FROM CACHE TO BELIEF: ONLINE PROBABILISTIC MEMORY FOR LONG-CONTEXT LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models that serve long documents, dialogues, and agents must reuse evidence that has left their attention window, and full attention over the whole history is costly. Compressed memories keep a bounded state instead, but store it as point estimates: a slot cannot tell whether it summarizes many consistent observations or a single noisy one, so how far to trust or overwrite it must be learned implicitly. We argue that bounded memory should maintain *beliefs* rather than points. **BeliefMem** is a lightweight adapter that keeps the history beyond the window of a frozen sliding-window model as a bank of Gaussian slot beliefs. One posterior responsibility decides where each evidence block is written and which slots are read; natural-parameter writes let accumulated precision resist overwriting, and a precision-weighted product of experts lets the read revise local evidence only as far as memory is more certain. Trained on 600 examples (27 minutes on one GPU at 7B) with 0.11–0.23% of the backbone parameters, BeliefMem improves six-task LongBench over the local model by 2.72, 4.05, and 2.65 points on Qwen2.5-3B/7B/14B, outperforms all nine Artificial Hippocampus Network (AHN) checkpoints at every scale, and transfers to LongBench v2 and InfiniteBench. A retrained point memory with the same interface trails it in every paired seed, and a causal probe shows that its benefit grows when the answer evidence lies beyond the window.

1 INTRODUCTION

Persistent agents, long dialogues, repository-scale code, and multi-document question answering require language models to reason over long, evolving histories (Park et al., 2023; Jimenez et al., 2024; Bai et al., 2024). Full attention over the whole history grows quadratically in cost, and even retained evidence is often poorly used as distractors accumulate (Liu et al., 2024; Hsieh et al., 2024; Bansal et al., 2026). A practical alternative keeps an exact local window and compresses older history into a bounded memory: recurrent segment states (Dai et al., 2019; Rae et al., 2019; Bulatov et al., 2022), memory tokens (Wang et al., 2024; Chevalier et al., 2023), or learned recurrent states attached to a frozen model (Fang et al., 2025).

These memories store *point estimates*, and a point cannot say how much evidence stands behind it. Consider two slots with similar content vectors: one has absorbed ten consistent mentions of an entity, the other a single noisy passage. A point memory treats them alike, so whether a slot should resist overwriting, or whether a read from disagreeing slots is reliable, must be learned implicitly from the task loss. Our insight is that a bounded memory should maintain *beliefs*: when each slot carries a precision that accumulates with evidence, these decisions follow from the state itself. A well-supported slot moves little under a new observation (raising its precision from 0.1 to 100 reduces the overwrite from 98% to 5%; Appendix F), and a read from disagreeing slots is broad and relied on less (Figure 1).

We turn this view into **BeliefMem** (Figure 2), an adapter that keeps the history beyond the local window of a frozen language model as Gaussian slot beliefs. One posterior responsibility decides where each evidence block is written and which slots are read, and a precision-weighted product of experts decides how far the read corrects the local route. Our contributions are:

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

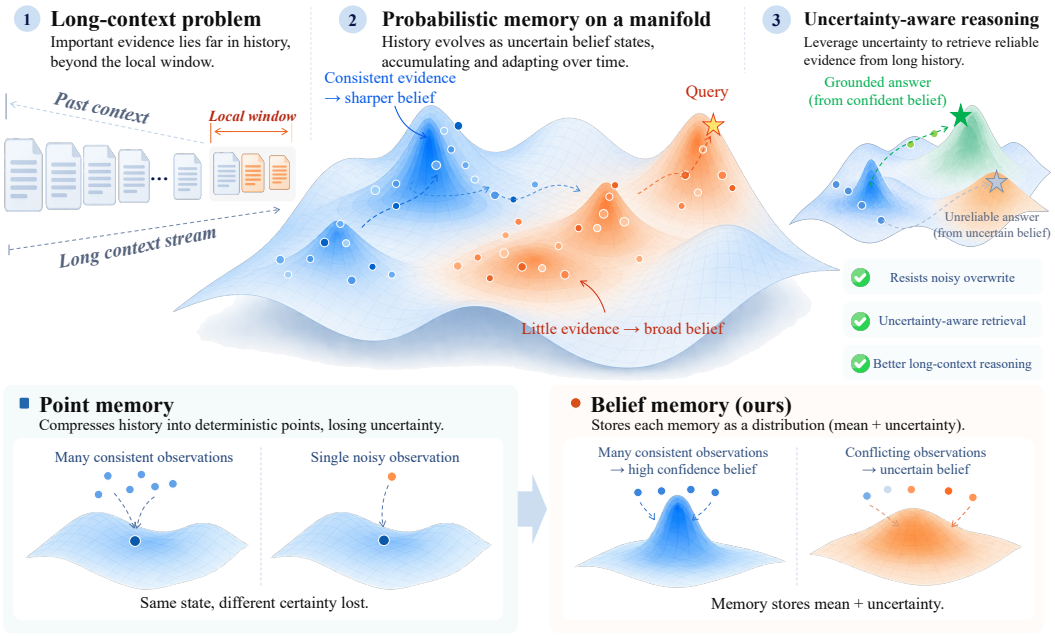


Figure 1: **From point memory to belief memory.** ❶ Needed evidence often lies beyond the local window. ❷ Consistent evidence sharpens a belief; a read from disagreeing slots is broad. ❸ These uncertainties decide which slots are read and how much the model relies on them. Bottom: a point memory maps consistent and noisy evidence to similar states; a belief memory tells them apart.

- A view of bounded long-context memory as online belief maintenance, and an adapter in which one responsibility couples assignment, natural-parameter writing, and posterior-predictive reading (Section 3).
- Controlled evidence that beliefs matter: against a retrained point memory with the same slots, width, decoder, data, and selection, the belief pathway improves six-task LongBench (LB6) by 1.33 points in all three paired seeds (Section 4.3).
- Gains at three scales over the local model and all nine AHN checkpoints, transfer to LongBench v2 and InfiniteBench, and a causal probe localizing the benefit to evidence beyond the window, with 0.11–0.23% trainable parameters and a 27-minute single-GPU adaptation at 7B (Section 4).

2 Related Work

External and latent memory for LLMs. Retrieval-augmented generation (Lewis et al., 2020; Borgeaud et al., 2022) and agent memories (Park et al., 2023; Packer et al., 2023; Zhong et al., 2024) append retrieved text, summaries, or records to the prompt, so their value depends on retrieval quality and prompt budget; MemAgent (Yu et al., 2026) learns a text-memory overwrite policy with reinforcement learning. Latent memories store machine-native states instead: cached key–value memories (Wang et al., 2023), memory tokens or banks (Wang et al., 2024; 2025), compressed context tokens (Mu et al., 2023; Ge et al., 2024; Chevalier et al., 2023), and associative or episodic memories (He et al., 2024; Das et al., 2024; Zhang et al., 2026). BeliefMem is a latent memory whose slots hold posteriors.

Compressed long-context inference. Transformer-XL (Dai et al., 2019), the Compressive Transformer (Rae et al., 2019), and the Recurrent Memory Transformer (Bulatov et al., 2022) carry compressed segment states; Infini-attention (Munkhdalai et al., 2024) and Titans (Behrouz et al., 2024) add online associative memories, and linear recurrent models (Gu & Dao, 2024; Peng et al., 2023; Yang et al., 2023) keep a fixed-size state. Key–value (KV) cache methods select (Zhang et al., 2023), stream (**?**), merge (Li et al., 2024), or compress (Liu et al., 2023) cached states. Closest to our

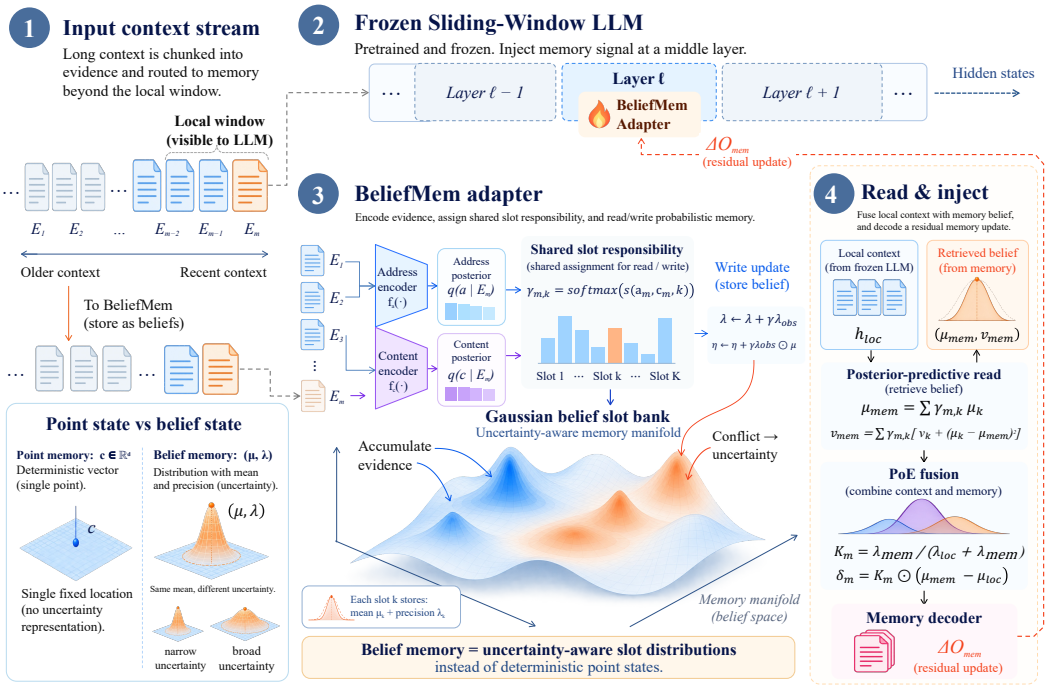


Figure 2: **Overview of BeliefMem.** ❶ Evidence blocks E_m stream in; the frozen model sees only the sinks and the local window. ❷ The adapter at layer ℓ adds a residual ΔO^{mem} to the attention output. ❸ One responsibility $\gamma_{m,k}$ assigns each block to Gaussian slot beliefs for both reading and writing; writes accumulate precision. ❹ The posterior-predictive read of the previous state is fused with the block's content posterior by a product of experts and decoded, conditioned on the query, into ΔO^{mem} (Section 3).

setting, Artificial Hippocampus Networks (AHN; Fang et al., 2025) and Memory Caching (Behrouz et al., 2026) pair an exact local window with a learned compressed state, and qTTT adapts query projections at test time (Bansal et al., 2026). All of these states are point estimates.

Probabilistic memory. The Kanerva Machine (Wu et al., 2018a) and its dynamic variant (Wu et al., 2018b) model memory as a Gaussian random matrix with analytic Bayesian writes for generative modeling of a few stored patterns, and Larimar (Das et al., 2024) uses such a memory for LLM knowledge editing. BeliefMem instead keeps per-slot diagonal beliefs with learned soft assignment inside a frozen language model and reads posterior-predictive moments, including cross-slot disagreement, that are fused with local evidence over long streams.

3 BeliefMem: Online Evidence Maintenance

Setting and notation. A frozen language model with parameters θ reads $x_{1:T}$. At position t its local route attends only to the attention sinks $\mathcal{S}$ (the first 128 tokens) and the sliding window $\mathcal{W}_t$ (the latest 8,064 tokens); we write $\ell_t = x_{\mathcal{S}\cup\mathcal{W}_t}$. The input is split into consecutive *evidence blocks* $E_1, E_2, \ldots$ of 256 tokens (during generation each new token forms its own block); cycle m processes E_m, and $m(t)$ is the block containing t. Superscripts a and c mark *address* (where evidence belongs) and *content* (what is retained); v is a variance, $\lambda = 1/v$ a precision, vector operations are elementwise, and ϑ collects all adapter parameters (full notation in Appendix H). BeliefMem approximates the full-history predictive distribution by a latent-memory model,

$$p^{\star}_{\theta}(x_t \mid x_{<t}) \approx \int p_{\theta,\vartheta}(x_t \mid \ell_t, z_t)\, p(z_t \mid M_{m(t)-1})\, dz_t, \tag{1}$$

where z_t summarizes the evidence outside the local view and M_m is a fixed-size bank of K_s Gaussian beliefs. Each cycle shares one slot assignment between reading and writing, and the read corrects the

local attention output for query q_t:

$$\begin{aligned} \gamma_m &= \mathrm{Assign}(M_{m-1}, E_m), \qquad (\mu_m^{\mathrm{mem}}, v_m^{\mathrm{mem}}) = \mathrm{Read}(M_{m-1}, \gamma_m), \\ M_m &= \mathrm{Write}(M_{m-1}, E_m, \gamma_m), \qquad O_t = \mathrm{Attn}(q_t; \ell_t) + \Delta O_t^{\mathrm{mem}}. \end{aligned} \tag{2}$$

Appendix I gives the latent model and its bound, and Algorithm 1 the complete cycle.

3.1 State and Evidence Encoding

At cycle m, slot $k \in \{1, \dots, K_s\}$ stores

$$M_{m,k} = (\eta^a_{m,k}, \lambda^a_{m,k}, \eta^c_{m,k}, \lambda^c_{m,k}, \alpha_{m,k}), \qquad \mu = \eta/\lambda, \quad v = 1/\lambda, \tag{3}$$

with diagonal precision $\lambda > 0$, precision-weighted mean $\eta = \lambda \odot \mu$, and Dirichlet assignment counts α_m; $\epsilon = 10^{-6}$ is a variance jitter. The initial state M_0 is learned: address means start orthogonal with unit variance, content precisions start near zero (an empty memory), and the counts start at the learned base counts α^0. At the adapted layer, two encoders (parameters $\phi \subset \vartheta$) map the block-averaged hidden states with the keys, $[h; k^{\mathrm{mha}}]$, and with the values, $[h; v^{\mathrm{mha}}]$, of the frozen multi-head attention (mha) on block E_m to posteriors $q_\phi(u \mid E_m) = \mathcal{N}(\mu^u_m, \mathrm{diag}\, v^u_m)$ for $u \in \{a, c\}$, of dimensions D_a and D_c.

3.2 One Responsibility for Reading and Writing

Holding the encoder posteriors fixed, the coordinate-ascent update of the slot assignment in the variational bound combines the expected log mixing weight with the expected log slot likelihoods (Appendix K). BeliefMem uses this form with an overlap score for content:

$$\gamma_{m,k} = \mathrm{softmax}_k \Big[\psi(\alpha_{m-1,k}) - \psi\Big(\textstyle\sum_j \alpha_{m-1,j}\Big) + A_{m,k} - \tau_m C_{m,k} \Big], \tag{4}$$

where ψ is the digamma function. With $\Delta^u_{m,k} = \mu^u_m - \mu^u_{m-1,k}$, $V^a_{m,k} = v^a_{m-1,k} + \epsilon$, and $V^c_{m,k} = v^c_{m-1,k} + v^c_m + \epsilon$,

$$\begin{aligned} A_{m,k} &= -\frac{1}{2D_a} \sum_j \left[\log V^a_{m,k,j} + \frac{v^a_{m,j} + (\Delta^a_{m,k,j})^2}{V^a_{m,k,j}} \right], \\ C_{m,k} &= \frac{1}{2D_c} \sum_j \left[\log V^c_{m,k,j} + \frac{(\Delta^c_{m,k,j})^2}{V^c_{m,k,j}} \right]. \end{aligned} \tag{5}$$

A is the exact expected Gaussian log density of the address, normalized by D_a, and C the negative log overlap of the content beliefs. Both depend on variances: a slot with a broad belief is a weaker match than an equally close precise slot. The content scale τ_m is predicted by a small network from score and uncertainty statistics (Appendix K); normalization, overlap, and τ_m are the departures from the exact update.

3.3 Predictive Read and Precision-Weighted Fusion

Using the *previous* state and the shared γ_m, the read is the content mixture's posterior predictive, summarized by its exact first two moments:

$$\mu^{\mathrm{mem}}_m = \sum_k \gamma_{m,k} \mu^c_{m-1,k}, \qquad v^{\mathrm{mem}}_m = \sum_k \gamma_{m,k} \left[v^c_{m-1,k} + (\mu^c_{m-1,k} - \mu^{\mathrm{mem}}_m)^2 \right]. \tag{6}$$

The second term measures disagreement between the slots being read, so an ambiguous read is uncertain even when every slot is precise. With $\lambda^{\mathrm{loc}}_m = (v^c_m + \epsilon)^{-1}$ and $\lambda^{\mathrm{mem}}_m = (v^{\mathrm{mem}}_m + \epsilon)^{-1}$, the Gaussian product of experts (PoE) of local and memory beliefs yields

$$K_m = \frac{\lambda^{\mathrm{mem}}_m}{\lambda^{\mathrm{loc}}_m + \lambda^{\mathrm{mem}}_m}, \quad \delta_m = K_m \odot (\mu^{\mathrm{mem}}_m - \mu^c_m), \quad \Delta O^{\mathrm{mem}}_t = D_\vartheta([q^{\mathrm{mha}}_t; \delta_m; K_m]). \tag{7}$$

K_m is a per-dimension reliance gain: the fused mean moves toward memory only as far as memory is more precise than local evidence. For each token t served by cycle m, the decoder D_ϑ (a two-layer MLP whose output layer is zero-initialized) maps its attention query q^{mha}_t, δ_m, and K_m to a residual added to the local attention output, so training starts from the local model. Because an empty slot has near-zero precision, a new memory is ignored until it has accumulated evidence (Appendix F).

3.4 Fractional Evidence Updates

After the read, block m is written with observation precision $\lambda_m^{u,\mathrm{obs}} = (v_m^u + \epsilon)^{-1}$:

$$\lambda_{m,k}^u = \lambda_{m-1,k}^u + \gamma_{m,k}\lambda_m^{u,\mathrm{obs}}, \qquad \eta_{m,k}^u = \eta_{m-1,k}^u + \gamma_{m,k}\lambda_m^{u,\mathrm{obs}} \odot \mu_m^u, \tag{8}$$

$$\alpha_{m,k} = (1-\rho)\alpha_{m-1,k} + \rho\alpha_k^0 + \gamma_{m,k}. \tag{9}$$

Eq. (8) is the analytic fractional-site update for the chosen Gaussian sites: precise incoming evidence moves a slot mean more, and accumulated precision resists overwriting. The Dirichlet refresh toward the learned base counts α^0 ($\rho = 0.01$) prevents early assignments from locking in the slot prior. Each cycle costs $O(K_s(D_a + D_c))$, independent of history length. Each block is read from M_{m-1} before it is written, and during generation each token is its own cycle, so every prediction depends only on its prefix (Appendix G).

3.5 Training Objective

Only the adapter is trained. The objective combines top-k distillation from the full-attention model, a small language-modeling term, and a memory loss:

$$\begin{aligned} \mathcal{L}_{\mathrm{train}} &= \mathcal{L}_{\mathrm{kd}}^{\text{top-}k} + \lambda_{\mathrm{lm}}\mathcal{L}_{\mathrm{lm}} + \beta\,\mathcal{L}_{\mathrm{KL}}, \\ \mathcal{L}_{\mathrm{KL}} &= \mathbb{E}_m\left[\mathrm{KL}\big(q_\phi(a \mid E_m) \,\|\, p(a \mid M_{m-1})\big) + \mathrm{KL}\big(q_\phi(c \mid E_m) \,\|\, p_{\mathrm{mem}}(c)\big)\right]. \end{aligned} \tag{10}$$

The first term replaces the likelihood of the evidence lower bound (ELBO) by top-k distillation over all positions, and $\mathcal{L}_{\mathrm{lm}}$ is the cross-entropy on answer tokens. The KL terms are the ELBO's prior terms with each slot mixture replaced by its moment-matched Gaussian: $p(a \mid M_{m-1})$ is the address mixture under the Dirichlet mean, and $p_{\mathrm{mem}}(c)$ is the read of Eq. (6), so they measure the surprise of new evidence under the memory's prediction (Appendix P). We use $\lambda_{\mathrm{lm}} = 0.05$ and $\beta = 10^{-3}$ (Appendix A.2).

4 Experiments

4.1 Experimental Setup

We attach BeliefMem to frozen Qwen2.5-Instruct backbones (Qwen Team, 2024) at 3B, 7B, and 14B. The local route uses 128 attention sinks and an 8,064-token sliding window; BeliefMem adds one memory layer with 256 slots and 256-token evidence blocks. The adapter is trained for 100 steps on a 600-example pool of 420 ChatQA2 (Xu et al., 2025), 120 non-target LongBench, and 60 synthetic recall examples; checkpoints are selected on held-out examples from this pool, which has no exact question or context overlap with the evaluation tasks (Appendix A.2). Our primary benchmark, **LB6**, averages DuReader, HotpotQA, MuSiQue, NarrativeQA, QMSum, and TriviaQA (Bai et al., 2024) over 200 examples each, with inputs up to 32k tokens. We compare with the local route (attention sinks plus sliding-window attention, Sinks+SWA), all nine official AHN checkpoints (Mamba2, DeltaNet (DN), and Gated DeltaNet (GDN) memories at each scale; Fang et al., 2025), and the full-attention teacher, all evaluated under the same examples, prompts, input limits, greedy decoding, and scorer. For transfer we evaluate all 503 LongBench v2 questions (Bai et al., 2025) and InfiniteBench (Zhang et al., 2024) with 128k inputs.

4.2 Main Results

BeliefMem is the strongest compressed method at every scale (Table 1). It improves over Sinks+SWA by **2.72, 4.05, and 2.65** points at 3B, 7B, and 14B, and over the strongest AHN checkpoint by **1.00, 1.48, and 1.13** points; every individual BeliefMem run exceeds the local route (Appendix B). It recovers 83%, 88%, and 59% of the gap between the local route and full attention, far more than any AHN variant, and even exceeds the teacher on DuReader at 7B/14B and on TriviaQA at 3B/7B. The largest gains over the local route are on DuReader and HotpotQA ($+5.40/+3.62$ at 3B, $+6.40/+7.11$ at 7B, $+6.17/+3.49$ at 14B), tasks whose supporting passages are spread across the input. MuSiQue remains the hardest case for all compressed methods.

The adaptation pool alone does not explain the gap: continuing AHN-GDN on the same 600 examples leaves BeliefMem ahead by 1.76 and 2.28 points at 3B and 7B (Appendix B.2).

Table 1: **LongBench results (LB6) on three Qwen2.5-Instruct backbones.** All methods are evaluated locally with identical examples, prompts, input limits, decoding, and scorer. NQA: NarrativeQA; HQA: HotpotQA; MuS: MuSiQue; DuR: DuReader (Chinese); QMS: QMSum; TQA: TriviaQA. BeliefMem (shaded) is the mean of three training runs (± standard deviation, SD, of the average); AHN rows are the nine official checkpoints. **Bold**: best and underline: second best among compressed methods; full attention is the teacher and upper reference.

	Single-doc	Multi-doc QA			Summ.	Few-shot	
Method	NQA	HQA	MuS	DuR	QMS	TQA	Avg. ↑
Qwen2.5-3B-Instruct							
Sinks+SWA	15.31	43.70	16.54	23.00	20.94	85.94	34.24
AHN-Mamba2	16.87	42.50	18.31	24.88	21.63	85.23	34.90
AHN-DN	19.15	42.49	**20.27**	25.55	**22.11**	86.17	35.96
AHN-GDN	**19.93**	42.04	19.37	25.83	21.78	85.56	35.75
BeliefMem	19.67	**47.32**	17.65	**28.40**	20.42	**88.30**	**36.96**±0.94
Full attention	19.70	46.64	20.10	28.58	22.98	87.10	37.52
Qwen2.5-7B-Instruct							
Sinks+SWA	22.54	51.07	21.84	25.33	21.45	88.34	38.43
AHN-Mamba2	24.95	52.47	27.74	26.75	21.55	89.16	40.44
AHN-DN	24.59	53.91	**29.24**	26.81	**21.98**	89.49	41.00
AHN-GDN	23.71	53.79	26.53	26.83	21.86	89.64	40.39
BeliefMem	**25.19**	**58.18**	27.30	**31.73**	21.93	**90.57**	**42.48**±0.28
Full attention	28.19	57.97	30.09	29.45	23.16	89.33	43.03
Qwen2.5-14B-Instruct							
Sinks+SWA	23.09	55.92	29.00	25.14	21.02	88.65	40.47
AHN-Mamba2	23.47	56.96	30.46	26.85	22.06	88.38	41.36
AHN-DN	22.55	58.37	**32.49**	26.85	**22.17**	87.01	41.57
AHN-GDN	**25.21**	58.14	31.51	26.63	22.11	88.35	41.99
BeliefMem	24.47	**59.41**	31.28	**31.31**	22.02	**90.22**	**43.12**±0.66
Full attention	26.03	62.11	38.19	29.01	23.65	90.69	44.95

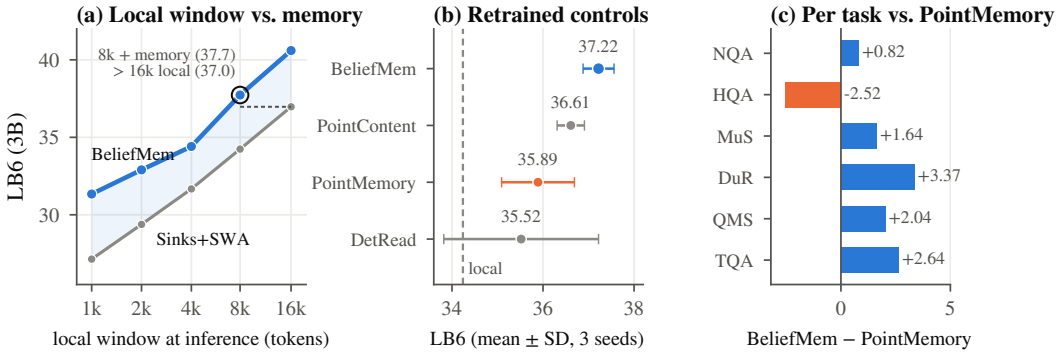


Figure 3: **Where the gain comes from.** (a) One 3B adapter (trained with the 8k window) evaluated with other windows; dashed: 16k local route. (b) Retrained 3B memory controls (three seeds each). (c) Per-task difference between BeliefMem and PointMemory.

4.3 Belief Memory versus Point Memory

Is the gain due to beliefs or to the query-conditioned interface? We retrain a **PointMemory** control with the same slots, width, decoder, data, and selection, but every belief replaced by a point estimate (no variances, natural parameters, Dirichlet counts, PoE, or KL; deterministic read and write). Two partial controls remove one pathway each: **DetRead** replaces the predictive read and PoE by a deterministic read, and **PointContent** uses point content with Gaussian addresses (Appendix B.1).

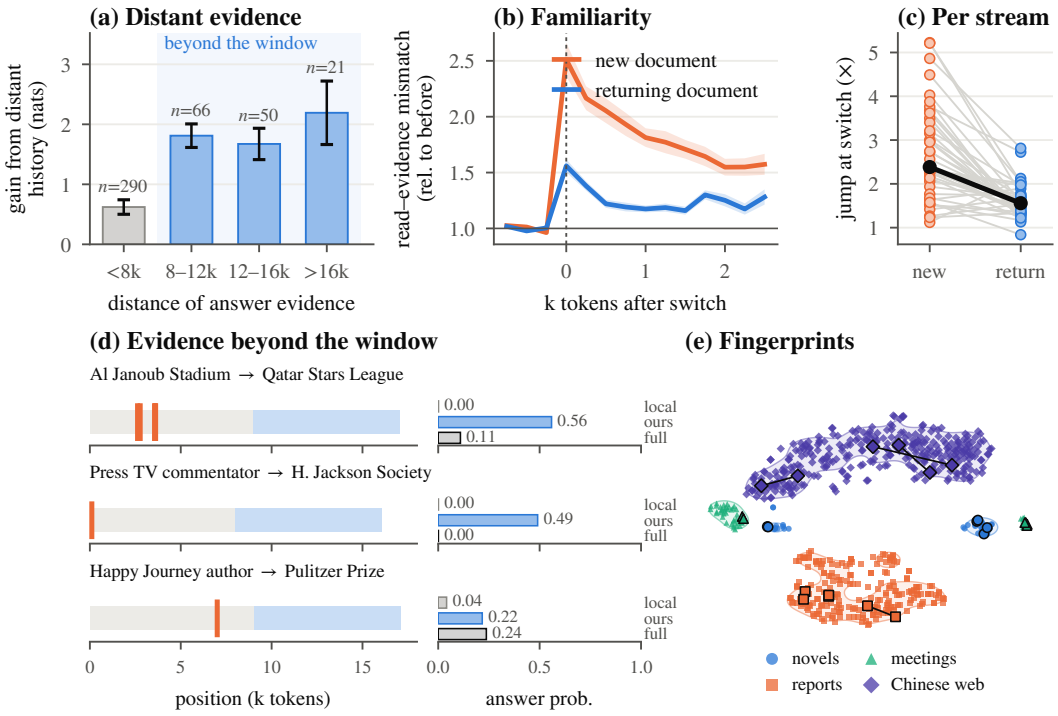


Figure 4: **What the trained 3B memory does on real text.** (a) Extra reduction in answer negative log-likelihood (NLL) from the full history, by distance of the last answer occurrence (difference-in-differences; mean ± standard error, SE). (b) Read–evidence mismatch around a switch to a new document or back to one read 12k tokens earlier (40 streams); (c) jump per stream. (d) Supporting passage (orange) outside the window (blue) and gold-answer probability. (e) t-SNE of final states for both 4k halves of 321 documents; outlined pairs are the halves of three random documents per source.

BeliefMem outperforms PointMemory by **1.33** LB6 points, and the gain is positive in every paired seed (**+0.49, +1.41, +2.10**; Appendix Figure 7), although PointMemory has only 7% fewer parameters (6.56M vs. 7.08M). The belief pathway wins on five of six tasks, most on DuReader (+3.37), TriviaQA (+2.64), and QMSum (+2.04), while PointMemory is better on HotpotQA (Figure 3b,c). Removing the predictive read alone costs 1.70 points, with losses on all six tasks, and point content costs 0.61 points; the full belief pathway is the best and the most stable configuration (SD 0.34).

4.4 Where Does the Memory Help?

Does the memory carry evidence from beyond the window? On 813 LongBench QA prompts longer than the window (seven datasets) we score the gold answer token by token through the KV cache, with the memory on and off, for the full prompt and for the prompt truncated to the sinks plus the last 8,064 tokens. The difference between the memory's benefit with and without the distant history isolates its contribution from that history, net of the local route and of answer-style adaptation. When the answer string occurs inside the window, the distant history adds only 0.62 nats; when it occurs 8–16k tokens back, it adds **1.7–1.8 nats**, and beyond 16k **2.2 nats**, with a positive effect on 82–91% of these prompts (Figure 4a): the benefit grows where the window cannot see the evidence.

Familiarity. We interleave four real documents as $A_1B_1C_1D_1A_2B_2C_2D_2$, where X_1 and X_2 are consecutive 4k-token spans of document X, and track the mismatch $1 - \cos(\mu_m^{\text{mem}}, \mu_m^c)$ between the memory read and the incoming evidence. Entering a new document raises the mismatch by 2.4× (median), whereas returning to a document already read raises it by only 1.6× and decays within 1k tokens (Figure 4b,c; 88% of streams, Wilcoxon $p < 10^{-7}$): read content stays familiar after 12k tokens of other text.

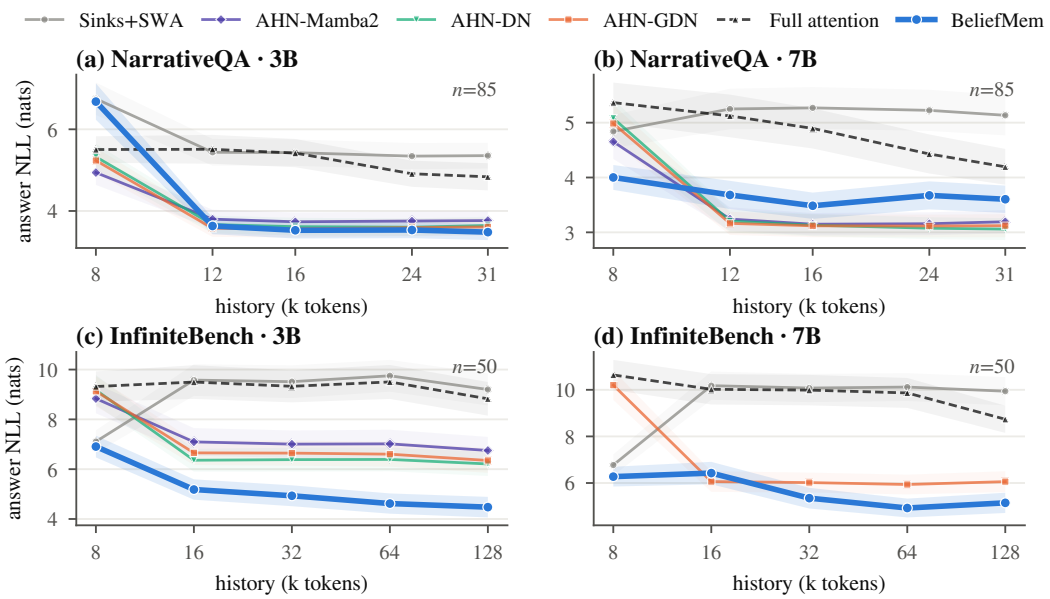


Figure 5: **Answer NLL versus available history.** Every method sees the first 128 and the last N tokens of the same questions (mean $\pm$ SE); InfiniteBench En.QA runs to 128k with YaRN (Peng et al., 2024). Compressed methods use the 8k window.

4.5 Does More History Help?

With growing history (Figure 5; Appendix F.2), Sinks+SWA does not improve, whereas the memory methods turn older text into lower answer NLL. BeliefMem reaches the lowest NLL on NarrativeQA at 3B with the full 32k prompt (3.48) and on InfiniteBench at every history length from 32k to 128k at both scales; at 128k it is 1.7 nats below the best AHN variant at 3B and 0.9 nats below AHN-GDN at 7B, and at 3B its NLL keeps falling up to 128k (6.91 to 4.48). On NarrativeQA at 7B, AHN reaches a lower NLL (3.06–3.19 vs. 3.60), although BeliefMem's NarrativeQA task score remains higher (Table 1; all values in Table 18).

4.6 What Does the Memory Retain?

Cases. When the supporting passage lies 10–16k tokens back, the local route gives the gold answer at most 0.04 probability, while BeliefMem recovers it (Figure 4d; cases against AHN in Appendix D). **Fingerprints.** We run the memory separately over two disjoint 4k-token halves of every unique LongBench context of at least 8k tokens from six sources (686 documents) and summarize each final state by its precision-weighted content mean. The two halves of a document retrieve each other with 23% top-1 accuracy among 686 documents (chance 0.15%; median rank 22), and the source is identified in 80% of cases (chance 17%; Figure 4e). Even the precision profile alone, which a point memory does not have, identifies the source in 66% of cases, so even the evidence statistics carry information about what was read. **Order invariance.** Natural-parameter updates add evidence, so the state should not depend on its order. Reordering the out-of-window passages of a multi-hop question leaves the final state almost unchanged (mean cosine 0.98 across six orders, versus 0.00 between different questions; 0.96 at 7B; Figure 9).

4.7 Transfer to Broader and Longer Contexts

LongBench v2. On all 503 questions (Table 2), BeliefMem is the best compressed method at 7B and 14B ($+1.39$ and $+0.60$ over AHN-GDN; $+6.56$ and $+6.76$ over the local route), with the largest 7B gains on dialogue history, code repositories, and structured data. At 3B it trails AHN, mainly on code repositories (Appendix C.1). Full attention scores 29.22, 30.42, and 36.18 at 3B, 7B, and 14B; BeliefMem exceeds it at 7B.

Table 2: **LongBench v2 (all 503 questions, accuracy %).** Question counts per category in parentheses; Overall is accuracy over all questions; ICL: in-context learning. Bold marks the best compressed method per scale and column; BeliefMem is shaded. †Full attention is the teacher reference at every scale; it is given a 512-token answer budget because it often reasons at length before stating an option.

Scale	Method	Code (50)	Long ICL (81)	Structured (33)	Dialogue (39)	Multi-doc (125)	Single-doc (175)	Overall (503)
3B	Sinks+SWA	18.00	27.16	27.27	17.95	21.60	33.14	26.24
	AHN-GDN	**24.00**	25.93	27.27	17.95	**22.40**	**37.71**	**28.43**
	BeliefMem	10.00	**29.63**	**39.39**	**25.64**	21.60	29.71	26.04
	Full attention†	18.00	35.80	42.42	15.38	22.40	34.86	29.22
7B	Sinks+SWA	28.00	28.40	6.06	10.26	20.80	35.43	26.04
	AHN-GDN	24.00	**32.10**	18.18	17.95	**30.40**	**38.86**	31.21
	BeliefMem	**34.00**	30.86	**24.24**	**30.77**	**30.40**	36.57	**32.60**
	Full attention†	28.00	28.40	27.27	23.08	28.00	36.00	30.42
14B	Sinks+SWA	22.00	25.93	3.03	15.38	30.40	28.00	25.05
	AHN-GDN	26.00	23.46	15.15	**35.90**	**36.00**	**34.86**	31.21
	BeliefMem	**32.00**	**29.63**	**21.21**	**35.90**	33.60	32.57	**31.81**
	Full attention†	28.00	34.57	33.33	41.03	36.00	38.86	36.18

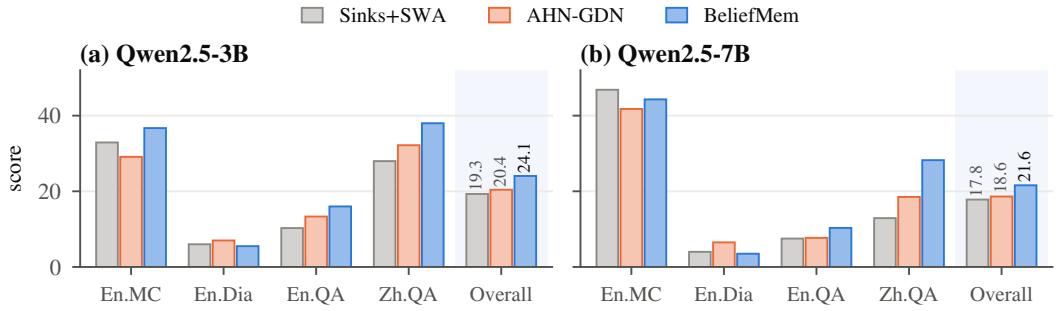


Figure 6: **InfiniteBench with 128k inputs.** Same eligible examples for all methods (79 En.MC, 200 En.Dia, 112 En.QA, 27 Zh.QA); Overall is the unweighted mean (exact values in Table 10).

InfiniteBench. With 128k inputs, BeliefMem exceeds AHN-GDN by 3.64 and 2.97 macro-average points at 3B and 7B and leads on both QA tasks at both scales (Figure 6; Appendix C.2).

4.8 Training Cost and Robustness

BeliefMem trains 0.11–0.23% of the backbone parameters, 35–44% fewer than AHN at 7B (Table 9). A 7B adapter trains on 600 examples in 27 minutes on one GPU, whereas AHN reports 10 hours on 32 A100 GPUs with 1B tokens (Fang et al., 2025): about 700 times fewer GPU-hours, on different GPUs. The memory helps at every local window from 1k to 16k tokens (+2.7 to +4.2 LB6), and the 8k model with memory exceeds the 16k local route (37.73 vs. 36.97; Figure 3a). The gains are also robust to the slot count and the data mixture (Appendix E).

5 Conclusion

We cast bounded long-context memory as online belief maintenance. BeliefMem keeps Gaussian slot beliefs beside a frozen sliding-window model, couples reading and writing through one posterior responsibility, and lets accumulated precision decide how far the read corrects the local route. Training 0.11–0.23% of the backbone parameters, it is the strongest compressed method on LongBench from 3B to 14B, the belief pathway beats a retrained point memory in every paired seed, and its benefit concentrates on evidence beyond the window. A fixed bank of diagonal beliefs still compresses history, so verbatim recall beyond the window remains weaker than with full attention (Appendix Q); memories that grow with the input or combine beliefs with retrieval are natural next steps.

Reproducibility Statement

Section 3 and Algorithm 1 specify the memory transition; Appendix A.2 lists all hyperparameters and training settings; Appendix A.1 identifies every experimental series and its reference run; and the derivations are in Appendices I onward. The adaptation pool is built from public datasets (ChatQA2, LongBench) plus programmatically generated recall examples, and all benchmarks are public. We will release code, the data-construction scripts, and adapter checkpoints.

AI Use Statement

We used LLM-based assistants as auxiliary tools to support implementation and experimentation, literature search, and writing. All reported numbers were computed by scripts from logged model outputs. The authors reviewed and verified all code, citations and claims, and take full responsibility for the content of this paper.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 3119–3137, 2024.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards Deeper Understanding and Reasoning on Realistic Long-context Multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3639–3664, 2025. doi: 10.18653/v1/2025.acl-long.183. URL https://aclanthology.org/2025.acl-long.183/.

Rachit Bansal, Aston Zhang, Rishabh Tiwari, Lovish Madaan, Venkata Sai Surya Subramanyam Duvvuri, Devvrit Khatri, David Brandfonbrener, David Alvarez-Melis, Prajjwal Bhargava, Mihir Kale, et al. Let's (not) just put things in context: Test-time training for long-context llms. In *International Conference on Learning Representations*, volume 2026, pp. 112130–112153, 2026.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to Memorize at Test Time. *arXiv preprint arXiv:2501.00663*, 2024.

Ali Behrouz, Zeman Li, Yuan Deng, Peilin Zhong, Meisam Razaviyayn, and Vahab Mirrokni. Memory caching: Rnns with growing memory. *arXiv preprint arXiv:2602.24281*, 2026.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.

Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. Recurrent memory transformer. *Advances in neural information processing systems*, 35:11079–11091, 2022.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting Language Models to Compress Contexts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 2978–2988, 2019.

Payel Das, Subhajit Chaudhury, Elliot Nelson, Igor Melnyk, Sarathkrishna Swaminathan, Sihui Dai, Aurélie Lozano, Georgios Kollias, Vijil Chenthamarakshan, Jiri Navratil, Soham Dan, and Pin-Yu Chen. Larimar: Large Language Models with Episodic Memory Control. *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Yunhao Fang, Weihao Yu, Shu Zhong, Qinghao Ye, Xuehan Xiong, and Lai Wei. Artificial hippocampus networks for efficient long-context modeling. *arXiv preprint arXiv:2510.07318*, 2025.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context Autoencoder for Context Compression in a Large Language Model. In *International Conference on Learning Representations*, 2024.

Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *First conference on language modeling*, 2024.

Zexue He, Leonid Karlinsky, Donghyun Kim, Julian McAuley, Dmitry Krotov, and Rogerio Feris. CAMELoT: Towards Large Language Models with Training-Free Consolidated Associative Memory. *arXiv preprint arXiv:2402.13449*, 2024.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations*, volume 2024, pp. 54107–54157, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM Knows What You Are Looking for Before Generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Yuhan Liu, Hanchen Li, Yihua Cheng, Siddhant Ray, Yuyang Huang, Qizheng Zhang, Kuntai Du, Jiayi Yao, Shan Lu, Ganesh Ananthanarayanan, et al. CacheGen: KV Cache Compression and Streaming for Fast Large Language Model Serving. *arXiv preprint arXiv:2310.07240*, 2023.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to Compress Prompts with Gist Tokens. In *Advances in Neural Information Processing Systems*, 2023.

Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. Leave No Context Behind: Efficient Infinite Context Transformers with Infini-attention. *arXiv preprint arXiv:2404.07143*, 2024.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, et al. RWKV: Reinventing RNNs for the Transformer Era. *arXiv preprint arXiv:2305.13048*, 2023.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *International Conference on Learning Representations*, 2024.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*, 2019.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. *Advances in neural information processing systems*, 36:74530–74543, 2023.

Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, et al. Memoryllm: Towards self-updatable large language models. *arXiv preprint arXiv:2402.04624*, 2024.

Yu Wang, Dmitry Krotov, Yuanzhe Hu, Yifan Gao, Wangchunshu Zhou, Julian McAuley, Dan Gutfreund, Rogerio Feris, and Zexue He. M+: Extending memoryllm with scalable long-term memory. *arXiv preprint arXiv:2502.00592*, 2025.

Yan Wu, Greg Wayne, Alex Graves, and Timothy Lillicrap. The Kanerva Machine: A Generative Distributed Memory. In *International Conference on Learning Representations*, 2018a.

Yan Wu, Greg Wayne, Karol Gregor, and Timothy Lillicrap. Learning Attractor Dynamics for Generative Memory. In *Advances in Neural Information Processing Systems*, 2018b.

Peng Xu, Wei Ping, Xianchao Wu, Chejian Xu, Zihan Liu, Mohammad Shoeybi, and Bryan Catanzaro. ChatQA 2: Bridging the Gap to Proprietary LLMs in Long Context and RAG Capabilities. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=4xdiqenzm1.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated Linear Attention Transformers with Hardware-Efficient Training. *arXiv preprint arXiv:2312.06635*, 2023.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. MemAgent: Reshaping Long-Context LLM with Multi-Conv RL-based Memory Agent. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2507.02259.

Guibin Zhang, Muxin Fu, and Shuicheng Yan. Memgen: Weaving generative latent memory for self-evolving agents. In *International Conference on Learning Representations*, volume 2026, pp. 22555–22588, 2026.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. Extending Long Context Evaluation Beyond 100K Tokens, 2024.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in neural information processing systems*, 36:34661–34710, 2023.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing Large Language Models with Long-Term Memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024.

A Experimental Details

A.1 Experimental series

Our results come from several independently trained series, summarized in Table 3. Each series retrains its own BeliefMem reference with the same recipe, so reference scores differ slightly across series; on LB6 they range from 36.96 to 37.27 at 3B, within the run-to-run SD of 0.34–0.94. Comparisons are always made within a series, and all series report LB6.

Table 3: **Experimental series.** All series except Cross-scale use Qwen2.5-3B (short-context checks also 7B). Reference is the BeliefMem score against which the conditions of the series are compared.

Series	Purpose	Seeds	Reference	Reported in
Cross-scale	Main comparison	3	LB6 36.96/42.48/43.12	Table 1
Memory controls	Belief vs. point memory	3	LB6 37.22	Figure 3
Slot sweep	Capacity	1	LB6 37.27	Table 12
Mixtures	Data recipe	3	LB6 36.96	Table 13
Pool size	600/800/1,000 examples	3	LB6 35.91	Table 14
Inference settings	Window, short context	1	LB6 37.73	Tables 15, 16
Probes	Inference-only analysis	–	answer NLL	Section 4.4

A.2 Hyperparameters and training

Table 4 lists all settings. The backbone is frozen; the adapter is trained with AdamW (weight decay 0.01, gradient clipping 1.0), a cosine schedule, batch size 1, bf16, and gradient checkpointing, with truncated backpropagation through four evidence blocks. To expose the memory to different amounts of out-of-window history, the student's window is sampled per example from $\{1, 2, 4, 8\}$k tokens with probabilities $0.1/0.2/0.3/0.4$; evaluation always uses the 8,064-token window. Top-k distillation is applied at every position of the sequence, whereas $\mathcal{L}_{\mathrm{lm}}$ is restricted to answer tokens. The KL weight is warmed up linearly over 1,000 steps, so within the 100 training steps its effective value grows to 10^{-4}.

Table 4: **Hyperparameters.** Shared values are listed once. α^0 is learned from its initial value.

	3B	7B	14B
Adapted layer	16	10	17
Slots K_s / D_a / D_c		256 / 128 / 128	
Encoder / decoder bottleneck		512 / 512 (MLP)	
Block / sinks / window (tokens)		256 / 128 / 8,064	
ρ / initial α^0 / ϵ		0.01 / 1.0 / 10^{-6}	
Training examples / steps		600 / 100 (one epoch)	
Learning rate	1.5×10^{-4}	10^{-4}	5×10^{-5}
Max. training length		24,576 tokens	
Distillation top-k / λ_{lm}		256 / 0.05	
KL weight β		10^{-3} (linear warm-up, 1,000 steps)	
GPUs / wall-clock	1	1 / 27 min	4 (FSDP) / 55 min
Trainable parameters	7.08M (0.23%)	12.00M (0.16%)	16.91M (0.11%)

Adaptation pool. The 600 training examples comprise 420 ChatQA2 examples, 120 examples from 15 LongBench datasets that are disjoint from the six LB6 tasks, and 60 programmatically generated recall examples; a further 90 examples form the validation split used for checkpoint selection. Exact question, context, and question–context overlap with the six target datasets is zero.

B Detailed Results

AHN checkpoints. We evaluate the official AHN-Mamba2, AHN-DN, and AHN-GDN checkpoints at 3B, 7B, and 14B with AHN's native memory implementation and the same LB6 examples, prompts,

Table 5: **Individual BeliefMem runs on LB6** (cross-scale series). Every run exceeds both Sinks+SWA (34.24/38.43/40.47) and the strongest AHN checkpoint at the same scale (35.96/41.00/41.99).

Scale	Run 1	Run 2	Run 3	Mean	SD
3B	37.94	36.05	36.89	36.96	0.94
7B	42.76	42.48	42.21	42.48	0.28
14B	43.69	42.39	43.28	43.12	0.66

input limits, decoding, and scorer as BeliefMem (Table 1). Our local averages are within 0.26 points of the published AHN averages for all nine checkpoints.

B.1 Retrained memory controls

All four conditions in Figure 3 are trained from scratch with the same frozen Qwen2.5-3B, 256 slots, latent width, query-conditioned decoder, adaptation pool, validation-based selection, and seeds (42, 1337, 3407). Table 6 specifies them and Table 7 gives the per-task differences. Figure 7 shows the paired LB6 scores for each seed.

Table 6: **Retrained memory controls on Qwen2.5-3B** (LB6, mean ± SD over three seeds). Each control removes belief components from BeliefMem while keeping slots, width, decoder, data, and selection fixed; PointMemory also drops $\mathcal{L}_{\mathrm{KL}}$. Δ is the change relative to BeliefMem.

Condition	Gaussian address	Gaussian content, belief write	Predictive read + PoE	LB6 ↑	Δ
BeliefMem	✓	✓	✓	**37.22**±0.34	–
PointContent	✓	✗	✗	36.61±0.30	−0.61
PointMemory	✗	✗	✗	35.89±0.80	−1.33
DetRead	✓	✓	✗	35.52±1.70	−1.70

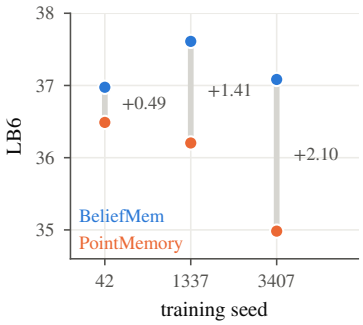


Figure 7: **Paired seeds.** LB6 of BeliefMem (blue) and PointMemory (orange) trained with the same seed; the gain is positive for every pair.

Table 7: **Per-task differences in the control series.** Positive values favor BeliefMem (task abbreviations as in Table 1).

BeliefMem −	NQA	HQA	MuS	DuR	QMS	TQA	LB6
PointMemory	+0.82	−2.52	+1.64	+3.37	+2.04	+2.64	**+1.33**
DetRead	+2.31	+0.16	+1.62	+4.06	+0.58	+1.45	**+1.70**

B.2 Same-pool AHN adaptation and training footprint

We continue training the official AHN-GDN checkpoints on exactly the same 600 examples, keeping AHN's objective and memory implementation and the backbone frozen (Table 8). The additional data slightly lowers AHN's LB6 at both scales, so access to the pool does not reproduce BeliefMem's benefit. Table 9 compares trainable parameters. For reference, AHN reports training on about 1B ChatQA2 tokens for about ten hours on 32 A100 GPUs at 7B (Fang et al., 2025), whereas a 7B BeliefMem adapter trains on 600 examples in about 27 minutes on one RTX PRO 6000, about 700

times fewer GPU-hours (0.45 vs. 320); the GPUs and objectives differ, so this ratio is not a controlled speedup.

Table 8: **AHN-GDN continued on the BeliefMem adaptation pool** (LB6).

Scale	AHN-GDN	AHN-GDN + pool	BeliefMem	Gain over adapted AHN
3B	35.75	35.20	36.96	+1.76
7B	40.39	40.20	42.48	+2.28

Table 9: **Trainable parameters.** AHN counts are those reported by Fang et al. (2025).

Scale	Method	Trainable parameters	Fraction of backbone
3B	AHN-Mamba2 / DN / GDN	11.9M / 11.8M / 13.0M	about 0.4%
	BeliefMem	7.08M	0.23%
7B	AHN-Mamba2 / DN / GDN	18.6M / 18.5M / 21.3M	0.2–0.3%
	BeliefMem	12.00M	0.16%

C Transfer Benchmarks

C.1 LongBench v2

Table 2 (Section 4.7) uses the official zero-shot prompt, middle truncation to 120k tokens, and the official answer parser for all methods and scales; the local route, AHN, and BeliefMem share the 8,064-token window. At 14B, the evaluated BeliefMem adapter is one of the three cross-scale runs. Full attention often reasons at length before stating an option, so we give it a 512-token answer budget at every scale; it still leaves 12, 35, and 18 of 503 answers unparsed at 3B, 7B, and 14B. With 120k inputs neither the 14B local route nor 14B full attention fits on one GPU, so we evaluate both with two-GPU layer sharding; on a 32k check the sharded run reproduces the single-GPU outputs exactly. Like the 7B local route, the 14B local route often fails to state an option within the answer budget (115 of 503 answers unparsed). At 3B, BeliefMem's gains on long in-context learning, structured data, and dialogue do not offset its weaker code-repository result (5 of 50 correct vs. 12 for AHN); only five repository examples appear in the adaptation pool. At 7B and 14B, BeliefMem answers more repository questions than AHN (17 vs. 12 and 16 vs. 13).

C.2 InfiniteBench and 128k inputs

Table 10: **InfiniteBench with 128k inputs.** All methods use the same eligible 79 En.MC, 200 En.Dia, 112 En.QA, and 27 Zh.QA examples; Overall is the unweighted mean of the four tasks. Bold marks the best method per scale and column; BeliefMem is shaded.

		English			Chinese	
Scale	Method	En.MC	En.Dia	En.QA	Zh.QA	Overall
3B	Sinks+SWA	32.91	6.00	10.28	27.98	19.29
	AHN-GDN	29.11	**7.00**	13.33	32.18	20.41
	BeliefMem	**36.71**	5.50	**16.00**	**37.99**	**24.05**
7B	Sinks+SWA	**46.84**	4.00	7.51	12.91	17.81
	AHN-GDN	41.77	**6.50**	7.70	18.51	18.62
	BeliefMem	44.30	3.50	**10.32**	**28.23**	**21.59**

The 128k route uses 128 sinks and a 32,640-token sliding window, together with BeliefMem's bounded state; Table 10 uses the same eligible rows for every method. At 14B, four-GPU layer sharding keeps a single online memory state across 128k inputs; on the evaluated rows BeliefMem scores 45.45 (En.MC, 77), 7.00 (En.Dia, 200), 10.69 (En.QA, 351), and 21.26 (Zh.QA, 189). Local AHN evaluation at 14B/128k was not possible because its gated-delta kernel exceeds the shared-memory limit of our GPUs.

D CASE STUDIES AGAINST AHN

Table 11 shows questions that BeliefMem answers correctly and AHN-GDN does not, chosen among English extractive examples where the gold answer occurs only outside the local route's view (more than 8,064 tokens before the end of the prompt). LB6 cases use the three cross-scale BeliefMem runs and the official AHN checkpoints on the same prompts. Such cases are selected: counting examples whose scores differ by at least 0.5, BeliefMem wins/loses against AHN-GDN on 11/3 (HotpotQA), 7/4 (MuSiQue), and 6/3 (NarrativeQA) examples at 7B, and on 9/6, 5/8, and 2/5 at 14B.

Table 11: **Evidence beyond the window: BeliefMem vs. AHN-GDN.** Distances are in tokens before the end of the prompt; ✓/✗ mark correct and wrong answers (MuSiQue case: F1 0.8).

(1) MuSiQue, Qwen2.5-7B	prompt 16.5k tokens; answer evidence 13.4k before the end
Question	What character in *A League of Their Own* was played by the actress who played Thelma in *Thelma and Louise*?
Evidence	"… director Penny Marshall talked about her initial interest in Demi Moore for the part of **Dottie Hinson** …"
BeliefMem	Dorothy "Dottie" Hinson (all three runs) ✓
AHN-GDN	Diane Keaton ✗
Sinks+SWA	Diane Katin ✗
(2) NarrativeQA, Qwen2.5-7B	prompt 31.9k tokens; nearest mention 15.7k before the end
Question	Where in Pennsylvania was Buck Rodgers investigating reports of phenomena?
Evidence	"The squadron of seven ships … to investigate the recent destruction of the GK-984 in the **Wyoming Valley** …"
BeliefMem	Wyoming Valley (all three runs) ✓
AHN-GDN	Philadelphia ✗
Sinks+SWA	Beaver Valley ✗
(3) HotpotQA, Qwen2.5-14B	prompt 16.1k tokens; second-hop passage 8.6k before the end
Question	What was the home city of the author of the famous novel tracing the life of a male roe deer from his birth onward?
Evidence	"… He also wrote for nearly all the major newspapers of **Vienna**. … His best remembered work is *Bambi* (1923)."
BeliefMem	Vienna (all three runs) ✓
AHN-GDN	Zürich ✗
Sinks+SWA	"None of the given passages contain information about …" ✗
(4) InfiniteBench En.MC, Qwen2.5-7B	prompt 128k tokens; evidence 120k before the end
Question	What country has just taken possession of the islands? (A) Spain (B) France (C) Portugal (D) The United States
Evidence	"… I saw nothing but the tri-coloured flag of **France** trailing over the stern of six vessels …"
BeliefMem	B. France ✓
AHN-GDN	"To answer the question based on the provided information:" (no option) ✗
Sinks+SWA	"The passage does not mention any country taking possession of the islands." ✗

E ROBUSTNESS

Tables 12–14 summarize these studies (default settings shaded). Every slot count exceeds the local route (34.24) by at least 1.67 points, and the validation-selected default of 256 slots is the best. Alternative mixtures, including ChatQA2 only and no synthetic data, stay within 0.31 points of one another, so the gain does not depend on a tuned recipe. The pool-size series trains for one epoch (600, 800, and 1,000 updates); enlarging the pool to 1,000 examples preserves the gain, which indicates that a small pool already suffices for the adapter.

BeliefMem improves LB6 by 2.7–4.2 points at every local window from 1k to 16k tokens, and the 8k-window model with memory exceeds the 16k-window local model (37.73 vs. 36.97). On short-context tasks, where the memory has little history to contribute, BeliefMem changes accuracy by -0.23 to $+2.13$ points (Table 16), so the memory does not degrade them.

Table 12: **Slot count** (LB6).

K_s	LB6
32	36.83
64	35.91
128	36.65
256	**37.27**
512	36.45
1024	36.98

Table 13: **Data mixture** (LB6, three-seed means). Proportions are ChatQA2 / LongBench / synthetic.

Mixture (%)	LB6
70/20/10 (default)	**36.96**
70/30/0	36.36
100/0/0	36.13
50/40/10	36.44

Table 14: **Pool size** (LB6, three seeds; local 34.24).

Examples	LB6
600	35.91$\pm$1.00
800	**35.97**$\pm$0.35
1,000	35.55$\pm$0.71

Table 15: **Local window at inference** (3B, LB6). The adapter is trained once.

Window	Local	BeliefMem	Δ
1,024	27.14	31.34	+4.20
2,048	29.38	32.90	+3.51
4,096	31.67	34.41	+2.74
8,064	34.24	37.73	+3.49
16,384	36.97	40.60	+3.64

Table 16: **Short-context tasks** (accuracy %). GSM8K: all 1,319 test problems, zero-shot chain of thought with the Qwen2.5 prompt, greedy decoding. Bold marks the better method per scale; BeliefMem is shaded.

	Method	ARC-C	HellaSwag	GSM8K
3B	Sinks+SWA	77.56	**75.30**	84.61
	BeliefMem	**79.69**	75.10	**85.06**
7B	Sinks+SWA	87.80	81.10	**90.83**
	BeliefMem	**88.05**	**82.00**	90.60

F Belief-State Diagnostics

F.1 Distant-history probe

The probe of Section 4.4 uses the formal 3B adapter and the first 200 examples of HotpotQA, 2WikiMQA, MuSiQue, NarrativeQA, TriviaQA, Qasper, and MultiFieldQA-en whose prompts exceed the local view (813 prompts, mean length 17.2k tokens). Prompts are built and tail-truncated to 32k tokens exactly as in evaluation. For the gold answer y we compute the per-token negative log-likelihood $\mathrm{NLL}(y)$ by prefilling the prompt and then feeding the answer one token at a time through the KV cache, so each answer token is its own evidence cycle. With Loc and Bel denoting the memory disabled and enabled on the same weights, and $\bar{x}$ the prompt truncated to the 128 sink tokens plus the last 8,064 tokens,

$$\mathrm{DiD} = \big[\mathrm{NLL}_{\mathrm{Loc}}(y \mid x) - \mathrm{NLL}_{\mathrm{Bel}}(y \mid x)\big] - \big[\mathrm{NLL}_{\mathrm{Loc}}(y \mid \bar{x}) - \mathrm{NLL}_{\mathrm{Bel}}(y \mid \bar{x})\big]. \tag{11}$$

Differencing within each input removes the adapter's effect that does not depend on history, and differencing across inputs removes the change of the local route itself (whose receptive field grows with depth). The distance of the evidence is the number of tokens between the last case-insensitive occurrence of a gold answer string in the context and the end of the prompt; yes/no answers and answers shorter than three characters are not matched. Table 17 reports the effect per dataset.

F.2 History-scaling protocol

For Figure 5 we use the 85 of the first 100 NarrativeQA prompts (LongBench) that fill the 32k input budget and the first 50 InfiniteBench En.QA prompts, which exceed 128k tokens, so every history length is available for every question. For a budget N, every method receives the first 128 tokens and the last $N - 128$ tokens of the same prompt, so the question and the nearest context are always visible and only the amount of older history changes. The gold answer is scored causally: BeliefMem, Sinks+SWA, and full attention feed the answer one token at a time through the KV cache, and AHN, whose memory is updated token by token, scores the answer in its official cached prefill. All methods share the 8,064-token window and 128 sinks except full attention, and InfiniteBench runs use YaRN (factor 4) for every method. Absolute NLLs also reflect how each adapter shapes its answer format, so they are best compared across history lengths within a method and at the longest history across methods. AHN-DN and AHN-Mamba2 were evaluated on NarrativeQA at both scales and on InfiniteBench at 3B.

Table 17: **Distant-history effect per dataset** (DiD in nats, higher means more benefit from the distant history). "Beyond" and "inside" split the prompts by whether the last answer occurrence lies beyond or inside the local window; $\% > 0$ is the share of prompts with a positive effect.

	All prompts			By evidence position		
Dataset	n	DiD	$\% > 0$	n beyond	DiD beyond	DiD inside
HotpotQA	172	1.10	63	31	1.95	0.12
2WikiMQA	52	1.05	52	7	0.56	0.63
MuSiQue	197	1.27	80	78	1.72	0.79
NarrativeQA	192	2.21	91	10	3.13	1.55
TriviaQA	137	0.96	82	5	1.44	1.52
Qasper	13	0.43	77	2	1.47	−0.03
MultiFieldQA-en	50	1.55	94	4	2.24	–
All	**813**	**1.39**	**78**	**137**	**1.82**	**0.62**

Table 18: **Answer NLL (nats, lower is better) by available history.** Each block uses the same questions at every history length. Bold: lowest and underline: second-lowest NLL among compressed methods per column; BeliefMem is shaded; full attention is shown for reference.

NarrativeQA, Qwen2.5-3B (n=85)	*8k*	*12k*	*16k*	*24k*	*31k*
Sinks+SWA	6.75	5.44	5.43	5.34	5.36
AHN-Mamba2	**4.94**	3.80	3.74	3.75	3.77
AHN-DN	5.33	3.67	3.62	3.60	3.64
AHN-GDN	5.24	**3.58**	3.56	3.59	3.61
BeliefMem	6.68	3.63	**3.53**	**3.53**	**3.48**
Full attention	5.51	5.51	5.42	4.91	4.84
NarrativeQA, Qwen2.5-7B (n=85)	*8k*	*12k*	*16k*	*24k*	*31k*
Sinks+SWA	4.84	5.25	5.27	5.22	5.13
AHN-Mamba2	4.65	3.24	3.15	3.16	3.19
AHN-DN	5.08	3.21	3.13	**3.08**	**3.06**
AHN-GDN	4.99	**3.16**	**3.12**	3.12	3.12
BeliefMem	**4.00**	3.68	3.48	3.67	3.60
Full attention	5.37	5.12	4.90	4.43	4.19
InfiniteBench En.QA, Qwen2.5-3B (n=50)	*8k*	*16k*	*32k*	*64k*	*128k*
Sinks+SWA	7.11	9.58	9.51	9.75	9.20
AHN-Mamba2	8.84	7.10	7.01	7.02	6.75
AHN-DN	9.18	6.36	6.38	6.39	6.21
AHN-GDN	9.13	6.65	6.64	6.60	6.35
BeliefMem	**6.91**	**5.18**	**4.93**	**4.62**	**4.48**
Full attention	9.32	9.50	9.33	9.50	8.83
InfiniteBench En.QA, Qwen2.5-7B (n=50)	*8k*	*16k*	*32k*	*64k*	*128k*
Sinks+SWA	6.77	10.18	10.08	10.12	9.94
AHN-GDN	10.20	**6.06**	6.02	5.94	6.06
BeliefMem	**6.27**	6.43	**5.35**	**4.93**	**5.15**
Full attention	10.64	10.02	9.99	9.87	8.74

Familiarity streams. Each of the 40 streams interleaves four documents drawn from different LongBench sources (novels, government reports, meetings, code, Chinese web text, news) as $A_1B_1C_1D_1A_2B_2C_2D_2$, where X_1, X_2 are consecutive 4,096-token spans of document X (32k tokens in total). For every evidence block we record the memory read mean μ_m^{mem} and the local content posterior mean μ_m^c. The jump at a switch is the mismatch in the first block of a span divided by the mismatch in the last block before it; Figure 4b normalizes by the mean of the three preceding blocks. New-document switches are B_1, C_1, D_1; returning switches are $A_2, \ldots, D_2$, each at least 12k tokens after the first span of the same document.

Belief-state trajectories. Figure 8 uses 80 streams of 32,768 tokens: 40 contiguous spans of single long documents and 40 concatenations of eight 4k-token spans from different documents. As evidence accumulates, the median predictive variance of the read contracts by more than an order of magnitude and the mean reliance gain rises from near zero to about 0.4; the two stream types give nearly identical trajectories, so the gain is governed by the amount of written evidence.

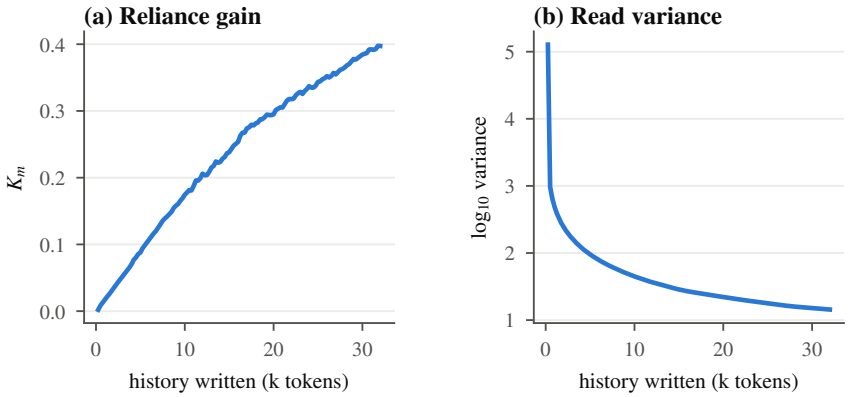


Figure 8: **Precision accumulation on real text** (3B; mean ± SE over 80 streams).

Order invariance. For Figure 9 we take 68 multi-hop questions (20 HotpotQA, 25 2WikiMQA, 23 MuSiQue), pad each context to about 24k tokens with passages from other questions of the same dataset, and evaluate it under six orders of the passages outside the local window; the passages in the last ∼9k tokens, which cover the local view, are held fixed. Each final memory state is standardized per dimension over all runs of a scale before computing cosines. Reordering leaves the state almost unchanged (mean cosine 0.98 at 3B and 0.96 at 7B), whereas states of different questions are uncorrelated (mean 0.00).

F.3 Controlled Gaussian-stream interventions

These module-level tests use the Qwen2.5-3B implementation with synthetic Gaussian observations. Means, counts, queries, and evidence order are held fixed while one belief statistic is changed (Table 19).

Table 19: **Effect of belief statistics on memory operations.** Each row changes one statistic while holding means, counts, queries, and evidence order fixed.

Intervention	Setting	Quantity	Before → after
Swap variances 0.1/2.0	two slots, equal means	responsibilities	0.74/0.26 → 0.26/0.74
Slot precision 0.1 → 100	obs. precision 5	fraction of mean moved	0.98 → 0.05
Top resp. 0.99 → 0.50	two disagreeing slots	between-slot variance	0.04 → 1.0
		reliance gain K	0.88 → 0.48
Repeated writes	consistent vs. noisy	memory variance	1.0 → 0.004 vs. 0.11

Variance changes assignment even when slot means agree with the query, accumulated precision resists overwriting, and disagreement between the slots being read lowers the reliance gain before decoding. These are the behaviors that a point memory must otherwise learn implicitly.

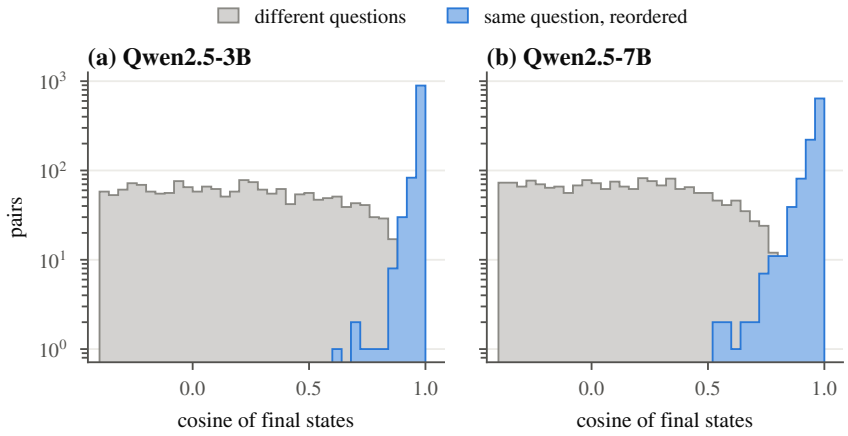


Figure 9: **Order invariance of the final memory state.** Cosine between final states of the same question under different passage orders (68×15 pairs) and between different questions ($68 \times 67/2$ pairs; first order of each question); log scale.

G Memory-Cycle Pseudocode and Causal Semantics

Algorithm 1 specifies the state transition; all reads of a cycle use the same previous state. The index m counts evidence cycles, whereas t indexes next-token predictions. E_m denotes observed evidence features, $\mathcal{T}_m$ the predictions allowed to consume that evidence, and $M^- = M_{m-1}$. K_s is the number of slots; K_m is the dimensionwise reliance gain. Superscripts a, c denote address and content. Vector variances are denoted v.

Algorithm 1 BeliefMem: encode, assign, read, fuse, and commit

Require: Previous state M^- ($M^- = M_0$, the learned initial state, at the start of a sequence); evidence block E_m; queries $\{q_t^{\text{mha}} : t \in \mathcal{T}_m\}$ of the positions served by this cycle; jitter $\epsilon > 0$; refresh rate $\rho \in [0, 1]$; learned base counts $\alpha^0 > 0$

Require: Every feature in E_m is a function of $x_{<t}$ for every scored prediction $t \in \mathcal{T}_m$; the evidence block has not already been committed

Encode pooled hidden/key and hidden/value features into (μ_m^a, v_m^a) and (μ_m^c, v_m^c), with positive variances.
Recover previous slot moments $\mu_k^u = \eta_k^u/\lambda_k^u$, $v_k^u = 1/\lambda_k^u$ for $u \in \{a, c\}$.
Compute $A_{m,k}, C_{m,k}$ for every slot using Eq. (5).
Compute $\tau_m = \text{softplus}(g_\omega(u_m))$ and $b_k = \psi(\alpha_k) - \psi(\sum_j \alpha_j) + A_{m,k} - \tau_m C_{m,k}$.
Set $\gamma_{m,k} = \exp(b_k - b_{\max})/\sum_j \exp(b_j - b_{\max})$, where $b_{\max} = \max_j b_j$.
Read $\mu_m^{\text{mem}} = \sum_k \gamma_{m,k}\mu_k^c$ and $v_m^{\text{mem}} = \sum_k \gamma_{m,k}[v_k^c + (\mu_k^c - \mu_m^{\text{mem}})^2]$.
Form $\lambda^{\text{loc}} = (v_m^c + \epsilon)^{-1}$, $\lambda^{\text{mem}} = (v_m^{\text{mem}} + \epsilon)^{-1}$.
Set $K_m = \lambda^{\text{mem}}/(\lambda^{\text{loc}} + \lambda^{\text{mem}})$ and $\delta_m = K_m \odot (\mu_m^{\text{mem}} - \mu_m^c)$.
for each causally eligible prediction $t \in \mathcal{T}_m$ **do**
    Add $D_\vartheta([q_t^{\text{mha}}; \delta_m; K_m])$ to the local attention output; continue the frozen model to its next-token distribution.
end for
for $u \in \{a, c\}$ and slot k **do**
    Set $\lambda_m^{u,\text{obs}} = (v_m^u + \epsilon)^{-1}$.
    Write $\lambda_{m,k}^u = \lambda_k^u + \gamma_{m,k}\lambda_m^{u,\text{obs}}$, $\eta_{m,k}^u = \eta_k^u + \gamma_{m,k}\lambda_m^{u,\text{obs}} \odot \mu_m^u$.
end for
Set $\alpha_{m,k} = (1 - \rho)\alpha_k + \rho\alpha_k^0 + \gamma_{m,k}$ for every slot; commit M_m once.

Ensure: Fixed-capacity state M_m and residuals computed from M^-

Causal schedule. The requirement in Algorithm 1 is met as follows. During prompt prefill, the prompt is fully observed before any output is produced, so each 256-token block is pooled once and

its residual serves the prompt positions of the block. During generation, each new token is processed as its own evidence cycle: the read for predicting x_{t+1} uses M_{t-1} and features of $x_{\le t}$, and the token is then written once. Every generated token therefore depends only on its prefix, and the probes in Section 4.4 score answers with the same schedule. Teacher-forced training applies block pooling to the whole sequence, so positions inside a block can see pooled features of later tokens of that block during training.

Complexity. The persistent state has $K_s(2D_a + 2D_c + 1)$ scalars per adapted layer, and routing, reading, and writing cost $O(K_s(D_a + D_c))$ per evidence cycle, independent of the history length. The number of cycles grows linearly with the input, and the local attention cost is that of the sliding window.

H NOTATION

We first summarize the notation used in the theoretical derivation. Tokens are indexed by t and memory cycles by m. A target block associated with cycle m is denoted by

$$X_m = x_{b_m:e_m-1} = (x_{b_m}, x_{b_m+1}, \dots, x_{e_m-1}), \tag{12}$$

where b_m and e_m are its start and exclusive end token indices. For token-level generation, the frozen sliding-window LLM observes only the local context

$$\ell_t = (x_i)_{i \in I_t}, \qquad I_t \subseteq \{1, \dots, t-1\}, \tag{13}$$

where I_t selects the accessible sink and sliding-window tokens. We write $\mathcal{C}_m$ for the observed context before the target block; within-block prefixes enter through the autoregressive factorization of the likelihood, not by conditioning on the entire target block. We reserve c_m for the content latent and E_m for the evidence supplied to the encoder and write. For every scored prediction $t \in \mathcal{T}_m$, all features of E_m must depend only on $x_{<t}$. Thus E_m and X_m need not be the same block: in our evaluation, prompt blocks are pooled after the whole prompt is observed and each generated token forms its own cycle (Section 3.4). Algorithm 1 makes the read and commit ordering explicit. We use σ^{u2} for diagonal variances below; it is the same quantity denoted by v^u in the main text, not a standard deviation.

BeliefMem maintains a finite memory state after each committed evidence block,

$$M_m = \{M_{m,k}\}_{k=1}^{K_s}, \tag{14}$$

where K_s is the number of memory slots and M_0 is a learned initial state whose content precisions start near zero. Each slot contains a factored address belief, a factored content belief, and a Dirichlet assignment count:

$$M_{m,k} = \left(\eta^a_{m,k}, \lambda^a_{m,k}, \eta^c_{m,k}, \lambda^c_{m,k}, \alpha_{m,k}\right). \tag{15}$$

The superscripts a and c refer to address and content latent variables. For a diagonal Gaussian, λ denotes precision and η denotes the precision-weighted mean:

$$\eta = \lambda \odot \mu, \qquad \mu = \eta \oslash \lambda, \qquad \sigma^2 = \lambda^{-1}, \tag{16}$$

where $\odot$ and $\oslash$ denote elementwise multiplication and division.

For chunk m, the latent variable $s_m \in \{1, \dots, K_s\}$ denotes the memory slot assignment, $a_m \in \mathbb{R}^{D_a}$ denotes the address latent, and $c_m \in \mathbb{R}^{D_c}$ denotes the content latent. The amortized encoder produces diagonal Gaussian posteriors

$$q_\phi(a_m \mid E_m) = \mathcal{N}\left(a_m; \mu^a_m, \operatorname{diag} \sigma^{a2}_m\right), \tag{17}$$

$$q_\phi(c_m \mid E_m) = \mathcal{N}\left(c_m; \mu^c_m, \operatorname{diag} \sigma^{c2}_m\right). \tag{18}$$

The categorical responsibility for assigning evidence cycle m to slot k is

$$\gamma_{m,k} \triangleq q(s_m = k), \qquad \sum_{k=1}^{K_s} \gamma_{m,k} = 1. \tag{19}$$

I GENERATIVE MODEL

BeliefMem starts from the following modeling question: given a frozen LLM that can only attend to a local window, how can the next-token distribution approximate the distribution conditioned on the full history? If a full-context LLM models $p_\theta(x_t \mid x_{<t})$, a finite-memory approximation must introduce a latent representation for the window-external evidence. We write this as

$$p^\star_\theta(x_t \mid x_{<t}) \approx \int p_{\theta,\vartheta}(x_t \mid \ell_t, z_t) p(z_t \mid M_{m(t)-1})\, dz_t, \tag{20}$$

where ℓ_t preserves local syntax and recent facts, while $m(t)$ denotes the memory cycle available to prediction t and $M_{m(t)-1}$ summarizes previously processed evidence. The backbone θ is frozen and ϑ denotes adapter parameters. The bounded-state approximation need not preserve all information in the exact history. In particular, the moment-based decoder does not evaluate this latent integral exactly.

For tractable online inference, BeliefMem uses chunk-level latent variables. The content latent c_m plays the role of the chunk-level long-range evidence used by the likelihood, while the address latent a_m explains where the evidence should be assigned in memory. Conditioned on the previous memory state M_{m-1}, the chunk-level generative model factorizes as

$$\begin{aligned} p_{\theta,\vartheta}(X_m, a_m, c_m, s_m, \pi_m \mid \mathcal{C}_m, M_{m-1}) = {} & p_{\theta,\vartheta}(X_m \mid \mathcal{C}_m, c_m) p(c_m \mid s_m, M_{m-1}) \\ & \cdot p(a_m \mid s_m, M_{m-1}) \pi_{m,s_m} \operatorname{Dir}(\pi_m; \alpha_{m-1}). \end{aligned} \tag{21}$$

The factors have direct probabilistic roles. First, $p_{\theta,\vartheta}(X_m \mid \mathcal{C}_m, c_m)$ is the autoregressive likelihood after memory content is injected into the frozen backbone:

$$p_{\theta,\vartheta}(X_m \mid \mathcal{C}_m, c_m) = \prod_{t=b_m}^{e_m-1} p_{\theta,\vartheta}(x_t \mid \ell_t(\mathcal{C}_m, x_{b_m:t-1}), c_m). \tag{22}$$

Second, $p(a_m \mid s_m, M_{m-1})$ measures address compatibility. Third, $p(c_m \mid s_m, M_{m-1})$ measures content compatibility. Finally, $p(s_m = k \mid \pi_m) = \pi_{m,k}$ and the Dirichlet distribution specifies uncertainty over these mixing weights. Keeping π_m explicit is essential for the expected-log-weight variational derivation below.

Integrating out π_m would give the categorical probability $\alpha_{m-1,k} / \sum_j \alpha_{m-1,j}$. The variational update instead uses the expected log weight under $q(\pi_m) = \operatorname{Dir}(\alpha_{m-1})$:

$$\mathbb{E}[\log \pi_{m,k}] = \psi(\alpha_{m-1,k}) - \psi\left(\sum_{j=1}^{K_s} \alpha_{m-1,j}\right), \tag{23}$$

where $\psi(\cdot)$ is the digamma function. The address and content conditional distributions are diagonal Gaussians defined by the corresponding slot beliefs:

$$p(a_m \mid s_m = k, M_{m-1}) = \mathcal{N}\big(a_m; \mu^a_{m-1,k}, \operatorname{diag} \sigma^{a2}_{m-1,k}\big), \tag{24}$$

$$p(c_m \mid s_m = k, M_{m-1}) = \mathcal{N}\big(c_m; \mu^c_{m-1,k}, \operatorname{diag} \sigma^{c2}_{m-1,k}\big). \tag{25}$$

The moment parameters in Eq. (25) are obtained from the natural parameters in Eq. (16).

J VARIATIONAL INFERENCE AND APPROXIMATE FILTERING

The marginal likelihood of chunk X_m under the generative model is obtained by summing over the discrete slot assignment and integrating over the continuous latents:

$$\begin{aligned} & \log p_{\theta,\vartheta}(X_m \mid \mathcal{C}_m, M_{m-1}) \\ & \quad = \log \sum_{s_m=1}^{K_s} \int\int\int p_{\theta,\vartheta}(X_m, a_m, c_m, s_m, \pi_m \mid \mathcal{C}_m, M_{m-1})\, da_m\, dc_m\, d\pi_m. \end{aligned} \tag{26}$$

Exact inference in Eq. (26) is intractable because the LLM likelihood is high-dimensional and the posterior over memory assignments must be updated online. We therefore introduce the mean-field variational family

$$q_m(a_m, c_m, s_m, \pi_m) = q_\phi(a_m \mid E_m) q_\phi(c_m \mid E_m) q(s_m) q(\pi_m), \tag{27}$$

where the address and content factors are amortized by the encoder and $q(s_m)$ is categorical. We hold $q(\pi_m) = \mathrm{Dir}(\alpha_{m-1})$ fixed in this routing derivation. Optimizing it would require an additional coordinate step. The evidence input E_m obeys the causal constraint above.

For compactness, let

$$p_m^{\theta,\vartheta}(a_m, c_m, s_m, \pi_m) \equiv p_{\theta,\vartheta}(X_m, a_m, c_m, s_m, \pi_m \mid \mathcal{C}_m, M_{m-1}). \tag{28}$$

We also write

$$p_{m-1}(a_m, c_m, s_m, \pi_m) \equiv p(a_m, c_m, s_m, \pi_m \mid M_{m-1}) \tag{29}$$

for the memory-induced prior over the chunk latents. Applying Jensen's inequality to Eq. (26) gives the chunk-level evidence lower bound:

$$\begin{aligned}
\log p_{\theta,\vartheta}(X_m \mid \mathcal{C}_m, M_{m-1}) &= \log \mathbb{E}_{q_m}\left[\frac{p_m^{\theta,\vartheta}(a_m, c_m, s_m, \pi_m)}{q_m(a_m, c_m, s_m, \pi_m)}\right] \\
&\geq \mathbb{E}_{q_m}\left[\log p_{\theta,\vartheta}(X_m \mid \mathcal{C}_m, c_m)\right] - D_{\mathrm{KL}}\big(q_m(a_m, c_m, s_m, \pi_m) \\
&\qquad \big\| p_{m-1}(a_m, c_m, s_m, \pi_m)\big) \\
&\triangleq \mathcal{L}_{\mathrm{ELBO}}(m).
\end{aligned} \tag{30}$$

Expanding the KL term using the factorization in Eq. (21) yields

$$\begin{aligned}
\mathcal{L}_{\mathrm{ELBO}}(m) = \; & \mathbb{E}_{q_\phi(c_m \mid E_m)}\left[\log p_{\theta,\vartheta}(X_m \mid \mathcal{C}_m, c_m)\right] \\
& + \mathbb{E}_{q(s_m)}\left[\mathbb{E}[\log \pi_{m,s_m}]\right] \\
& + \mathbb{E}_{q(s_m) q_\phi(a_m \mid E_m)}\left[\log p(a_m \mid s_m, M_{m-1})\right] \\
& + \mathbb{E}_{q(s_m) q_\phi(c_m \mid E_m)}\left[\log p(c_m \mid s_m, M_{m-1})\right] \\
& + \mathcal{H}[q(s_m)] + \mathcal{H}[q_\phi(a_m \mid E_m)] + \mathcal{H}[q_\phi(c_m \mid E_m)].
\end{aligned} \tag{31}$$

The π_m-KL is zero because its variational factor equals its prior. Equation (31) motivates the later formulas: optimizing the categorical factor gives an exact variational responsibility for this model, while the Gaussian prior terms give expected compatibility scores. The normalized overlap router in the implementation is an approximation to this coordinate update. The standard ELBO has unit KL weight; distillation and a free regularization coefficient define the separate training surrogate in Eq. (10).

After inferring the latent variables, the memory must be updated. Let S_m denote a hypothetical latent memory configuration. An exact dynamic Bayesian filter would have the prediction and correction steps

$$p(S_m \mid E_{1:m}) \propto p(E_m \mid S_m) \int p(S_m \mid S_{m-1}) p(S_{m-1} \mid E_{1:m-1})\, dS_{m-1}. \tag{32}$$

With uncertain slot assignments, the resulting posterior generally leaves a fixed factored-Gaussian family. For a chosen transition and likelihood, let $\widetilde{p}_m$ denote the normalized correction of an approximate previous posterior. A general assumed-density filtering projection would solve

$$q_m^{\mathrm{ADF}} = \arg\min_{q \in \mathcal{F}} \mathrm{KL}(\widetilde{p}_m \| q). \tag{33}$$

MAP estimation instead selects a mode, and is not equivalent to this distributional projection. BeliefMem does not solve the general projection above: it fixes fractional responsibilities and uses Gaussian sites whose natural parameters add analytically (Appendix N).

K Posterior Responsibility

We now derive the slot responsibility from the ELBO. Holding the continuous variational factors fixed, the coordinate-ascent update for the categorical factor is

$$\log q^\star(s_m = k) = \mathbb{E}_{q_\phi(a_m \mid E_m) q_\phi(c_m \mid E_m) q(\pi_m)}\left[\log p(a_m, c_m, s_m = k, \pi_m \mid M_{m-1})\right] + \text{const}, \tag{34}$$

where the constant does not depend on k. Using the factorization in Eq. (21), this becomes

$$\log q^\star(s_m = k) = \mathbb{E}[\log \pi_{m,k}] + \mathbb{E}_{q_\phi(a_m|E_m)}\left[\log p(a_m \mid s_m = k, M_{m-1})\right] + \mathbb{E}_{q_\phi(c_m|E_m)}\left[\log p(c_m \mid s_m = k, M_{m-1})\right] + \text{const.} \tag{35}$$

The address term follows from the expected log likelihood of a Gaussian under another Gaussian. For $q_\phi(a_m \mid E_m) = \mathcal{N}(\mu^a_m, \operatorname{diag}\sigma^{a2}_m)$ and slot belief $p(a_m \mid s_m = k, M_{m-1}) = \mathcal{N}(\mu^a_{m-1,k}, \operatorname{diag}\sigma^{a2}_{m-1,k})$,

$$\begin{aligned}&\mathbb{E}_{q_\phi(a_m|E_m)}\left[\log p(a_m \mid s_m = k, M_{m-1})\right] \\ &= -\frac{1}{2}\sum_{j=1}^{D_a}\left[\log(2\pi\sigma^{a2}_{m-1,k,j}) + \frac{\sigma^{a2}_{m,j} + (\mu^a_{m,j} - \mu^a_{m-1,k,j})^2}{\sigma^{a2}_{m-1,k,j}}\right].\end{aligned} \tag{36}$$

Dropping the slot-independent constant $-\frac{D_a}{2}\log(2\pi)$ leaves the softmax unchanged. Dividing only the address term by D_a, however, changes its strength relative to the prior. With this normalization and variance jitter, the practical address compatibility is

$$A_{m,k} = -\frac{1}{2}\operatorname{mean}_j\left[\log(\sigma^{a2}_{m-1,k,j} + \epsilon) + \frac{\sigma^{a2}_{m,j} + (\mu^a_{m,j} - \mu^a_{m-1,k,j})^2}{\sigma^{a2}_{m-1,k,j} + \epsilon}\right]. \tag{37}$$

The exact content term is another expected log density. The implementation instead uses an overlap energy. For independent draws from the local content posterior and one slot belief, their difference is distributed as

$$c_m - c_{m-1,k} \sim \mathcal{N}\left(\mu^c_m - \mu^c_{m-1,k}, \operatorname{diag}(\sigma^{c2}_m + \sigma^{c2}_{m-1,k})\right). \tag{38}$$

Evaluating this difference density at zero gives the overlap identity

$$\int q_\phi(c_m \mid E_m) p(c_m \mid s_m = k, M_{m-1})\, dc_m = \mathcal{N}\big(\mu^c_m; \mu^c_{m-1,k}, \operatorname{diag}(\sigma^{c2}_m + \sigma^{c2}_{m-1,k})\big). \tag{39}$$

Thus, after dimension normalization and jitter, the negative log overlap is

$$C_{m,k} = \frac{1}{2}\operatorname{mean}_j\left[\log(\sigma^{c2}_{m-1,k,j} + \sigma^{c2}_{m,j} + \epsilon) + \frac{(\mu^c_{m,j} - \mu^c_{m-1,k,j})^2}{\sigma^{c2}_{m-1,k,j} + \sigma^{c2}_{m,j} + \epsilon}\right]. \tag{40}$$

In the exact Gaussian likelihood expansion, the relative scale is fixed. The implemented overlap energy instead uses a learned nonnegative coefficient τ_m, distinct from slot and observation precision. Its input $u_m \in \mathbb{R}^8$ summarizes address/content dispersion and margins, slot-prior entropy, and chunk/memory uncertainty:

$$\tau_m = \operatorname{softplus}(g_\omega(u_m)). \tag{41}$$

This is an implementation approximation to the content likelihood temperature. Combining Eqs. (23), (37), and (40) with the learned overlap approximation, the implemented responsibility logit is

$$r_{m,k} = \mathbb{E}[\log \pi_{m,k}] + A_{m,k} - \tau_m C_{m,k}, \qquad \gamma_{m,k} = \frac{\exp(r_{m,k})}{\sum_{j=1}^{K_s}\exp(r_{m,j})}. \tag{42}$$

This γ_m is a variationally motivated approximate responsibility. The overlap uses summed variances, whereas the exact expected log density uses encoder variance in its numerator; these are distinct operations. Dimension normalization and learned scaling further alter the coordinate update. Sharing the resulting γ_m still makes reading and fractional writing refer to the same inferred evidence ownership.

L Posterior Predictive Reading

Online generation requires read-before-write: predictions using cycle m read M_{m-1} rather than the state that already includes its current write. The encoder evidence E_m must independently satisfy the causal constraint in Algorithm 1. Given γ_m, define the memory read as

$$p(c_m \mid \gamma_m, M_{m-1}) = \sum_{k=1}^{K_s}\gamma_{m,k}\mathcal{N}\left(c_m; \mu^c_{m-1,k}, \operatorname{diag}\sigma^{c2}_{m-1,k}\right). \tag{43}$$

The exact distribution in Eq. (43) is a Gaussian mixture. To pass a fixed-size belief to the emission module, BeliefMem matches its first two moments. The predictive mean is

$$\mu_m^{\text{mem}} = \mathbb{E}[c_m \mid \gamma_m, M_{m-1}] = \sum_{k=1}^{K_s} \gamma_{m,k} \mu_{m-1,k}^c. \tag{44}$$

The predictive second raw moment is

$$\mathbb{E}[c_m \odot c_m \mid \gamma_m, M_{m-1}] = \sum_{k=1}^{K_s} \gamma_{m,k} \left(\sigma_{m-1,k}^{c2} + \mu_{m-1,k}^c \odot \mu_{m-1,k}^c \right). \tag{45}$$

Therefore the diagonal predictive variance is

$$\begin{aligned} \sigma_m^{\text{mem2}} &= \mathbb{E}[c_m \odot c_m \mid \gamma_m, M_{m-1}] - \mu_m^{\text{mem}} \odot \mu_m^{\text{mem}} \\ &= \sum_{k=1}^{K_s} \gamma_{m,k} \left[\sigma_{m-1,k}^{c2} + (\mu_{m-1,k}^c - \mu_m^{\text{mem}}) \odot (\mu_{m-1,k}^c - \mu_m^{\text{mem}}) \right]. \end{aligned} \tag{46}$$

Equations (44)–(46) are exact moment identities for the mixture in Eq. (43); the approximation is the subsequent projection of the mixture to a single diagonal Gaussian.

For completeness, the full covariance before diagonal projection is

$$\text{Cov}(c_m) = \sum_k \gamma_{m,k} \operatorname{diag} \sigma_{m-1,k}^{c2} + \sum_k \gamma_{m,k} (\mu_{m-1,k}^c - \mu_m^{\text{mem}})(\mu_{m-1,k}^c - \mu_m^{\text{mem}})^\top. \tag{47}$$

The second term has off-diagonal entries even though every component is diagonal. The projection preserves the marginal variances but discards these cross terms and the mixture's component identities. The centered variance expression also avoids subtracting two large raw second moments numerically.

M Product-of-Experts Fusion

After moment projection and variance jitter, the local and memory experts are diagonal Gaussians over the same content latent c_m:

$$q_{\text{loc}}(c_m) = \mathcal{N}\left(c_m; \mu_m^c, \operatorname{diag}(\sigma_m^{c2} + \epsilon)\right), \tag{48}$$

$$q_{\text{mem}}(c_m) \approx \mathcal{N}\left(c_m; \mu_m^{\text{mem}}, \operatorname{diag}(\sigma_m^{\text{mem2}} + \epsilon)\right). \tag{49}$$

BeliefMem defines their normalized product as a fusion operator:

$$q_{\text{fus}}(c_m) \propto q_{\text{loc}}(c_m) q_{\text{mem}}(c_m). \tag{50}$$

For diagonal Gaussians, the product remains diagonal Gaussian because natural parameters add. Define

$$\lambda_m^{\text{loc}} = \frac{1}{\sigma_m^{c2} + \epsilon}, \qquad \eta_m^{\text{loc}} = \lambda_m^{\text{loc}} \odot \mu_m^c, \tag{51}$$

and

$$\lambda_m^{\text{mem}} = \frac{1}{\sigma_m^{\text{mem2}} + \epsilon}, \qquad \eta_m^{\text{mem}} = \lambda_m^{\text{mem}} \odot \mu_m^{\text{mem}}. \tag{52}$$

Then

$$\lambda_m^{\text{fus}} = \lambda_m^{\text{loc}} + \lambda_m^{\text{mem}}, \tag{53}$$

$$\eta_m^{\text{fus}} = \eta_m^{\text{loc}} + \eta_m^{\text{mem}}, \tag{54}$$

$$\sigma_m^{\text{fus2}} = \left(\lambda_m^{\text{fus}}\right)^{-1}, \tag{55}$$

$$\mu_m^{\text{fus}} = \frac{\lambda_m^{\text{loc}} \odot \mu_m^c + \lambda_m^{\text{mem}} \odot \mu_m^{\text{mem}}}{\lambda_m^{\text{loc}} + \lambda_m^{\text{mem}}}. \tag{56}$$

It is useful to rewrite Eq. (56) as a correction to the local posterior. Define the memory precision gain

$$K_m = \frac{\lambda_m^{\text{mem}}}{\lambda_m^{\text{loc}} + \lambda_m^{\text{mem}}}, \tag{57}$$

where the division is elementwise. Then

$$\mu_m^{\mathrm{fus}} = \mu_m^c + K_m \odot (\mu_m^{\mathrm{mem}} - \mu_m^c), \tag{58}$$

$$\delta_m \triangleq \mu_m^{\mathrm{fus}} - \mu_m^c = K_m \odot (\mu_m^{\mathrm{mem}} - \mu_m^c). \tag{59}$$

Thus K_m is a dimensionwise reliance gain, with $0 < K_{m,j} < 1$ for finite positive precisions, and δ_m is the correction passed to the emission module. Higher memory variance lowers this gain at fixed local variance. For each causally eligible token, its query conditions the residual decoder:

$$\Delta O_t^{\mathrm{mem}} = D_\vartheta \left([q_t^{\mathrm{mha}}; \delta_m; K_m] \right), \qquad t \in \mathcal{T}_m. \tag{60}$$

The decoder's zero-initialized output projection preserves the local route at initialization. Equation (60) is a moment-based realization of memory-conditioned prediction, not an exact integration of a nonlinear language-model likelihood over latent content.

The product algebra does not establish that these experts are independent evidence posteriors. If q_1 and q_2 were posteriors formed from a shared prior p_0 and conditionally independent observations, Bayes' rule would give

$$p(c \mid y_1, y_2) \propto \frac{q_1(c) q_2(c)}{p_0(c)}. \tag{61}$$

Here routing also depends on the local evidence. We therefore use PoE as the specified fusion operator and do not identify K_m with a calibrated probability of answer correctness.

N Fractional-Site Memory Update

After reading and emitting, BeliefMem writes the observed chunk evidence into memory using fractional Gaussian observation sites. This is the analytic factored update motivated by bounded filtering; it is not an exact solution to the unrestricted projection in Eq. (33).

Let z_k^u be the slot prototype for $u \in \{a, c\}$. For a fixed assignment weight, interpret the encoder output as the fractional Gaussian site

$$t_{m,k}^u(z_k^u) = \mathcal{N}\big(\mu_m^u; z_k^u, \mathrm{diag}(\sigma_m^{u2} + \epsilon)\big)^{\gamma_{m,k}}. \tag{62}$$

The encoder's posterior variance is used as observation-site uncertainty. This is a specified approximation, not an independently observed ground-truth likelihood or a marginalization over the unknown assignment.

For the address posterior, define the observation precision and natural mean parameter

$$\lambda_m^{a,\mathrm{obs}} = \frac{1}{\sigma_m^{a2} + \epsilon}, \qquad \eta_m^{a,\mathrm{obs}} = \lambda_m^{a,\mathrm{obs}} \odot \mu_m^a. \tag{63}$$

For the content posterior,

$$\lambda_m^{c,\mathrm{obs}} = \frac{1}{\sigma_m^{c2} + \epsilon}, \qquad \eta_m^{c,\mathrm{obs}} = \lambda_m^{c,\mathrm{obs}} \odot \mu_m^c. \tag{64}$$

The reason these quantities add is the canonical form of a diagonal Gaussian:

$$\log \mathcal{N}(z; \mu, \mathrm{diag}\, \lambda^{-1}) = \eta^\top z - \frac{1}{2} z^\top \mathrm{diag}(\lambda) z - A(\eta, \lambda), \qquad \eta = \lambda \odot \mu. \tag{65}$$

Here the log-normalizer is

$$A(\eta, \lambda) = \frac{1}{2} \left[D \log(2\pi) - \sum_j \log \lambda_j + \sum_j \frac{\eta_j^2}{\lambda_j} \right]. \tag{66}$$

The canonical linear and quadratic coefficients are η and $-\frac{1}{2}\lambda$. Multiplying by the site in Eq. (62) therefore adds the responsibility-weighted natural parameters, giving

$$\lambda_{m,k}^a = \lambda_{m-1,k}^a + \gamma_{m,k} \lambda_m^{a,\mathrm{obs}}, \qquad \eta_{m,k}^a = \eta_{m-1,k}^a + \gamma_{m,k} \eta_m^{a,\mathrm{obs}}, \tag{67}$$

$$\lambda_{m,k}^c = \lambda_{m-1,k}^c + \gamma_{m,k} \lambda_m^{c,\mathrm{obs}}, \qquad \eta_{m,k}^c = \eta_{m-1,k}^c + \gamma_{m,k} \eta_m^{c,\mathrm{obs}}. \tag{68}$$

The updated moment parameters are recovered by Eq. (16). For example,

$$\mu^c_{m,k} = \frac{\eta^c_{m-1,k} + \gamma_{m,k}\eta^{c,\mathrm{obs}}_m}{\lambda^c_{m-1,k} + \gamma_{m,k}\lambda^{c,\mathrm{obs}}_m}, \qquad \sigma^{c2}_{m,k} = \left(\lambda^c_{m-1,k} + \gamma_{m,k}\lambda^{c,\mathrm{obs}}_m\right)^{-1}. \tag{69}$$

Equation (69) shows explicitly that a high-confidence observation has a large precision and therefore moves the slot mean more strongly, while an uncertain observation contributes weakly.

Equivalently, the mean update separates assignment from overwrite strength:

$$\begin{aligned} w^u_{m,k} &= \frac{\gamma_{m,k}\lambda^{u,\mathrm{obs}}_m}{\lambda^u_{m-1,k} + \gamma_{m,k}\lambda^{u,\mathrm{obs}}_m}, \\ \mu^u_{m,k} &= \mu^u_{m-1,k} + w^u_{m,k} \odot (\mu^u_m - \mu^u_{m-1,k}). \end{aligned} \tag{70}$$

At fixed incoming evidence, a larger stored precision decreases $w^u_{m,k}$. These identities are exact for the selected fractional sites. Marginalizing an uncertain assignment would generally introduce additional mixture terms.

The slot prior uses a fractional categorical site $\prod_k \pi_k^{\gamma_{m,k}}$. Multiplying this site by a Dirichlet density adds its exponents. Without refresh, this gives

$$\alpha_{m,k} = \alpha_{m-1,k} + \gamma_{m,k}. \tag{71}$$

BeliefMem uses a lightweight dynamic prior to prevent early chunks from permanently dominating the slot posterior:

$$\tilde{\alpha}_{m-1,k} = (1-\rho)\alpha_{m-1,k} + \rho\alpha^0_k, \qquad \alpha_{m,k} = \tilde{\alpha}_{m-1,k} + \gamma_{m,k}. \tag{72}$$

Here $\alpha^0 > 0$ is the base Dirichlet parameter and $0 \le \rho \le 1$. The refresh can be written as a normalized geometric blend:

$$\widetilde{p}(\pi) \propto \mathrm{Dir}(\pi; \alpha_{m-1})^{1-\rho}\,\mathrm{Dir}(\pi; \alpha^0)^{\rho} \propto \mathrm{Dir}(\pi; (1-\rho)\alpha_{m-1} + \rho\alpha^0). \tag{73}$$

It is a dynamic modeling choice, not exact inference under an unchanged static prior. The router uses the previous counts before this write. Count refresh reduces assignment-prior lock-in but does not decay Gaussian precision, which can continue accumulating over repeated updates.

O Dirichlet Counts and Numerical Jitter

For assignment weights with prior $\mathrm{Dir}(\boldsymbol{\pi}; \boldsymbol{\alpha}^0)$, multiplying fixed fractional categorical sites with $n_k = \sum_m \gamma_{m,k}$ (without refresh) gives the posterior $\mathrm{Dir}(\boldsymbol{\pi}; \alpha^0_1 + n_1, \ldots, \alpha^0_{K_s} + n_{K_s})$, whose expected log weight $\psi(\alpha^0_k + n_k) - \psi(\sum_j(\alpha^0_j + n_j))$ is the first term of the responsibility logit (Eq. (23)). Fractional counts have a power-likelihood interpretation rather than counting integer observations, and the refresh of Eq. (72) replaces $\alpha^0 + n$ by its geometrically blended version.

Variances enter every precision as $\sigma^2 + \epsilon$, so each incoming site has precision at most $1/\epsilon$. This jitter is a numerical safeguard, not a prior; it does not bound the accumulated slot precision, which grows with the number of writes.

P Approximation Boundaries

We close by separating the exact probabilistic derivations from implementation approximations.

Exact under the stated model. The ELBO in Eq. (30) follows directly from Jensen's inequality. The coordinate update in Eq. (34) is the standard variational update for a categorical latent variable. The address expected log likelihood in Eq. (36) is the exact expectation of a Gaussian log likelihood under a Gaussian posterior. The predictive mean and variance in Eqs. (44)–(46) are exact first and second moments of the responsibility-weighted mixture. The Product-of-Experts equations in Eqs. (50)–(59) are exact for diagonal Gaussian experts. The natural-parameter addition in Eqs. (67)–(68) follows from exponential-family conjugacy for Gaussian factors.

Implementation approximations. BeliefMem uses amortized evidence posteriors rather than exact token-level posteriors. The factorization in Eq. (27) is a mean-field approximation. Diagonal slot beliefs and moment projection discard cross-dimensional covariance and mixture structure. Fixed capacity and fractional sites replace exact posterior growth. The normalized content overlap and learned τ_m are design choices, not the exact categorical coordinate update. Top-k distillation substitutes for the likelihood objective, and the freely weighted regularizer is not the same ELBO. Dirichlet refresh changes the assignment prior but does not guarantee indefinite Gaussian-state plasticity. These boundaries do not invalidate the analytic identities; they specify how those identities are used in the adapter.

Training regularization. The memory terms of Eq. (10) are closed-form surrogates of the ELBO's prior terms. For content, Eq. (31) contains $\sum_k \gamma_{m,k}\, \mathbb{E}_q[\log p(c_m \mid s_m{=}k, M_{m-1})] + \mathcal{H}[q_\phi(c_m \mid E_m)] = -\sum_k \gamma_{m,k} \mathrm{KL}(q_\phi \,\|\, p_k)$, which by joint convexity upper-bounds the KL to the mixture $\sum_k \gamma_{m,k} p_k$. The implementation instead uses the KL to the moment-matched read $p_{\mathrm{mem}}(c) = \mathcal{N}(\mu_m^{\mathrm{mem}}, \operatorname{diag} v_m^{\mathrm{mem}})$, which has the closed form

$$\mathrm{KL}(\mathcal{N}(\mu, \operatorname{diag} v) \,\|\, \mathcal{N}(\mu_0, \operatorname{diag} v_0)) = \frac{1}{2} \sum_j \left[\log \frac{v_{0,j}}{v_j} + \frac{v_j + (\mu_j - \mu_{0,j})^2}{v_{0,j}} - 1 \right]. \tag{74}$$

For the address, the reference is the moment-matched prior predictive mixture $\sum_k \mathbb{E}[\pi_k] \mathcal{N}(\mu^a_{m-1,k}, \operatorname{diag} v^a_{m-1,k})$, i.e., the memory's prediction before routing. Both terms measure how surprising the new evidence is under the current memory. Together with top-k distillation in place of the likelihood and the weight $\beta \neq 1$, they make Eq. (10) a training surrogate rather than the ELBO.

Q LIMITATIONS

Fixed slots and diagonal beliefs compress history: verbatim recall of an arbitrary string (e.g., a passkey) beyond the window is far weaker than with full attention, and slot precision is not a calibrated answer probability. Gains depend on task and backbone (e.g., 3B code repositories, 7B history scaling); controls use three seeds, and external checkpoints differ in training history.