur. Block-recurrent transformers. In *Advances in Neural Information Processing Systems*, 2022.

Ben Jacques et al. DeepSITH: Efficient learning via decomposition of what and when across time scales. In *Advances in Neural Information Processing Systems*, 2021.

Ben Jacques et al. A deep convolutional neural network that is invariant to time rescaling. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and Francois Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Urvashi Khandelwal et al. Generalization through memorization: Nearest neighbor language models. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

Jan Koutnik, Klaus Greff, Faustino Gomez, and Jurgen Schmidhuber. A clockwork RNN. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Benjamin J. Kraus, Robert J. Robinson, John A. White, Howard Eichenbaum, and Michael E. Hasselmo. Hippocampal 'time cells': Time versus path integration. *Neuron*, 78:1090–1101, 2013.

Yuri Kuratov et al. BABILong: Testing the limits of LLMs with long context reasoning-in-a-haystack. In *Advances in Neural Information Processing Systems*, 2024.

Aakash Lahoti et al. Mamba-3: Improved sequence modeling using state space principles. In *Proceedings of the 14th International Conference on Learning Representations*, 2026.

Patrick Lewis et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, 2020.

Opher Lieber et al. Jamba: A hybrid transformer-Mamba language model, 2024. arXiv:2403.19887.

Xuezhe Ma, Chunting Zhou, Xiang Kong, Junxian He, Liangke Gui, Graham Neubig, Jonathan May, and Luke Zettlemoyer. Mega: Moving average equipped gated attention. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

Christopher J. MacDonald, Kyle Q. Lepage, Uri T. Eden, and Howard Eichenbaum. Hippocampal 'time cells' bridge the gap in memory for discontiguous events. *Neuron*, 71:737–749, 2011.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, and Francesco Barbieri. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Tsendsuren Munkhdalai et al. Leave no context behind: Efficient infinite context transformers with Infini-attention, 2024. arXiv:2404.07143.

Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. Phased LSTM: Accelerating recurrent network training for long or event-based sequences. In *Advances in Neural Information Processing Systems*, 2016.

Antonio Orvieto, Samuel L. Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Charles Packer et al. MemGPT: Towards LLMs as operating systems, 2023. arXiv:2310.08560.

Eva Pastalkova, Vladimir Itskov, Asohan Amarasingham, and Gyorgy Buzsaki. Internally generated cell assembly sequences in the rat hippocampus. *Science*, 321:1322–1327, 2008.

Bo Peng et al. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Jack W. Rae et al. Compressive transformers for long-range sequence modelling. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

Imanol Schlag, Kazuki Irie, and Jurgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Karthik H. Shankar and Marc W. Howard. A scale-invariant internal representation of time. *Neural Computation*, 24:134–193, 2012.

Karthik H. Shankar and Marc W. Howard. Optimally fuzzy temporal memory. *Journal of Machine Learning Research*, 14:3785–3812, 2013.

Karthik H. Shankar and Marc W. Howard. Scale-invariant temporal history: Optimal slicing of the past in an uncertain world. *Neural Computation*, 2018.

Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified state space layers for sequence modeling. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

Yu Sun et al. Learning to (learn at test time): RNNs with expressive hidden states, 2024. arXiv:2407.04620.

Yutao Sun et al. Retentive network: A successor to transformer for large language models, 2023. arXiv:2307.08621.

Mohammad Tavakoli et al. Beyond a million tokens: Benchmarking and enhancing long-term memory in LLMs. In *Proceedings of the 14th International Conference on Learning Representations*, 2026.

Zoran Tiganj, Michael E. Hasselmo, and Marc W. Howard. A simple biophysically plausible model for long time constants in single neurons. *Hippocampus*, 25:27–37, 2015.

Albert Tsao, Jorgen Sugar, Li Lu, Cheng Wang, James J. Knierim, May-Britt Moser, and Edvard I. Moser. Integrating time from experience in the lateral entorhinal cortex. *Nature*, 561:57–62, 2018.

Aaron R. Voelker and Chris Eliasmith. Improving spiking dynamical networks: Accurate delays, higher-order synapses, and time cells. *Neural Computation*, 30:569–609, 2018.

Aaron R. Voelker, Ivana Kajic, and Chris Eliasmith. Legendre memory units: Continuous-time representation in recurrent neural networks. In *Advances in Neural Information Processing Systems*, 2019.

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *Proceedings of the 13th International Conference on Learning Representations*, 2025.

Yuhuai Wu et al. Memorizing transformers. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.

Xu et al. From single to multi-granularity: Toward long-term memory association and selection of conversational agents. In *Proceedings of the 14th International Conference on Learning Representations*, 2026.

Jing Xu, Arthur Szlam, and Jason Weston. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *Proceedings of the 13th International Conference on Learning Representations*, 2025.

Songlin Yang et al. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems*, 2024a.

Songlin Yang et al. Gated linear attention transformers with hardware-efficient training. In *Proceedings of the 41st International Conference on Machine Learning*, 2024b.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. HELMET: How to evaluate long-context language models effectively and thoroughly. In *Proceedings of the 13th International Conference on Learning Representations*, 2025.

Xinrong Zhang et al. ∞bench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, 2024.

A PRELIMINARIES AND DETAILED DERIVATIONS

This appendix expands §5, and records the general form of the recurrence set aside in §2.3.

A.1 THE GENERAL UNROLLING

Without simplifying the gate, write the per-step transition as $\boldsymbol{G}_t := \mathrm{Diag}(\boldsymbol{\alpha}_t) - \beta_t \boldsymbol{k}_t^\top \boldsymbol{k}_t$, so that equation 1 is $\boldsymbol{S}_t = \boldsymbol{G}_t \boldsymbol{S}_{t-1} + \beta_t \boldsymbol{k}_t^\top \boldsymbol{t}_t$. Iterating from $\boldsymbol{S}_{t_0} = \boldsymbol{M}_0$,

$$\boldsymbol{S}_t = \Big(\prod_{r=t_0+1}^{t} \boldsymbol{G}_r \Big) \boldsymbol{M}_0 \; + \; \sum_{u=t_0+1}^{t} \Big(\prod_{r=u+1}^{t} \boldsymbol{G}_r \Big) \beta_u \boldsymbol{k}_u^\top \boldsymbol{t}_u, \tag{16}$$

products taken in decreasing order of r. Every write is multiplied by exactly those transitions that occurred after it.

Two reductions recover the form used in the body. If the gate is scalar, $\mathrm{Diag}(\boldsymbol{\alpha}_t) = \bar{\alpha} I$, and the delta term is dropped, then $\prod_{r=u+1}^{t} \boldsymbol{G}_r = \bar{\alpha}^{\,t-u} I$ and equation 16 collapses to equation 3. If the gate is diagonal but not scalar, the product is diagonal with entries $\prod_r \alpha_{r,i}$: each key coordinate decays at its own rate, but each still has only one rate. The delta term makes the product non-diagonal; it subtracts the component of the stored state along the re-used address, which removes superseded values. Neither introduces a second time constant for a given coordinate, which is the claim of §2.3.

A.2 FROM THE RECURRENCE TO THE POLYNOMIAL

Again in two steps. Reading is $\boldsymbol{o} = \boldsymbol{q}\boldsymbol{S}_t$, and since $\boldsymbol{q}(\boldsymbol{k}^\top \boldsymbol{t}) = (\boldsymbol{q}\boldsymbol{k}^\top)\boldsymbol{t} = \langle \boldsymbol{q}, \boldsymbol{k} \rangle \boldsymbol{t}$, the query turns each stored outer product into its value scaled by a key match.

Step 1 — apply the query to equation 3, in absolute position:

$$\boldsymbol{o} = \sum_{\substack{u=1 \\ \text{(oldest)}}}^{\substack{t \\ \text{(newest)}}} \bar{\alpha}^{\,t-u} \; \beta_u \, \langle \boldsymbol{k}_u, \boldsymbol{q} \rangle \; \boldsymbol{t}_u. \tag{17}$$

Step 2 — re-index by depth $\Delta = t - u$, as in equation 4:

$$\boldsymbol{o} = \sum_{\substack{\Delta=0 \\ \text{(current token)}}}^{\substack{t-1 \\ \text{(start of context)}}} \bar{\alpha}^{\,\Delta} \; \underbrace{\beta_{t-\Delta} \langle \boldsymbol{k}_{t-\Delta}, \boldsymbol{q} \rangle}_{= w_\Delta \in \mathbb{R}} \; \underbrace{\boldsymbol{t}_{t-\Delta}}_{= \boldsymbol{u}_\Delta \in \mathbb{R}^{d_v}}, \tag{18}$$

which is equation 13 with $t = T$. The scalar w_Δ measures how strongly the probe matches the key written Δ tokens back; the vector $\boldsymbol{u}_\Delta$ is the value stored there. Only w_Δ depends on the probe; only $\bar{\alpha}^\Delta$ depends on the retention.

The reading is a vector. To treat it as a scalar polynomial, fix any $\boldsymbol{e} \in \mathbb{R}^{d_v}$ and take $\varphi = \langle \boldsymbol{e}, \boldsymbol{o} \rangle$, giving $c_\Delta = w_\Delta \langle \boldsymbol{e}, \boldsymbol{u}_\Delta \rangle$ and $\varphi = P(\bar{\alpha})$ as in equation 14. Nothing is lost by this: the vector reading contains at least as much information as any one projection of it, so a negative result about the scalar problem is a negative result about the vector problem, and the positive results of Appendix D may be run on any convenient $\boldsymbol{e}$.

Repeating for each head m replaces $\bar{\alpha}$ by $\bar{\alpha}_m$, so the grid reports $P(\bar{\alpha}_1), \ldots, P(\bar{\alpha}_M)$: M evaluations of one fixed polynomial at M different points.

A.3 ON THE SPARSITY ASSUMPTION

The sparsity assumption of §5 is an idealisation in two respects, and it is worth being explicit about both.

It is exact sparsity, not approximate. Real relevance profiles have many small nonzero coefficients rather than exact zeros. The honest version is that $\boldsymbol{c}$ is well approximated by its s largest entries,

and the tail acts as an additive perturbation of the readings. Recovery then inherits the conditioning of the problem, which Proposition **??** and equation **??** quantify: the tail is harmless when the nodes are well separated and fatal when they coalesce.

The sparsity is probe-dependent, and set partly by the key geometry. Because w_Δ contains $\langle \boldsymbol{k}_{t-\Delta}, \boldsymbol{q} \rangle$, a probe that matches many keys weakly has a dense profile even if only a few tokens are semantically relevant. This is the coupling made explicit in equation 26: good key separation makes s small, poor key separation inflates it. Sparsity is therefore not a free assumption about language but a joint property of the question and the learned key space.

A useful diagnostic follows: s can be estimated empirically for a given probe by counting how many positions carry a key match above a threshold, and tracked against M as the context grows.

A.4 How $\bar{\alpha}$ and τ appear in the implementation

The analysis treats $\bar{\alpha}$ as a per-head constant. The implementation does not store it. It stores a per-head rate $A_{\log}$ and a per-head, per-channel bias `dt_bias`, and forms the log-retention in the kernel as

$$\log \bar{\alpha}_{m,c,t} \;=\; -\underbrace{\exp\big(A_{\log,m}\big)}_{=:A_m} \cdot \underbrace{\operatorname{softplus}\big(f_{\mathrm{proj}}(x)_{c,t} + \texttt{dt_bias}_{m,c}\big)}_{=:\,\mathrm{d}t_{m,c,t}}, \tag{19}$$

so that by equation 5

$$\bar{\alpha}_{m,c,t} = e^{-A_m\,\mathrm{d}t_{m,c,t}}, \qquad \tau_{m,c,t} = \frac{1}{A_m\,\mathrm{d}t_{m,c,t}}. \tag{20}$$

(The kernel multiplies the cumulative sum by $1/\log 2$ so that it may exponentiate base two; this cancels and does not enter equation 19.)

Three points follow, and they qualify every number reported in §4.

$\bar{\alpha}$ is not a parameter. It is a function of the input through f_{proj}. Only A_m is a fixed per-head quantity. The constant-$\bar{\alpha}$ idealisation of §2.3 is therefore exact only when $\mathrm{d}t$ is stationary across the span being analysed; the standing caveats record this.

Units. τ is measured in tokens only if $\mathrm{d}t$ is taken as 1. The tables of §4 report the *static* timescale

$$\tau_m^{\text{stat}} := 1/A_m \qquad \text{or} \qquad \tau_{m,c}^{\text{stat}} := 1/\big[A_m \operatorname{softplus}(\texttt{dt_bias}_{m,c})\big], \tag{21}$$

obtained by setting $f_{\mathrm{proj}} = 0$. Realised timescales are $\tau^{\text{stat}}/\mathrm{d}t$, and since $\mathrm{d}t < 1$ over much of its range they are correspondingly *longer*. A tabulated value of, say, 2 should not be read as a memory of two tokens; the absolute scale requires measuring $\mathrm{d}t$ on real inputs, which we have not done.

What this does not affect. Every conclusion drawn in §4 — spans, octaves, step ratios, coefficients of variation — is a ratio of timescales within a layer, in which any common factor of $\mathrm{d}t$ cancels. The finding that coverage moved from 4.00 to 6.7 octaves is independent of this convention.

At initialisation the A-grid is exactly a τ-grid. `dt_bias` is initialised to zero, so $\mathrm{d}t_{m,c,t}$ is the same function of the input for every head, and equation 20 gives $\tau_m \propto 1/A_m$ exactly. Placing A on a geometric grid therefore places τ on one, which is what §3 intends. With the default $A \in [1, 16]$ and $\mathrm{d}t = \operatorname{softplus}(0) = \log 2$, the initial static timescales are

$$\tau_m^{\text{init}} = \frac{1}{A_m \log 2} \in [\,0.090,\ 1.443\,], \tag{22}$$

of span exactly $16\times$ and constant ratio $16^{1/7} = 1.486$.

A second route to irregularity. After training, `dt_bias` is no longer zero and varies by head *and* channel, so τ is no longer proportional to $1/A$. The grid can therefore become irregular even if A itself remains perfectly geometric: dispersion in `dt_bias` is an independent mechanism alongside drift in A. The A component and the `dt_bias` component contribute separately to the pooled figure of 6.7 octaves. Separating their contributions is a straightforward extension we have not made.

A.5 WHY THE RETENTIONS MUST BE GEOMETRIC

Equation equation 8 dictates the answer. Because the loss depends on the *factor* by which a retention differs from the depth being read, and not on the number of tokens between them, write $c(\Delta, \tau) := \max\{\Delta/\tau,\, \tau/\Delta\}$ for that factor: $c = 1$ is a perfect match and larger c is worse. A head at $\tau = 1000$ serves depth 2000 exactly as well as a head at $\tau = 10$ serves depth 20, both having $c = 2$. A design fixes a band $[\Delta_{\min}, \Delta_{\max}]$ of depths it must serve, the shallowest and the deepest, say tens of tokens and 10^5 and a set of retentions is adequate over that band exactly when every depth in it has some retention with c below a chosen ceiling.

Taking logarithms turns this into a tiling problem. A head with timescale τ serves the depths $[\tau/c,\, \tau c]$, an interval whose width *in tokens* grows in proportion to τ, hundreds of tokens at $\tau = 1000$, a few at $\tau = 10$, but whose width *on a logarithmic axis* is $2 \log c$ for every head alike. Covering the band without gaps is therefore the problem of laying equal-width tiles along an interval of length $\log(\Delta_{\max}/\Delta_{\min})$, and equal tiles must be laid at equal spacing. Equal spacing in $\log \tau$ is constant *ratio* in τ: the retentions must form a geometric sequence, and no other arrangement covers the band without leaving gaps or wasting heads.

Two things follow, and §3 makes both precise. The ratio between neighbouring timescales fixes the *quality* of coverage, the hardest depth to serve being the one that falls midway between two of them on the log axis. And because the tiles are of equal width in logarithm, the number of heads needed grows with the *logarithm* of the range to be covered rather than with the range itself, which is what makes spanning five orders of magnitude with a handful of heads possible at all. Spacing the timescales uniformly instead would place almost all of them at the long end and leave the short end bare, because equal differences are not equal ratios.

A.6 THE TWO-HEAD CASE IN FULL

Example 2 (Two heads, one fact)**.** *Let $M = 2$ and $s = 1$. The two readings are $\varphi_1 = a\,\bar{\alpha}_1^{\Delta}$ and $\varphi_2 = a\,\bar{\alpha}_2^{\Delta}$. Dividing eliminates the amplitude, and taking logarithms gives*

$$\Delta = \frac{\log(\varphi_2/\varphi_1)}{\log(\bar{\alpha}_2/\bar{\alpha}_1)}, \qquad a = \varphi_1\,\bar{\alpha}_1^{-\Delta}. \tag{23}$$

Two numbers recover both the content of the fact and how far back it lies*, whatever the context length, and the only thing required is $\bar{\alpha}_1 \neq \bar{\alpha}_2$. A single head cannot do this at all: one reading confounds a strong distant trace with a weak recent one.*

We prove this in detail. Let $s = 1$, with one fact of amplitude $a \neq 0$ at depth $\Delta \geq 0$, and two distinct retentions $\bar{\alpha}_1 \neq \bar{\alpha}_2$ in $(0, 1)$. The readings are $\varphi_1 = a\bar{\alpha}_1^{\Delta}$, $\varphi_2 = a\bar{\alpha}_2^{\Delta}$. Since $a \neq 0$ and $\bar{\alpha}_i^{\Delta} > 0$, both are nonzero and of the same sign, so their ratio is positive and

$$\frac{\varphi_2}{\varphi_1} = \Big(\frac{\bar{\alpha}_2}{\bar{\alpha}_1}\Big)^{\Delta} \implies \Delta = \frac{\log(\varphi_2/\varphi_1)}{\log(\bar{\alpha}_2/\bar{\alpha}_1)}, \tag{24}$$

the denominator being nonzero precisely because the retentions differ. The amplitude then follows as $a = \varphi_1 \bar{\alpha}_1^{-\Delta}$. Both steps are exact and neither involves T.

The contrast with a single head is the point. With $M = 1$ we observe only $\varphi_1 = a\bar{\alpha}_1^{\Delta}$, one equation in two unknowns: every pair (a, Δ) with $a = \varphi_1 \bar{\alpha}_1^{-\Delta}$ explains it equally well. A strong distant trace and a weak recent one are indistinguishable. Adding one more retention breaks exactly that degeneracy, and this is the elementary form of the claim that M, not d_k, governs how much of the past can be disentangled.

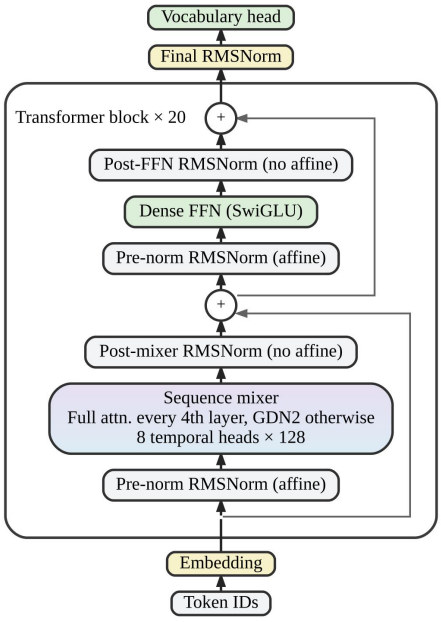


Figure 3: Twenty-block architecture. Every fourth block uses full attention; the others use GDN2.

B MODEL ARCHITECTURE

d_{model}	2560 *(cfg)*
layers	20; pattern `[gdn, gdn, gdn, attn]` $\times 5$
GDN layers	15: $0, 1, 2,\ 4, 5, 6,\ 8, 9, 10,\ 12, 13, 14,\ 16, 17, 18$
attention layers	5: $3, 7, 11, 15, 19$
softmax heads / KV heads	40 / 5 *(cfg)*
FFN	dense, `ffn_dim_multiplier` $= 4.0$, width 10,240 *(cfg)*
parameters	2.87B (embed+head 1.03B, attn 74M, GDN $\sim$187M, FFN 1.57B)

The mixture-of-experts sibling replaces the dense FFN with 256 experts at top-8 routing, 52.4B total and $\sim$1.23B active at the same active width; the recurrent machinery is identical.

A GDN layer , shard 0 of 8:

tensor	shard shape	role
`attention.A`	$(1,)$	$A_{\log}$, **one per head**; $M = 8$ across shards
`attention.dt_bias`	$(128,)$	per head $\times$ channel; 8×128 total
`attention.f_proj_layer2.weight`	$(128, 320)$	input-dependent forget gate $\boldsymbol{g}_t$
`attention.g_proj_layer2.weight`	$(128, 320)$	output gate
`attention.{q,k,v}_conv1d.weight`	$(128, 1, 4)$	short causal conv, width 4
`attention.wqkvbfg.weight`	$(385, 2560)$	fused q, k, v, β, f, g
`attention.wqkvbfg.shared_weights.{0,1}`	$(40, 2560)$	low-rank shared factors
`attention.o_proj.weight`	$(320, 1024)$	output, value width 8×128

(cfg): `num_heads` $=$ 8, `head_dim` $=$ 128, `temporal_heads` $=$ `True`, `f_head_repeat_factor` $=$ 16, `low_rank_compression_ratio` $=$ 8. Note `temporal_heads` is a boolean selecting the geometric initialisation, not a count; the count is `num_heads`.

State size. Each head carries $\boldsymbol{S}^{(m)} \in \mathbb{R}^{128\times128}$, so a layer holds $8 \times 16{,}384 = 131{,}072$ scalars and the network $\approx 1.97 \times 10^6$ — fixed in T. A KV cache on the same geometry at $T = 10^5$ would be $\approx 2.6 \times 10^9$.

Realised retention. The kernel assembles decay from $A_{\log}$, `dt_bias` and the projected gate: $\text{rate}_{m,c}(x) = \exp(A_m)\,\text{softplus}(\texttt{dt_bias}_{m,c} + f_{\text{proj}}(x)_c)$. §4 measures the static part ($f_{\text{proj}} = 0$); the realised spectrum requires forward passes.

Attention layers carry only `qkv_proj.weight` $(720, 2560)$, `qkv_proj.norm.weight` $(320,)$ and `o_proj.weight` $(320, 2560)$: grouped-query softmax attention, no recurrent state. These are the $M = T$ layers, the only place an arbitrary-depth lookup is exact.

C RL SETUP

Objective. Group-relative policy optimisation (GRPO) on the 2.9B Gated DeltaNet policy, initialised from the SFT checkpoint at step 20. Per prompt the sampler draws $G{=}8$ completions at temperature 1.0 and top-p 1.0, and completion i takes the group-centred advantage $A_i = r_i - \frac{1}{G}\sum_j r_j$ — no value network, no cross-batch reward normalisation. AdamW, learning rate 1×10^{-5}, weight decay 0, gradient clip 1.0, warmup from $0.1\times$ peak, 64 rollouts per step, sequence length 32,768. The only regulariser is a backward KL of 0.01 against the sampler policy.

Reward. The reward is verifiable: every prompt ships a gold string, so no judge model is involved and r is a deterministic function of two strings. The reward is a binary recall, $r = 1$ if the gold string appears in prediction and 0 otherwise (`require_full_match: false`, `anchor_len: 50`); the exact-match term is logged as an unweighted probe rather than optimised.

Prompt. Each training row is a single user turn under the `[INST]` framing the SFT checkpoints were tuned on, carrying a recall marker, the conversation and the question.

Single-turn recall prompt

```
[INST] «MEMORY_RECALL_TASK»
This block is a memory-recall exercise.  The rules below apply ONLY
inside this block and to no other kind of question.
1.  Answer strictly from the conversation given below; do not use
outside knowledge.
2.  Answer only the question that is asked.  Never write an additional
Question:  or Answer:  of your own.
3.  Follow whatever format that task asks for.
«BEGIN_CONVERSATION»
⟨conversation⟩
Question:  ⟨question⟩
Answer:  [/INST]
```

Data. `memory_recall_train_inst.jsonl`: 37,142 prompts, 1.23 GB, mixing HotpotQA and LoCoMo roughly 54/46 by count, both from *train* splits only. Prompt length is heavily skewed — median 8,266 characters, mean 30,582, 95th percentile 90,098, maximum 103,932 — while gold answers are short, median 14 characters. Note the ceiling: at a 32,768-token sequence length nothing in RL exposes the policy to more than roughly 100K characters, so every BEAM split above 100K is out-of-distribution for this stage.

Reading the reward curve. The recall reward climbs from 0.029 to 0.631 over the run (Figure 4),

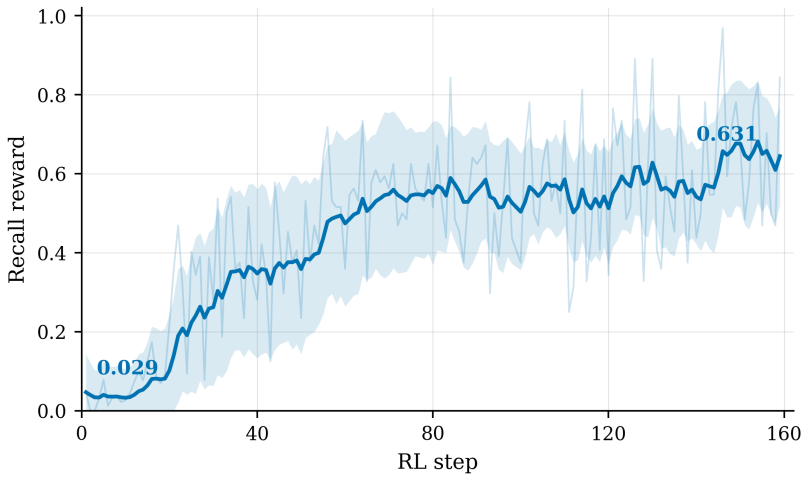


Figure 4: Recall reward over the single-turn RL run, all 159 completed steps (the evaluated checkpoint is step 160). The faint trace is the per-step value, the solid line an exponential moving average, and the band ± 1 within-group standard deviation. The reward rises from 0.029 to 0.631, but it asks only whether the gold string appears somewhere in the response, so the rise is not by itself evidence that answers improved.

D PRONY RECOVERY UNDER GEOMETRIC RETENTIONS

Remark 3 (When a constructive algorithm exists). If the *retentions* are placed in geometric progression, $\bar{\alpha}_m = \bar{\alpha}^m$, then with $z_j := \bar{\alpha}^{\Delta_j}$ the readings equation **??** become power sums $\varphi_m = \sum_j a_j z_j^m$, and Prony's method recovers the depths and amplitudes from exactly $2s$ readings. This is the second special case: a sufficiency result, with $M = 2s$ retentions enough for s facts at any depths. Appendix D proves it in full.

That arrangement is *not* the grid equation 9: $\bar{\alpha}_m = \bar{\alpha}^m$ means $1/\tau_m = m/\tau_1$, i.e. *harmonic* spacing of timescales, whereas equation 9 is geometric in τ. The two objectives pull apart. Geometric τ buys scale-invariant coverage; harmonic τ buys a constructive inversion but crowds its timescales near τ_1 and spans only a factor M. Which is preferable, or whether a hybrid attains both, we leave open; the measurements of §4 concern equation 9.

The recovery statement quoted above is classical, and we do not reprove it. We record the hypotheses precisely, since the arrangement it concerns is not the one temporal heads use, and we prove the two steps that are specific to this setting: that depths map injectively to nodes, and that they can be read back.

E THE CAPACITY ACCOUNT, AND THE AXIS IT LEAVES OPEN

The standard explanation for weak retrieval in linear attention is a capacity argument, stated here in our notation so the two can be set side by side. With no decay and N unit-norm keys, $\boldsymbol{S} = \sum_i \boldsymbol{k}_i^\top \boldsymbol{t}_i$,

$$\boldsymbol{q} = \boldsymbol{k}_j : \quad \boldsymbol{k}_j \boldsymbol{S} = \boldsymbol{t}_j + \underbrace{\sum_{i \neq j} \langle \boldsymbol{k}_i, \boldsymbol{k}_j \rangle \boldsymbol{t}_i}_{\text{interference}}, \qquad \left\| \textstyle\sum_{i \neq j} \langle \boldsymbol{k}_i, \boldsymbol{k}_j \rangle \boldsymbol{t}_i \right\| \leq \mu (N-1) \max_i \|\boldsymbol{t}_i\|. \tag{25}$$

Signal-to-interference falls like $1/\mu N$; for random keys $\mu \asymp \sqrt{\log N / d_k}$; and $\operatorname{rank}(\boldsymbol{S}) \leq d_k$, so at most d_k mutually orthogonal traces are stored losslessly. That last figure is the exact ceiling but not the operative one: allowing coherence μ, the number of unit vectors packable in $\mathbb{R}^{d_k}$ grows like $\exp(\mu^2 d_k/4)$, passing $d_k = 128$ once $\mu \gtrsim 0.4$. What limits practice is tolerable interference, not 128 — which leaves room for a second constraint to bind.

E.1 What the account explains, and what it cannot see

The prescription drawn from equation 25 is a larger head dimension **[cite: Sun et al., head-dimension scaling]**, usually paired with a second observation: associations can only be added, never erased, so errors accumulate — which gated variants Yang et al. (2024b); Gu & Dao (2023) and the delta rule **?** mitigate. We agree with all of it. Our claim is only that it is silent on one axis, for a structural reason and with a different logical status.

The object it analyses has no time in it. That state is a *bag*: position does not appear, so the analysis cannot speak about *when* something was stored. Adding the gate — the very fix the account credits — puts position inside the coefficient, as equation 3 makes explicit. The account reads the gate as overload reduction, which is true and first-order, but $\bar{\alpha}^{\Delta}$ is also a *timestamp*: undecodable with one retention, decodable with M on a grid. That is Example 2 and Remark 3, and it is the content of temporal heads.

The two bounds differ in status. equation 25 holds unconditionally. The recovery view requires sparse relevance, a geometric grid, and retentions stable across the span. We are claiming an *additional* constraint that exists only in a multi-retention grid, not a replacement for the capacity one.

E.2 The two bounds constrain orthogonal axes

	bounds	failure it causes
d_k, μ	the **key** axis	facts with similar keys become confusable
M	the **time** axis	facts at similar depths become confusable

Facts with nearly identical keys collide wherever they sit, and no number of heads separates them; facts with distinct keys that one probe excites must be separated in time, and no increase in d_k does that. The axes are coupled only through s: because c_{Δ} contains the key match, sparsity is set by addressing quality, so

$$\boxed{d_k \text{ and } \mu \text{ determine how small } s \text{ is;} \qquad M \text{ determines how large } s \text{ may be.}} \tag{26}$$

Capacity, on this reading, does not fail by exhausting slots. It fails by degrading selectivity, inflating s until it exceeds what the available retentions can separate.

F Abstention

Abstention — declining to answer when the corpus does not contain the answer — behaves unlike the other nine abilities and is reported here rather than in Table 1, where its magnitude would dominate the averages.

Table 2: Abstention accuracy by corpus length. Best per row in bold.

Length	Ours	Gemma-3-27B-it	Qwen2.5-32B	Qwen3-30B-A3B
100K	0.025	0.725	**0.900**	0.300
500K	0.136	0.914	**1.000**	0.329
1M	0.046	0.900	**0.986**	0.386
10M	0.050	**0.950**	**0.950**	0.250

The gap is the largest in the paper and it runs the wrong way: the baselines abstain correctly 73–100% of the time while ours manages 2.5–14%. The reason it is reported apart from the other abilities is that it rewards exactly the limitation this paper is about. A model whose window admits only a fraction of the corpus has, for most unanswerable questions, genuinely seen nothing that answers them, so "the answer is not here" is correct by default; a model that ingests the whole corpus has to establish the same conclusion the hard way. The measurements show this directly. Gemma sees 97% of the 100K corpus and abstains correctly 0.725 of the time; at 10M it sees 0.97% and abstains 0.950 of the time — its score *improves* as its coverage collapses. Qwen2.5, with a 29,000-token window, scores 0.900–1.000 throughout. Ours ingests every corpus in full at all four

lengths and scores 0.025–0.136. Averaging such a quantity alongside recall abilities would credit truncation as if it were memory.

Beyond that, the difference is behavioural rather than architectural. Abstention requires certifying that *nothing* in the context answers the question, which under the account of §5 is the one query type sparse recovery cannot serve: a bounded state can report what it retains, but establishing that a fact is absent is not a recovery problem. The instruction-tuned baselines have also been trained to refuse, and ours has not. Excluding it from Table 1 avoids letting one ability of this kind set the averages; including it would raise every baseline average by 0.03–0.09 and ours by at most 0.01.

G BEAM WITH YARN-EXTENDED WINDOWS

Serving the Qwen checkpoints beyond their declared window tests whether the gap in Table 1 is an artefact of truncation. It is not: extending the window raises both Qwens at every corpus length we measured, 100K through 10M, and leaves the ordering of models unchanged. Figure 5 shows the two shortest lengths, where the extension buys the most.

H TOKEN-LEVEL PRECISION AND RECALL

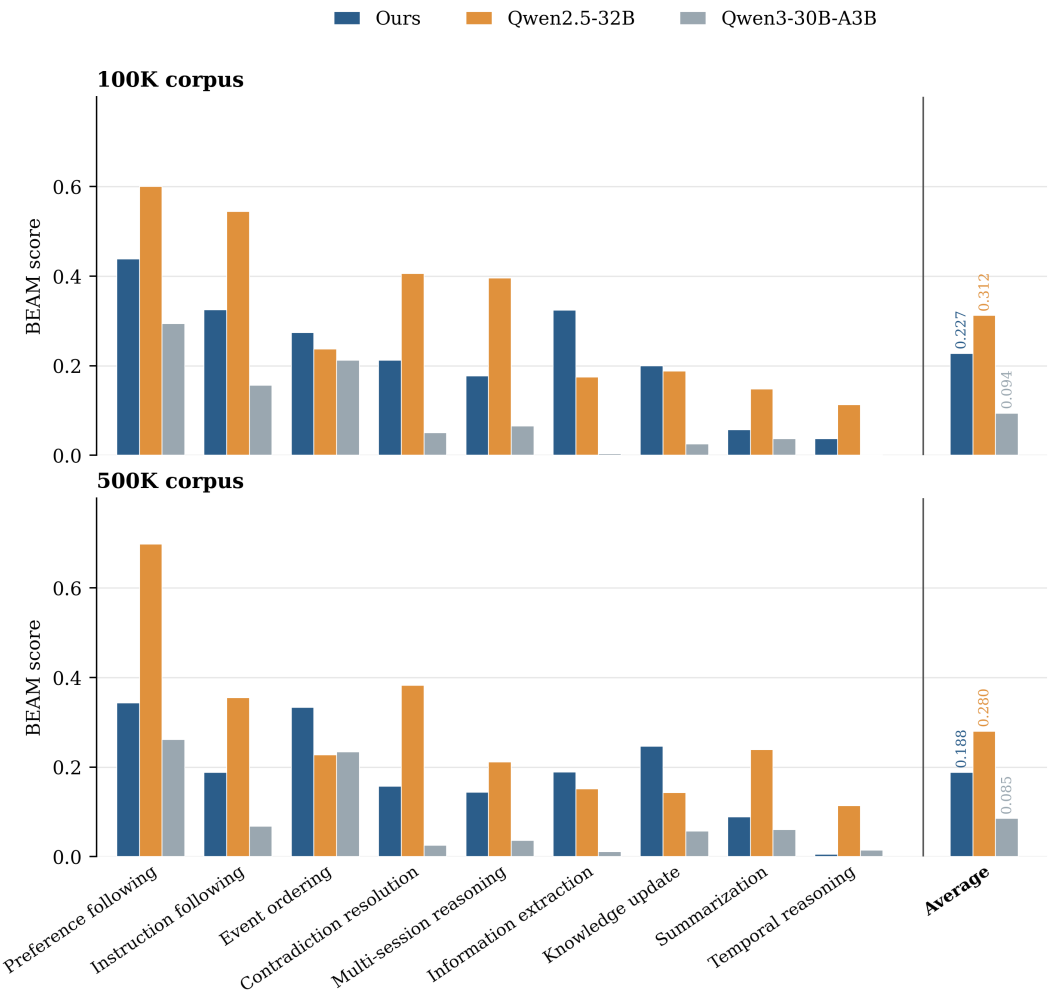


Figure 5: BEAM by ability at the two shortest corpus lengths, with both Qwen checkpoints YaRN-extended to ≈120K tokens and ours served unbounded. Gemma is omitted: its native 128K window needs no extension, so it is not a like-for-like comparison here. Abilities are ordered by mean score at 100K and held in that order in both panels. Abstention is excluded (Appendix F) and the average is over the nine abilities shown. Even with the windows extended the pattern of Table 1 persists: Qwen2.5 leads on aggregate, driven by preference and instruction following, while ours leads on *information extraction* and *event ordering* at 100K and on *event ordering* and *knowledge update* at 500K — the abilities that require locating a specific fact at a specific depth rather than conditioning on the conversation as a whole.

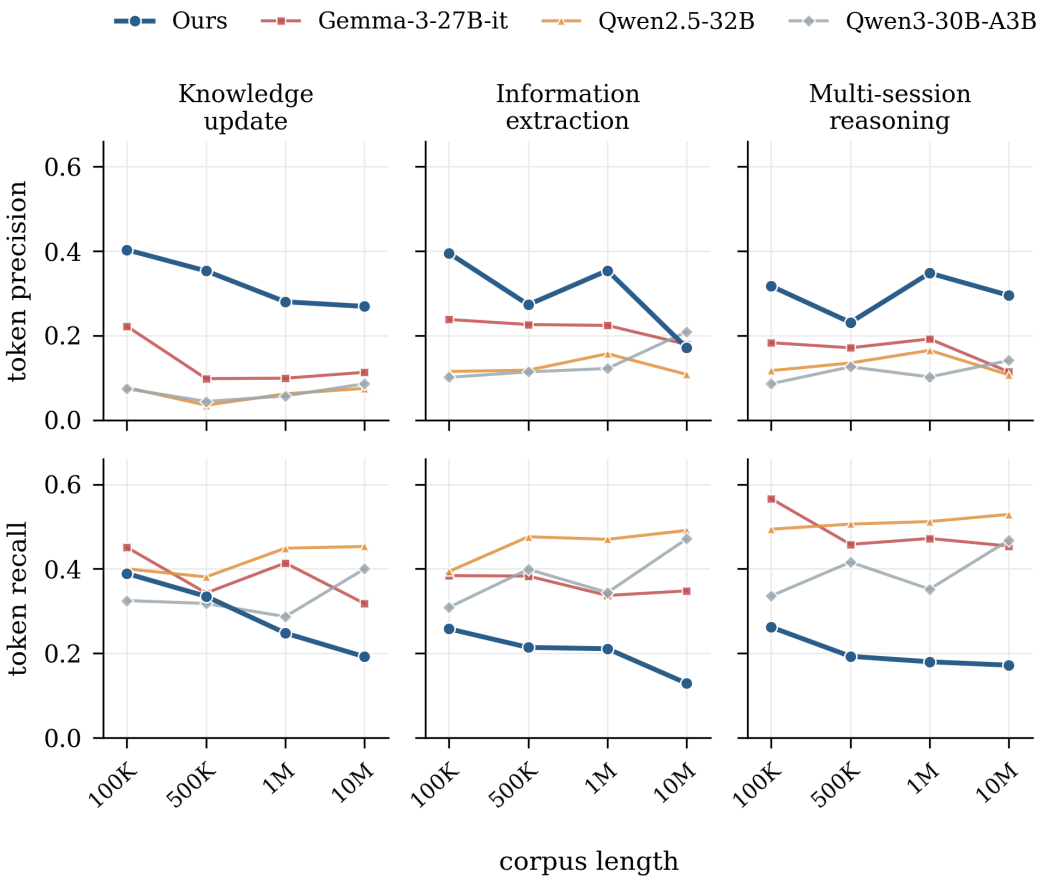


Figure 6: Token-level precision (top) and recall (bottom) against corpus length, for the three retrieval-style abilities. **Ours separates the two metrics in the way Appendix ?? predicts**: highest precision of the four models at every length on knowledge update and multi-session reasoning, and at three of four on information extraction, while simultaneously the lowest recall at every length on information extraction and multi-session reasoning. The baselines show the opposite profile, sitting low on precision and high on recall.

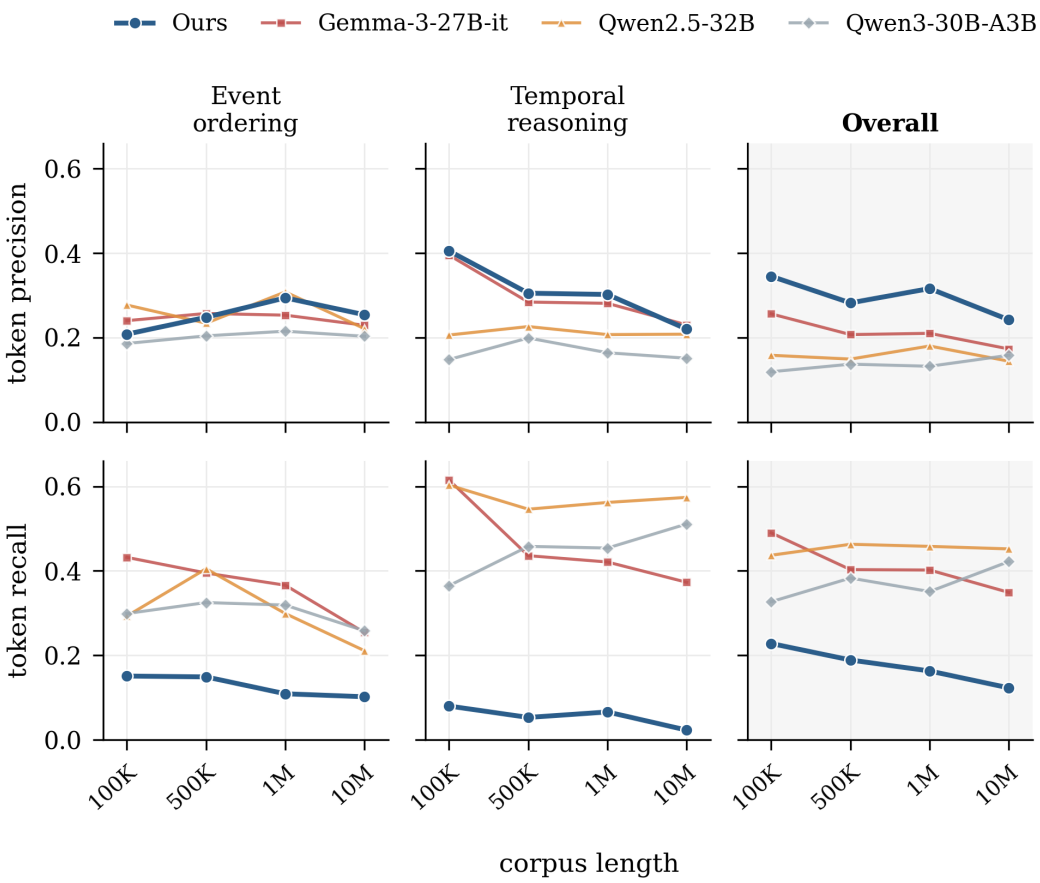


Figure 7: As Figure 6, for the two ordering-and-time abilities and the micro-average over all five. *Event ordering* is the one exception to the pattern: ours leads on precision at only one of the four lengths, and all four models converge there. *Temporal reasoning* shows the asymmetry at its most extreme — precision comparable to Gemma's at 100K, recall an order of magnitude below it, falling to 0.023 at 10M. In the *Overall* column ours has the highest precision and the lowest recall at all four lengths without exception.