Resilient Latent Readouts for Long-Context Question Answering

Jingyi Liao[1,2] **Rongcheng Tu**[1] **Wenhao Sun**[1] **Yiting Li**[2] **Zhao Jin**[1]

Khin Mi Mi Aung[2] **Zhuoyi Lin**[2] **Xun Xu**[2,*] **Dacheng Tao**[1,*]

[1] Nanyang Technological University
[2]Institute of Advanced Intelligence and Computing, A*STAR
jingyi012@e.ntu.edu.sg xu_xun@a-star.edu.sg dacheng.tao@ntu.edu.sg

Abstract

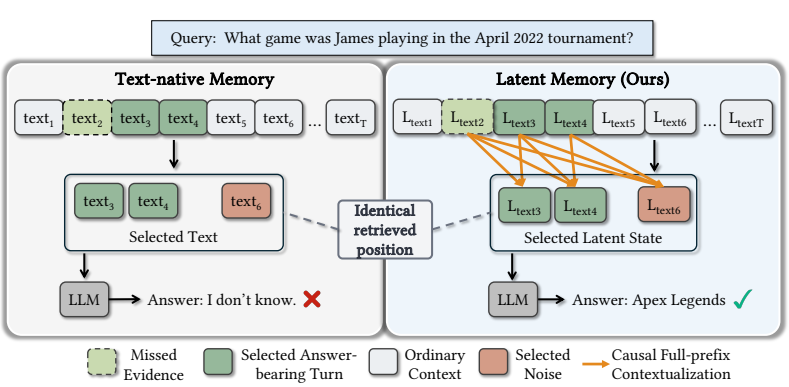
Long-context question answering often asks a model to answer from extended documents, multi-turn dialogues, or persistent histories. Directly prefilling the full history preserves all observed tokens, but it is expensive and can expose generation to substantial irrelevant or spuriously salient context, which may hinder evidence localization and answer synthesis. This motivates compact memory interfaces that provide a short, query-conditioned readout of the history. Existing systems predominantly instantiate this readout as text, where the reader sees only what was selected and any evidence left out is unrecoverable, leaving the interface fragile to selection errors. We argue that memory readout has two coupled axes that should be optimized separately: *evidential coverage*, whether all right support is selected, and *inferential resilience*, how much answerability remains under imperfect selection. To jointly address these two axes, we propose LIRA (**L**atent **I**nformation **R**eadout and **A**ssembly), a latent-memory reader that exposes full-prefix contextualized states rather than re-encoded text. It amortizes one full-prefix pass into a reusable latent store; for each query, it localizes candidate positions with calibrated endogenous attention, repairs incomplete support with coverage-aware planning, and packs selected states into a position-consistent readout for autoregressive generation. Across three open-weight backbones on LoCoMo and Loong, LIRA is the strongest non-oracle compact-memory method and surpasses Full Context on 8 of 9 LoCoMo metrics. When localization misses all supporting evidence, LIRA improves F1 by up to 13.3 points over a text-native control with identical positions, suggesting that full-prefix states preserve answer-relevant influence discarded by text-native re-encoding.

1 Introduction

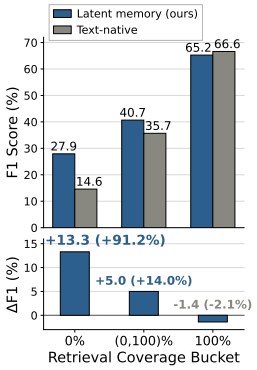
Long-context question answering requires a language model to identify and integrate evidence from extended documents, multi-turn dialogues, and persistent user histories [2, 11, 19]. A natural high-cost reference is to run a full-prefix pass over the entire history before answering. This preserves all observed content, but it also leaves the reader with a large active context, making inference expensive and forcing generation to compete with substantial irrelevant or spuriously salient information. [17]. This motivates an *oracle compact memory*: a short, query-conditioned readout that retains all answer-supporting evidence while discarding irrelevant history.

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Latent-state memory under identical localization error.



(b) F1 score (%) by retrieval coverage.

Figure 1: **(a)** Given identical selected positions that omit an earlier answer-bearing segment, text-native memory re-encodes the selected text under a truncated prefix, losing both the omitted text and its contextual influence on retained tokens. Latent-state memory reads the corresponding states from the original full-prefix prefill, where that influence is preserved. **(b)** Stratified by oracle-coverage of the retrieval output, latent-state reuse and text-native re-prefill behave near-identically when coverage is full, but the gap widens monotonically as coverage drops, reaching $+13.3$ F1 at zero coverage.

A common text-native instantiation returns ordinary text, passages, notes, summaries, or retrieved records, that is serialized into a short prompt, after which the reader performs a new prefill pass over this shortened context [15, 21, 39, 4]. Under this interface, the oracle target becomes primarily a textual coverage requirement: the compact prompt must explicitly express the evidence needed for the answer. This design is simple and model-agnostic, but brittle: facts not verbalized in the compact prompt cannot be reliably recovered from the memory readout.

Existing retrieval and long-term memory systems primarily respond to this coverage problem by improving localization [23]. Embedding-based retrievers learn semantic representations for passage selection [14, 36], while structured retrievers organize long histories into hierarchical or graph-based memories [9, 25]. Although they differ in how memories are indexed, summarized, and retrieved, they share a common objective: increasing the probability that the short memory output explicitly contains the answer-supporting evidence. Sharper localization is useful and necessary for approaching the oracle, but they leave aside the other side of the coin: whether a readout can remain answerable when it omits some or all of the answer-supporting evidence. We therefore distinguish two coupled axes of memory readout: *evidential coverage*, which measures whether the selected readout contains the answer-supporting evidence, and *inferential resilience*, which measures how much answerability remains when that coverage is imperfect.

This *inferential resilience* goal shifts attention from what is selected to what can the selected representation preserves. Under a non-oracle selection, answer-supporting spans are omitted as text, but those spans may still have shaped the full-prefix states of retained positions. In a causal Transformer, later-layer states are formed through repeated attention to each position's causal prefix, so a retained state can carry representational influence from earlier turns or document spans that are not themselves selected [29]. Text-native readout loses this channel when it serializes the selected tokens and re-encodes them under a truncated prefix. This motivates a *latent memory interface*, where a memory read returns executable, full-prefix contextualized model states rather than text that must be re-encoded [7, 18]; Figure 1a illustrates this representational difference. We validate this mechanism through a coverage-stratified controlled comparison: latent and text-native readers receive the same selected positions, so differences isolate memory representation rather than retrieval quality. Figure 1b shows that latent memory preserves more answerability as oracle-support coverage degrades, consistent with retained states carrying information from the full prefix.

With this latent-memory readout, improving *evidential coverage* becomes selecting the right positions to expose as latent states. A simple strategy is to retain positions with the highest attention scores [35], but raw attention over the full prefix is not purely evidential: it also reflects query-independent attractors such as attention sinks, role markers, delimiters, and recent turns. The readout should be calibrated to select states by query-conditioned support rather than by full-prefix salience alone.

In this work, we introduce LIRA (**L**atent **I**nformation **R**eadout and **A**ssembly), a latent-memory reader for long-context question answering. LIRA begins with a full-prefix pass over the history and materializing the resulting latent states as reusable memory. For each query, LIRA constructs a compact latent readout under a fixed budget: (i) It first uses calibrated endogenous attention to identify history positions that provide query-specific answer support. (ii) Because these initial selections may still miss part of the evidence, a coverage-aware planner adds nearby bridge context and sparse long-range residual positions when needed. (iii) We then assembles the selected full-prefix latent states into a position-consistent readout through paired through RoPE re-indexing. The resulting readout exposes answer-relevant positions while preserving the full-history contextualization carried by their latent states.

In summary, we (i) identify *inferential resilience* as a complementary objective to evidential coverage for memory reading, and motivate a latent-memory interface that preserves full-prefix contextualization rather than re-encoded text; (ii) propose LIRA, which combines calibrated endogenous localization, coverage-aware readout completion, and position-consistent latent-state assembly; and (iii) show empirically that LIRA is the strongest non-oracle compact-memory method on long-term conversational and multi-document QA across three open-weight backbones, with up to $+13.3$ F1 over a text-native control under identical positions when localization misses entirely.

2 Related Work

Memory-based methods for long-context QA. Existing retrieval and memory systems compress long histories into a query-conditioned readout for the reader. Retrieval-augmented generation inserts retrieved passages into the prompt [15, 14, 37]; structured variants build higher-level textual memories such as hierarchical summaries [25] or graph/community memories [5, 10]; prompt-compression methods rewrite tokens before inference [12]; and long-term agent-memory systems maintain persistent stores of textual records or summaries [21, 4, 33, 6]. All share a text-native readout: selected content is re-encoded by the reader as ordinary context. MemArt [35] is the closest latent-state precedent, storing dialogue turns as reusable latent state retrieved by compressed-key scoring; its block-level reuse, however, does not by itself address query-conditioned evidence selection or coverage under incomplete support. We study a complementary interface in which the readout consists of reusable latent states with explicit coverage planning.

KV-cache compression for full-cache fidelity. A separate line of work in efficient LLM inference reduces runtime memory footprint and decoding cost through KV-cache compression [16, 3] and selective state reuse [26]. Methods such as H2O, SnapKV, PyramidKV, Quest, and KVZip retain, evict, sparsify, or reconstruct cached states under a limited cache budget [38, 16, 3, 27, 13]. Their objective is to approximate full-cache inference behavior during a single forward pass. In contrast, we treat latent states as a query-conditioned memory interface whose objective is maximizing answerability under a compact readout budget, rather than approximating full-cache behavior.

3 Method

Figure 2 gives an overview of LIRA. Given a long-context history, LIRA first runs a full-prefix pass and materialized the resulting latent states as reusable memory. For each query, it constructs a compact latent readout in three stages. First, LIRA performs calibrated endogenous localization: it uses the reader's own question-conditioned attention to score history positions, and subtracts a neutral-probe baseline to remove query-independent attractors (§3.1). Second, it converts the localized anchors into a budgeted readout plan with coverage-aware completion, adding nearby bridge context or sparse long-range residual positions when the initial anchors appear insufficient (§3.2). Third, it assembles the selected full-prefix states into a position-consistent readout through RoPE re-indexing (§3.3). The reader then answers from this compact latent readout, where the long-context history is represented by selected states formed under the query-related full-prefix contextualized states rather than by text re-encoded under a truncated prefix.

Amortized latent-memory setup. Let $x_{1:P}$ denote the fixed prefix before the current question q. The first n_{sys} positions hold the system prompt and chat-template tokens and are always retained, leaving $\Omega = \{n_{\text{sys}} + 1, \ldots, P\}$ as the selectable history positions, over which we apply a per-query readout budget B. We segment the history into an ordered sequence $H = (\sigma_1, \ldots, \sigma_T)$ of contiguous

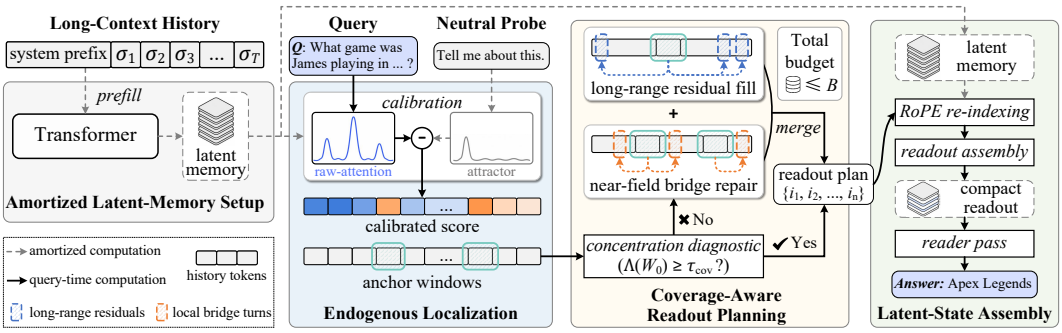


Figure 2: Overview of LIRA. From left to right, the figure traces through four stages: amortized latent-memory setup, calibrated endogenous localization, coverage-aware readout planning, and position-consistent latent-state assembly.

spans, and write $I_t \subseteq \Omega$ for the token positions of segment σ_t; segments are dialogue turns for conversational histories and newline-delimited text blocks otherwise. This simple segmentation provides coherent units for downstream anchor selection. A one-time full-prefix pass over $x_{1:P}$ materializes the latent memory store $\mathcal{M}(H) = \{(K_\ell, V_\ell)\}_{\ell=1}^{L}$, where K_ℓ and V_ℓ are the full-prefix key and value states at layer ℓ. These resulting full-prefix states form a reusable latent memory for later questions over the same history.

3.1 Calibrated Endogenous Localization

The first stage assigns a question-conditioned localization score to each selectable history position. We process the question against the amortized latent store and use the reader's attention from question tokens to stored prefix positions as an endogenous signal for candidate answer support. Drawing on the widely held observation that early-layer attention maps are often dominated by low-level syntactic and formatting patterns [28, 24], we restrict scoring to the final K_{loc} layers to reduce localization overhead and isolate signals more likely to exhibit retrieval-oriented attention [31].

Let q be tokenized as $y_{1:m}$, for a late layer ℓ and head h, let $A^{(\ell,h)}(q) \in \mathbb{R}_+^{m\times P}$ denote the attention block from question tokens to prefix positions, where $A_{j,i}^{(\ell,h)}(q)$ is the attention weight from token y_j to prefix position i. We assign each history position its strongest question-conditioned attention:

$$a_i(q) = \max_{\ell,h,j} A_{j,i}^{(\ell,h)}(q; \mathcal{M}(H)), \qquad i \in \Omega, \tag{1}$$

where the maximum ranges over all heads, all question-token positions $1 \leq j \leq m$, and layers $\ell = L - K_{\text{loc}} + 1, \ldots, L$. We apply max-pooling within each segment to ensure the recalled position remains sensitive to sparse, high-magnitude signals.

Attractor calibration. Long histories contain positions that attract attention regardless of the user question, such as BOS tokens, role markers, delimiters, and recent turns [32]. To separate these attractors from question-specific evidence, we use a fixed, task-agnostic neutral probe $\tilde{q}$.

For the baseline pass, $\tilde{q}$ attends to the same latent memory store $\mathcal{M}(H)$ used by the actual question. We then max-pool the probe-to-history attention over probe tokens, heads, and the final K_{loc} layers, yielding a query-independent salience baseline for each history position. Because $\tilde{q}$ contains no task-specific content, its attention captures history-specific salience rather than query-conditioned evidence; the exact probe string is detailed in Appendix.

Let $\widetilde{A}^{(\ell,h)}(\tilde{q}; \mathcal{M}(H)) \in \mathbb{R}_+^{\tilde{m}\times P}$ denote the attention block from neutral-probe tokens to cached positions. The attractor baseline is

$$b_i(H) = \max_{\ell,h,\tilde{j}} \widetilde{A}_{\tilde{j},i}^{(\ell,h)}(\tilde{q}; \mathcal{M}(H)), \qquad i \in \Omega, \tag{2}$$

where the maximum ranges over all heads, all neutral-probe token positions $1 \leq \tilde{j} \leq \tilde{m}$, and layers $\ell = L - K_{\text{loc}} + 1, \ldots, L$. The calibrated localization score is

$$s_i(q; H) = \big[a_i(q; H) - b_i(H)\big]_+, \qquad i \in \Omega. \tag{3}$$

After calibration, a position receives a positive score only when its question-conditioned attention exceeds its history-specific baseline. Figure 3 demonstrates that s_i achieves higher gold-token recall than raw attention a_i across retained-token budgets. The ablation study in §4.4 and Appendix C empirically validates the contribution of our attractor calibration.

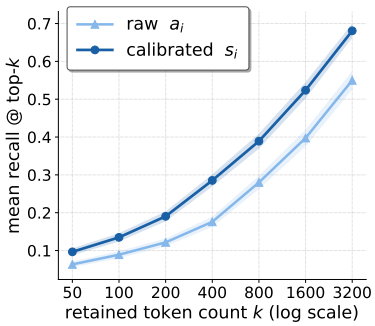


Figure 3: **Token-level localization recall.** History tokens are ranked by raw score a_i or calibrated score s_i. Recall measures the fraction of gold evidence tokens covered by the top-k positions, averaged over examples.

Segment-level anchoring. Token-level scores can be noisy at the level of individual subwords, while long-context QA often benefits from selecting coherent local context units. We therefore aggregate $s_i(q)$ over each segment σ_t:

$$u_t = \lambda \max_{i \in I_t} s_i(q) + (1-\lambda) \frac{1}{|I_t|} \sum_{i \in I_t} s_i(q), \qquad 0 \le \lambda \le 1. \tag{4}$$

The max term preserves sharp evidence peaks, while the mean term favors segments with sustained relevance rather than an isolated peak. We select top N_{anchor} segments by u_t and merge segments separated by fewer than δ intervening segments. This yields an initial set of anchor windows $\hat{\mathcal{W}}$, each represented as a token interval in Ω, and its covered token set $W_0 = \bigcup_{w \in \hat{\mathcal{W}}}$ and $w \subseteq \Omega$.

3.2 Coverage-Aware Readout Planning

The initial anchor set W_0 is often sufficient for questions whose evidence is concentrated in one or a few high-scoring segments. However, not all long-context questions are single-site localization problems. Some require near-field context, such as antecedents and clarifications around an anchor; others require isolated distant facts, or diffuse evidence scattered across the history, as in summary-like or trace-style queries. We therefore use a coverage diagnostic to decide whether the anchor plan should be expanded before assembling the compact latent readout.

Evidence-concentration diagnostic. For any token set $W \subseteq \Omega$, let $\mu_q(W) = \sum_{i \in W} s_i(q)$ denote the calibrated localization mass assigned to W. We measure whether the anchor set concentrates question-specific signal by the normalized density ratio

$$\Lambda_q(W_0) = \frac{\mu_q(W_0) + \epsilon}{\mu_q(\Omega) + \epsilon} \cdot \frac{|\Omega|}{|W_0|}. \tag{5}$$

Values near 1 indicate that the anchors are no more concentrated than the history background, while larger values indicate that W_0 concentrates question-conditioned signal. This diagnostic uses only the reader's localization scores and the size of the selected region; it does not use gold support annotations or answer text. If $\Lambda(W_0) \ge \tau_{\text{cov}}$, we treat the anchors as sufficiently concentrated and use W_0 directly. Otherwise, the planner triggers two targeted completion steps: near-field bridge repair adds local discourse context around the anchors, and long-range residual fill adds sparse high-scoring positions for distant missed evidence.

Near-field bridge repair. The first completion mechanism restores local context around the anchors. Uniformly widening every selected window would spend budget on many irrelevant tokens, so we score only unselected segments and admit a small number of nearby bridge segments. Because bridge segments are intended to provide discourse continuity rather than a single sharp answer span, we use a lower peak weight $\alpha < \lambda$:

$$u_t^{\text{br}} = \alpha \max_{i \in I_t} s_i(q) + (1-\alpha) \frac{1}{|I_t|} \sum_{i \in I_t} s_i(q), \qquad I_t \cap W_0 = \varnothing. \tag{6}$$

We combine this relevance with proximity to the nearest selected window,

$$d_t = \min_{w \in \hat{\mathcal{W}}} \text{dist}(I_t, w), \qquad \pi_t = \left[1 - \frac{d_t}{r_{\text{br}}} \right]_+, \tag{7}$$

where $\mathrm{dist}(I_t, w)$ is the token distance between intervals and r_{br} is the bridge radius. The bridge score is $g_t = u_t^{\mathrm{br}}\,\pi_t$. We rank nearby unselected segments by g_t admit the top N_{br} bridge segments $\mathcal{B}_{\mathrm{br}}$. The repaired local support is $W_1 = W_0 \cup \bigcup_{t\in\mathcal{B}_{\mathrm{br}}} I_t$.

Long-range residual fill. While bridge repair is local by design, its proximity factor can miss evidence that is far from all selected segments. We therefore add a sparse token-level fill for distant evidence. This step is natural for latent memory since an isolated retained state remains full-prefix contextualized, whereas serializing the same sparse tokens as text would be fragmented and hard to interpret. $B_{\mathrm{extra}} = \max\{0, B - |W_1|\}$. We select the highest-scoring uncovered positions:

$$E = \mathrm{TopK}_{B_{\mathrm{extra}}}\big(\Omega \setminus W_1;\, s_i(q)\big). \tag{8}$$

The final readout plan is

$$S(H,q) = \begin{cases} W_0, & \Lambda_q(W_0) \geq \tau_{\mathrm{cov}}, \\ W_1 \cup E, & \Lambda_q(W_0) < \tau_{\mathrm{cov}}, \end{cases} \tag{9}$$

If the anchor set fails the concentration test, the planner augments it with N_{br} bridge segments and then fills the remaining token budget with the highest-scoring uncovered positions; otherwise it returns W_0 unchanged.

3.3 Position-Consistent Latent-State Assembly

Section 3.2 returns a readout plan $S(H,q) = \{i_1 < \cdots < i_n\} \subseteq \Omega$ of absolute history positions. To maintain the compact positional encoding of memory segments from sparse positions, LIRA packs the selected full-prefix states after the always-retained system prefix, forming a compact latent readout.

For each layer ℓ, retained states are sourced as paired key–value cells $(\bar{K}_\ell[k], \bar{V}_\ell[k]) = (K_\ell[i_k], V_\ell[i_k])$ for $\ell = 1,\ldots,L$ and $k = 1,\ldots,n$, so attention reads a value from the same full-prefix position as its matched key. Since the token originally at i_k is assigned compact position $\rho(i_k) = n_{\mathrm{sys}} + k$, its key must also be re-rotated from the RoPE phase of i_k to that of $\rho(i_k)$, as illustrated in Figure 4.

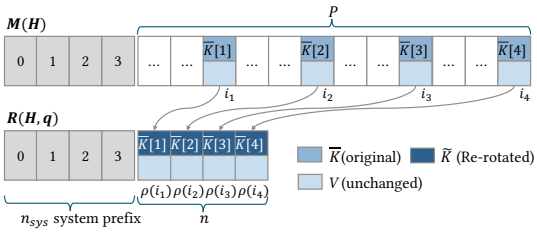


Figure 4: **Position-consistent latent-state assembly.** Gathered keys are RoPE re-indexed from original positions i_k to compact positions $\rho(i_k)$.

RoPE re-indexing. Let $\rho(i_k) = n_{\mathrm{sys}} + k$ denote the compact position assigned to retained history position i_k. When the latent store contains post-RoPE keys, the gathered key $\bar{K}_\ell[k]$ is still encoded with the RoPE phase of its original position i_k. We therefore re-index it to the compact coordinate:

$$\tilde{K}_\ell[k] = R_{\rho(i_k)} R_{i_k}^{-1} \bar{K}_\ell[k] = R_{\rho(i_k)-i_k} \bar{K}_\ell[k], \tag{10}$$

where R_p denotes the RoPE rotation at position p, applied headwise over RoPE channels. If the inference stack exposes pre-RoPE keys, this reduces to applying $R_{\rho(i_k)}$ to the raw key. Values are not RoPE-rotated and are copied unchanged as $\tilde{V}_\ell[k] = \bar{V}_\ell[k]$, since under RoPE position is encoded entirely through query–key rotation; the value stream carries no positional phase and is invariant under the remapping ρ.

Readout assembly and reader pass. Let $n = |S(H,q)|$ be the total number of retained history positions, and let $q = (y_1,\ldots,y_m)$ be the tokenized question. With $K_\ell^{\mathrm{sys}} = K_\ell^H[1 : n_{\mathrm{sys}}]$ and $V_\ell^{\mathrm{sys}} = V_\ell^H[1 : n_{\mathrm{sys}}]$, the final layer-$\ell$ compact latent readout is

$$\mathcal{R}_\ell(H,q) = \Big([K_\ell^{\mathrm{sys}}; \tilde{K}_\ell], [V_\ell^{\mathrm{sys}}; \tilde{V}_\ell]\Big), \qquad \mathcal{R}(H,q) = \{\mathcal{R}_\ell(H,q)\}_{\ell=1}^{L}. \tag{11}$$

This readout has compact prefix length $n_{\mathrm{sys}} + n$, consisting of the always-retained system states followed by the selected history states. The question tokens are placed after this compact readout at positions $n_{\mathrm{sys}} + n + 1,\ldots,n_{\mathrm{sys}} + n + m$, followed by standard autoregressive answer decoding.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate LIRA on two complementary benchmarks: *LoCoMo* [19], a long-term conversational QA benchmark over multi-session dialogue histories, and *Loong* [30], a multilingual (English and Chinese) multi-document long-context QA benchmark spanning four reasoning categories: Spotlight Locating, Comparison, Clustering, and Chain of Reason, that vary in evidence locality and aggregation complexity. We use Set 1 and Set 2 (10–50K and 50–100K tokens) Loong; full dataset statistics are deferred to Appendix B.1.

Backbone models. We evaluate all methods with three open-weight backbones: Llama-3.1-8B-Instruct [8], Qwen2.5-7B-Instruct [22], and Qwen3-30B-A3B [34], covering Llama and Qwen model families and includes both dense and sparsely activated architectures.

Baselines. *Full Context* feeds the largest feasible raw history into the reader as a high-cost reference. Because LoCoMo provides gold supporting-turn annotations, we also report an *Oracle Evidence* reference: the same backbone is given only the gold supporting turns, serialized as text and re-encoded before answering. This is not a deployable memory method, but a per-backbone upperbound for turn-level localization. *Dense-RAG* retrieves dialogue turns or document chunks with BGE-M3 [20] and re-encodes the retrieved text with the same backbone under a matched token budget. *Conversational textual-memory systems* include Mem0 [4], A-MEM [33], and LightMem [6]; we evaluate them on LoCoMo, their native multi-session memory setting. *KV-cache compression baselines* include PyramidKV [3] and SnapKV [16]. For the repeated-query setting studied here, we retain the full-prefix KV states as a reusable store and apply each baseline's selection policy separately for each question. This amortized implementation avoids repeating the full-history prefill for every query. We match their per-query retained-history budgets to LIRA's budget. Thus, these implementations share the full-prefix persistent-storage requirement, while differing in how they construct the active cache used for answering. MemArt [35] is a *latent-memory baseline*, testing whether direct block-level reuse of latent state is sufficient.

Evaluation protocol. For each dataset–model pair, all methods share the same backbone, prompt, and decoding settings, except where an official baseline protocol differs. We report token-level F1, BLEU, and GPT-4.1-judged accuracy on LoCoMo, and GPT-4.1-judged accuracy under Loong's official multilingual evaluation [1]. The judge prompt templates are provided in Appendix E.

Table 1: Main results on LoCoMo. All metrics (F1, BLEU, ACC) are reported as percentages. Best per column in **bold**, second-best underlined (references excluded from ranking). Shaded rows are references; Δ reports LIRA's gap to the strongest non-reference baseline.

Method	Llama-3.1-8B			Qwen2.5-7B			Qwen3-30B-A3B		
	F1	BLEU	ACC	F1	BLEU	ACC	F1	BLEU	ACC
Oracle Evidence	62.83	55.48	84.72	62.28	53.95	85.69	61.38	51.24	85.76
Full Context	51.43	44.23	76.43	45.23	38.67	70.66	52.33	44.86	83.14
Text retrieval									
Dense-RAG	52.01	44.89	77.10	51.28	44.47	71.30	50.90	44.00	76.87
Textual memory									
Mem0	14.89	10.50	50.85	33.83	28.00	50.00	34.37	28.74	55.89
A-MEM	20.80	16.70	43.70	25.20	20.50	43.90	29.20	25.10	45.70
LightMem	30.59	23.93	67.01	42.39	35.45	63.70	44.15	37.19	73.51
KV cache compression									
PyramidKV	51.81	44.43	76.85	26.21	19.30	65.67	51.97	44.57	80.63
SnapKV	52.14	44.59	77.04	32.42	25.86	60.96	48.00	40.30	80.11
Latent state memory									
MemArt	47.72	40.29	74.43	41.67	33.54	71.35	51.54	44.54	76.39
LIRA (Ours)	**56.86**	**49.47**	**80.40**	**51.35**	**44.51**	**71.60**	**54.57**	**47.56**	**81.56**
Δ vs. best baseline	+4.72	+4.58	+3.30	+0.07	+0.04	+0.25	+2.60	+2.99	+0.93

4.2 Long-term Conversational QA on LoCoMo

Table 1 reports the main results on LoCoMo. LIRA is the strongest non-oracle method across all three backbones and surpasses the high-cost Full Context reference on eight of nine metrics; the only exception is judged accuracy on Qwen3-30B-A3B, where Full Context retains a 1.58-point lead but LIRA still gains +2.24 F1 and +2.70 BLEU, indicating tighter token-level alignment under the shorter context. Dense-RAG is competitive on LoCoMo, indicating that most queries match relevant turns by surface lexical or semantic similarity alone, so a strong embedding retriever already covers the core difficulty. Conversational textual-memory systems lag because their memory is written by the same 7B–8B backbone capping construction quality and the rewrite step can drop answer-critical spans with no query-time recourse. KV compression baselines are backbone-unstable. On Qwen2.5-7B, PyramidKV and SnapKV drop sharply in F1 (26.21, 32.42) while judged accuracy holds (65.67, 60.96): compressed-cache decoding preserves semantic correctness but loosens the concise answer format needed for high token overlap. Both methods select positions by proxies for full-cache fidelity rather than query-conditioned evidence, sensitive to backbone-specific attention statistics, e.g., sinks, head specialization, layerwise concentration, that differ across model families. This supports our view that runtime cache compression and memory readout optimize different objectives. LIRA shares the latent-memory interface with MemArt but adds query-conditioned readout planning; the resulting 9.14 / 9.68 / 3.03 F1 gap across backbones isolates this contribution. Per-stage latency breakdowns and a quality–latency trade-off analysis are provided in Appendix A.

4.3 Class-wise Behavior Analysis on Loong

We use Loong as a diagnostic for non-conversational, multilingual long-context QA. Table 2 shows that LIRA is the strongest method across all four categories, with the largest gains at the two ends of the locality spectrum. On *Spotlight Locating* which mainly single-anchor questions, LIRA leads the strongest baseline by 5.03 points (66.28 vs. 61.25), consistent with calibrated endogenous localization being most useful when evidence is sharply focal. On *Clustering* which requires diffuse-evidence aggregation, LIRA leads by 4.09 points (38.90 vs. 34.81); this gain is driven by coverage-aware readout planning: when the concentration diagnostic flags a diffuse evidence distribution, the planner extends the readout with long-range residuals, restoring the global context. Dense-RAG's behavior on Loong is also informative. Its 15.40 overall accuracy which is far below Full Context (41.23), and in sharp contrast to its competitive LoCoMo performance, indicates that Loong's multi-document questions resist surface-level lexical matching against retrieved chunks. Under the same selection of positions, latent-state readout yields substantially stronger answering than text-native re-encoding, supporting our broader claim that full-prefix latent states are more robust than re-prefilled text when the answer depends on representational influence accumulated across the document. Full aggregate Loong results over both length sets and all evaluated backbones are deferred to the Appendix D.

Table 2: Accuracy comparison across four reasoning categories on Loong (Llama-3.1-8B, Set 1). Results are reported as percentages. Best per column in **bold**, second-best underlined; the reference row is shaded; Δ reports LIRA's gap to the strongest non-reference baseline.

Method	Spotlight Locating	Comparison	Clustering	Chain of Reason	Overall
Full Context	58.49	32.73	33.23	38.14	41.23
Dense-RAG	20.32	16.68	8.01	18.72	15.40
PyramidKV	61.25	35.12	34.23	50.90	43.84
SnapKV	60.58	36.57	34.81	51.32	44.32
MemArt	37.17	31.68	24.40	33.85	30.78
LIRA (Ours)	**66.28**	**37.20**	**38.90**	**53.72**	**47.51**
Δ vs. best baseline	+5.03	+0.63	+4.09	+2.40	+3.19

4.4 Ablation Studies

Robustness to Incomplete Localization. To isolate the effect of the readout interface, we compare LIRA with a text-native re-prefill control under *identical selected positions*. We disable coverage-aware completion in this controlled comparison since its bridge and sparse-fill steps yield non-contiguous selections that suit latent-state assembly but produce a fragmented, non-fluent prompt

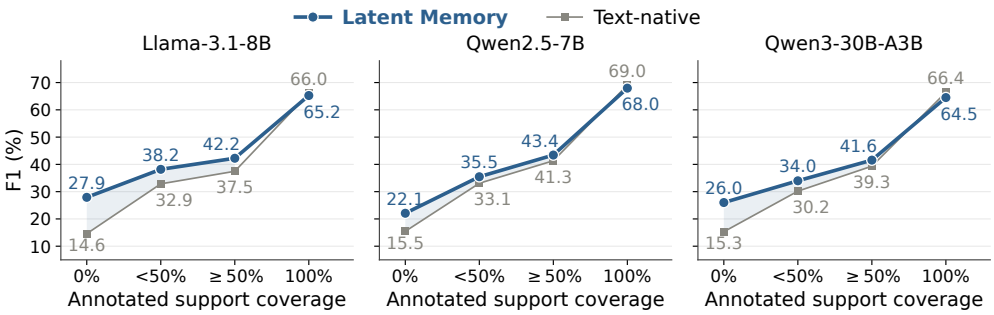


Figure 5: **Robustness to incomplete localization.** F1 on LoCoMo stratified by oracle evidence coverage; LIRA and the text-native re-prefill control share the same selected positions. Latent-state gains are largest under zero or partial coverage.

for the text-native baseline. Disabling it leaves the readout representation as the only remaining difference between the two systems.

Under this controlled setting, we stratify all LoCoMo samples by the fraction of annotated oracle evidence covered by the selected context. Figure 5 shows a consistent pattern across all three backbones: latent-state readout is most beneficial when localization is incomplete or entirely missed. LIRA improves F1 over text-native re-prefill by 13.33, 6.62, and 10.70 points on Llama-3.1-8B, Qwen2.5-7B, and Qwen3-30B-A3B, respectively. The selected positions in this bucket contain none of the annotated evidence, yet the retained latent states still encode contextual influence accumulated from the omitted earlier turns during the original full-prefix pass; the reader exploits this preserved influence to recover answer support, yielding stronger robustness to localization errors than the text-native control. When oracle evidence is fully covered, the two interfaces become comparable and text-native re-prefill is slightly ahead by 1–2 points across backbones: once all explicit answer evidence is already present in the selected text, the additional full-prefix contextualization preserved by latent states provides less benefit and may introduce non-support signal from the broader history. We further provide the qualitative results in Appendix C.

Budget Sensitivity. Figure 6 varies the active reader budget B on LoCoMo. We retain the controlled setting from 4.4: coverage-aware completion is disabled, and LIRA is compared against text-native re-prefill, MemArt, and Dense-RAG under matched budgets. Across the sweep, LIRA consistently outperforms the text-native re-prefill variant and MemArt, and matches the Full Context reference with only 1,024 retained dialogue tokens. Dense-RAG improves steadily as its retrieval budget grows, narrowly exceeding LIRA at the largest setting and indicating that text retrieval becomes competitive once enough raw context is admitted. In the tight-budget regime where compact memory matters most, however, latent memory yields substantially higher F1 than text-native memory, supporting our claim that preserving full-prefix contextualization improves budget efficiency under imperfect localization.

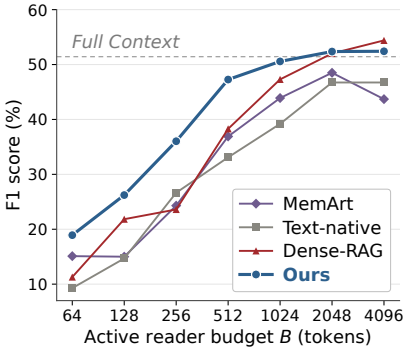


Figure 6: **Budget sensitivity on LoCoMo.** F1 score vs. active reader budget B on Llama-3.1-8B.

Table 3: Component analysis on LoCoMo with Llama-3.1-8B. All metrics are reported as percentages.

Variant	Calibration	Coverage Compl.	RoPE Re-idx.	F1	BLEU	ACC
Full LIRA	✓	✓	✓	**56.86**	**49.47**	**80.40**
− Calibration	×	✓	✓	41.93	35.38	62.99
− Coverage Completion	✓	×	✓	53.93	47.12	75.19
− RoPE Re-indexing	✓	✓	×	14.58	11.56	19.48

Component Analysis. Table 3 ablates the three components of LIRA on LoCoMo with Llama-3.1-8B. *RoPE re-indexing* produces the largest drop, with F1/ACC falling from 56.86/80.40 to 14.58/19.48 when removed; without it, retained keys carry the RoPE phases of their original positions while the reader queries them at compact coordinates, leaving attention under an inconsistent positional geometry. *Attractor calibration* is the dominant localization gain: removing the neutral-probe subtraction drops F1/ACC by 14.93/17.41, confirming that raw question attention is dominated by query-independent attractors and that subtracting the baseline is what exposes question-specific support. *Coverage-aware completion* contributes a smaller but consistent improvement of 2.93/2.35/5.21 on F1/BLEU/ACC, matching its intended role as a gated repair mechanism that activates only when the concentration diagnostic flags diffuse anchors.

5 Conclusion

We presented LIRA, a latent-memory reader that reads full-prefix contextualized states from an amortized memory store rather than re-encoding selected text. It combines calibrated endogenous localization, coverage-aware readout planning, and position-consistent latent-state assembly. Across LoCoMo and Loong on three open-weight backbones, LIRA is the strongest non-oracle compact-memory method, with the largest gains under partial or zero retrieval coverage. These findings suggest that long-context memory systems should optimize not only evidence selection but also the resilience of selected representations under imperfect localization.

Acknowledgments and Disclosure of Funding

This research is supported by the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP), Grant No. I2501E0053; the National Research Foundation, Singapore, under its NRF Professorship Award No. NRF-P2024-001; and the National Research Foundation, Singapore, under its Smart Nation & Digital Government Translational R&D Grant (TRANS), Award No. TRANS2023-TGC03.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] 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)*, pages 3119–3137, 2024.

[3] Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, et al. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*, 2024.

[4] Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

[5] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

[6] Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, et al. Lightmem: Lightweight and efficient memory-augmented generation. *arXiv preprint arXiv:2510.18866*, 2025.

[7] In Gim, Guojun Chen, Seung-seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt cache: Modular attention reuse for low-latency inference. *Proceedings of Machine Learning and Systems*, 6:325–338, 2024.

[8] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[9] Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*, 2025.

[10] Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*, 2025.

[11] 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? In *First Conference on Language Modeling (COLM)*, 2024.

[12] Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models, 2023.

[13] Jang-Hyun Kim, Jinuk Kim, Sangwoo Kwon, Jae W Lee, Sangdoo Yun, and Hyun Oh Song. Kvzip: Query-agnostic kv cache compression with context reconstruction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[14] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-Embed: Improved techniques for training LLMs as generalist embedding models. In *International Conference on Learning Representations (ICLR)*, 2025.

[15] 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.

[16] 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. In *NeurIPS*, 2024.

[17] 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.

[18] Dongyang Ma, Yan Wang, and Lan Tian. Block-attention for efficient prefilling. *arXiv preprint arXiv:2409.15355*, 2024.

[19] Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, 2024.

[20] Multi-Linguality Multi-Functionality Multi-Granularity. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024.

[21] Charles Packer, Vivian Fang, Shishir_G Patil, Kevin Lin, Sarah Wooders, and Joseph_E Gonzalez. Memgpt: towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

[22] Qwen et al. Qwen2.5 technical report, 2025.

[23] Shidu Ren, Rong-cheng Tu, Hang Zhou, and Xiao Luo. Pace-mem: Prefix-aware complementary evidence for multimodal agents, 2026.

[24] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the association for computational linguistics*, 8:842–866, 2020.

[25] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: recursive abstractive processing for tree-organized retrieval. In *ICLR*. OpenReview.net, 2024.

[26] Wenhao Sun, Rong-Cheng Tu, Yifu Ding, Zhao Jin, Jingyi Liao, Yongcheng Jing, and Dacheng Tao. Spa-cache: Singular proxies for adaptive caching in diffusion language models. *CoRR*, abs/2602.02544, 2026.

[27] Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context llm inference. *arXiv preprint arXiv:2406.10774*, 2024.

[28] Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4593–4601, 2019.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[30] Minzheng Wang, Longze Chen, Fu Cheng, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, Yunshui Li, Min Yang, Fei Huang, and Yongbin Li. Leave no document behind: Benchmarking long-context llms with extended multi-doc QA. In *EMNLP*, pages 5627–5646. Association for Computational Linguistics, 2024.

[31] Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Haotian Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. *arXiv preprint arXiv:2404.15574*, 2024.

[32] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *ICLR*, 2024.

[33] Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*, 2025.

[34] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[35] Yuan Zeng, Pengfei Zuo, Min Lyu, Xingkun Yang, Huatao Wu, Yinlong Xu, and Zhou Yu. KVCache-centric memory for LLM agents, 2026.

[36] Yanzhao Zhang and et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025.

[37] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025.

[38] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *NeurIPS*, 2023.

[39] 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, pages 19724–19731, 2024.

A Efficiency Analysis

Table 4 reports a stage-wise efficiency breakdown on LoCoMo. We decompose inference into *Build*, *Retrieve*, and *Decode*. *Build* denotes the one-time full-history prefill cost; *Retrieve* denotes the per-query cache selection, retrieval, or latent readout assembly cost; and *Decode* denotes autoregressive answer generation. The reported times are amortized milliseconds per QA example.

In our LoCoMo timing split, 1,500 QA examples share only 10 long conversations, so the build cost is amortized over many questions, while retrieval and decoding are incurred for each query. We also report the average generated length and decoding milliseconds per generated token, which separates per-token cache efficiency from end-to-end wall-clock latency. This distinction is important because LoCoMo answers are short. Compact readouts can substantially reduce per-token decoding cost, but the absolute decoding saving may still be smaller than the additional per-query retrieval or assembly overhead.

We summarize our findings as below:

- **The decode-time advantage of KV compression is muted on LoCoMo by the dataset's short response length.** The per-token cost (*ms/token*) shows that compaction methods decode about 1.5–2.0× faster per token than Full Context. However, because LoCoMO answers are concise by design, averaging well under 15 generated tokens, the absolute decode-time saving is small. A compact method is faster end-to-end only when

$$D_{\text{full}} - D_{\text{compact}} > R_{\text{compact}},$$

  where D is decode time and R is the per-query retrieval or cache-selection overhead. In this short-output regime, the per-token speedup does not always offset the additional per-query overhead.

- **LIRA achieves the lowest decode latency among compaction methods.** Across all three backbones, LIRA produces the fewest generated tokens (7.21 / 6.01 / 8.21 on LLaMA-3.1-8B / Qwen2.5-7B / Qwen3-30B-A3B) and obtains the lowest *Decode* time among compact methods. This is consistent with its query-conditioned latent readout: when the localized evidence is sufficiently concentrated, the reader operates on a focused compact context rather than a uniformly budget-filled cache. However, LIRA does not always have the lowest *ms/token*, and its larger retrieval and assembly cost dominates total latency in the current implementation.

- **SnapKV and PyramidKV show a Qwen2.5-specific gap between token-level F1 and judged correctness.** On Qwen2.5-7B, SnapKV and PyramidKV obtain much lower F1 (32.42 and 26.21, compared with 45.23 for Full Context), while judged accuracy degrades less severely. Their generations are also longer than Full Context's (12.28 and 11.89 tokens versus 7.89), so the F1 drop is not caused by truncated answers. Instead, the compressed-cache reader appears to preserve partial semantic correctness while weakening the concise answer format needed for high token overlap on short-answer QA. This supports our main analysis that cache-compression policies optimize full-cache fidelity rather than query-conditioned evidence support, and can interact differently with backbone-specific attention patterns.

- **Text-memory baselines are reported separately because their costs are not directly decomposable.** Methods such as LightMem construct textual memories through multiple LLM passes and do not share the same build–retrieve–decode decomposition as KV-cache or latent-state readout methods. We therefore report their quality in the main results and, when included in the efficiency appendix, mark their timing as end-to-end memory construction/retrieval cost rather than as a directly comparable cache-readout latency.

- **LIRA's Retrieve cost trades latency for quality.** Our calibrated selection is 2.0–2.4× slower than SnapKV's at the *Retrieve* stage, but yields 4.7–25.1 absolute F1 points of improvement. The Pareto trade-off favors quality at the small per-query cost (< 310 ms even on the 30B-A3B model).

Table 4: Per-query efficiency and F1 score on LoCoMo across three model backbones. *Build*: prefill latency averaged per query. *Retrieve*: token selection / cache compaction cost. *Decode*: total generation latency. *Total* = Build + Retrieve + Decode. *Gen len*: average tokens generated. *ms/token* = Decode / Gen len. Latencies in ms (lower is better); F1 higher is better. **Bold** marks the best value per column.

(a) LLaMA-3.1-8B

Method	Build (ms)	Retrieve (ms)	Decode (ms)	Total (ms)	Gen len	ms/token	F1 (%)
Full context	44.01	0.00	217.20	261.21	8.85	24.54	51.43
SnapKV	43.12	**94.63**	110.50	**248.25**	7.62	14.50	52.14
PyramidKV	**42.57**	150.64	116.00	309.21	8.83	**13.14**	51.81
LIRA	42.69	232.91	**98.40**	374.00	7.21	13.65	**56.86**

(b) Qwen2.5-7B

Method	Build (ms)	Retrieve (ms)	Decode (ms)	Total (ms)	Gen len	ms/token	F1 (%)
Full context	38.05	0.00	181.34	219.39	7.89	22.98	45.23
SnapKV	35.12	**112.01**	155.20	**302.33**	12.28	**12.64**	32.42
PyramidKV	**34.42**	119.25	162.12	315.79	11.89	13.63	26.21
LIRA	36.86	221.61	**89.90**	348.37	6.01	14.96	**51.35**

(c) Qwen3-30B-A3B

Method	Build (ms)	Retrieve (ms)	Decode (ms)	Total (ms)	Gen len	ms/token	F1 (%)
Full context	70.12	0.00	288.11	358.23	10.23	28.16	52.33
SnapKV	69.30	**155.21**	142.20	**366.71**	10.01	**14.21**	48.00
PyramidKV	**68.15**	200.98	161.90	431.03	9.98	16.22	51.97
LIRA	68.33	312.40	**128.70**	509.43	8.21	15.68	**54.57**

B Implementation Details and Hyperparameter Setting

B.1 Dataset Details

LoCoMo. [19] construct LoCoMo through a machine–human hybrid pipeline: an LLM agent grounded on a multi-sentence persona and a causally connected event graph generates multi-session dialogues, which are then verified and edited by human annotators for long-range coherence and event consistency. The publicly released subset (LoCoMo-10) contains 10 conversations spanning up to 32 sessions and roughly 600 turns each, with an average per-conversation length around 9K tokens. The accompanying QA benchmark covers five reasoning categories that probe distinct memory-retrieval skills: *single-hop*, *multi-hop*, *temporal*, *open-domain*, and *adversarial*. Following prior work on compact-memory QA [4, 6, 35] we evaluate on the first four categories and exclude the adversarial subset. This yields roughly 1,500 QA pairs across the 10 LoCoMo-10 conversations, and report token-level F1, BLEU, and GPT-4.1-judged accuracy under matched decoding settings across all methods.

Loong. [30] introduce Loong as a multi-document long-context QA benchmark. Each instance averages 11 documents drawn from three real-world scenarios, financial reports, legal cases, and academic papers, with bilingual coverage of English and Chinese. Loong organizes test instances along two orthogonal axes: four reasoning categories that vary in evidence locality and aggregation complexity: *Spotlight Locating*, *Comparison*, *Clustering*, and *Chain of Reasoning* and four length sets (10–50K, 50–100K, 100–200K, and >200K tokens). Given the effective context limits and computational constraints of our three open-weight backbones (Llama-3.1-8B, Qwen2.5-7B, Qwen3-30B-A3B), we evaluate on Set 1 and Set 2, with average context lengths of 37.8K and 75.6K tokens, respectively, and follow Loong's official multilingual evaluation protocol with a GPT-4.1 judge.

B.2 Hyperparameter Setting

Table 5 lists the hyperparameters used for LIRA. Unless otherwise stated, the same configuration is used across all backbones and datasets. All experiments were run on one NVIDIA B200 GPU.

Table 5: Hyperparameter settings.

Parameter	Value
Active reader budget B	2,048 (LoCoMo) / 8192 (Loong)
Localization layers K_{loc}	4 (LoCoMo) / 8 (Loong)
Neutral probe $\tilde{q}$	*"Tell me about it."*
Anchor mix weight λ	0.7
Number of anchors N_{anchor}	2
Coverage threshold τ_{cov}	25.0
Bridge mix weight α	0.5
Number of bridge segments N_{br}	2 (LoCoMo) / 0 (Loong)
Max new tokens	32 (LoCoMo) / 512 (Loong)

C Qualitative Results

C.1 Coverage Distribution and Recovery Statistics

We measure annotated support coverage as the fraction of gold supporting evidence covered by the selected history positions. Table 6 reports the distribution over all 1,525 LoCoMo questions with coverage-aware completion enabled. Coverage is computed from the final selected positions; gold annotations are used only for evaluation and are unavailable to the localizer and planner.

Table 6: Annotated support coverage of the final readout on LoCoMo with coverage-aware completion enabled. Each entry reports the number of questions and its percentage of the 1,525-question evaluation set.

Coverage	Llama-3.1-8B	Qwen2.5-7B	Qwen3-30B-A3B
0%	477 (31.3%)	596 (39.1%)	325 (21.3%)
$(0\%, 50\%)$	149 (9.8%)	132 (8.7%)	146 (9.6%)
$[50\%, 100\%)$	181 (11.9%)	167 (11.0%)	203 (13.3%)
100%	718 (47.1%)	630 (41.3%)	851 (55.8%)

C.2 Case Studies for Robustness Evaluation

We present three representative qualitative case studies illustrating distinct failure modes of text-native readout on LoCoMo and how LIRA addresses them. Each case reports the oracle evidence, and the resulting predictions from both systems. The two methods consume the *same* retrieved positions but differ in their readout: text-native baselines re-prefill the text at these positions, whereas LIRA use the latent state.

Case Study 1: Single-hop QA with semantic similar distractor.

Question. *What new yoga pose did Deborah try?*

Oracle evidence.

```
[9:17 am, 26 June 2023] Deborah: I'd rather show you a photo. This is
also a new yoga pose that we tried. It is a tree pose.
```

Retrieved text.

```
[9:48 am, 4 February 2023] Jolene: Wow, that's awesome! What new
poses did you try?
[9:48 am, 4 February 2023] Deborah: We were trying out different dance
poses, like Warrior II, which builds strength and boosts focus. ...
```

Predictions.

Method	Output	
Gold answer	*tree pose*	—
Text-native	"*Warrior II, a modified pose*"	✗
LIRA (Latent Memory)	"*Tree pose*"	✓

Analysis. Retrieval surfaces a topically similar distractor ("*dance* poses, Warrior II" from February) instead of the oracle evidence ("tree pose" from June). Text-native compaction follows the mistaken retrieval; LIRA bypasses it through latent representation of the full conversation.

Case Study 2: Multi-hop QA with incomplete evidence.

Question. *Which countries did James visit in July 2022?*

Oracle evidence. The dataset annotates two evidence spans across separate turns; each mentions a city that must be resolved to its country.

```
[5:13 pm, 9 July 2022] James:  ...By the way, I bought air tickets to
Toronto, and I'm leaving the day after tomorrow evening.
[9:49 am, 22 July 2022] James:  Everything went great!  In addition, I
even managed to get out to another country.  The city of Nuuk, if you
know.  I stayed there quite a bit...
```

Retrieved text.

```
[9:49 am, 22 July 2022] John:  Yeah, the bond between us and our pets
is amazing...By the way, how was your trip?
[9:49 am, 22 July 2022] James:  Everything went great!  In addition, I
even managed to get out to another country.  The city of Nuuk, if you
know.  I stayed there quite a bit, but at least I had one more country
to add to my bucket list!...
```

(Retrieval surfaces only the July 22 turn; the July 9 mention of Toronto is missed.)

Predictions.

Method	Output	
Gold answer	*Canada, Greenland*	—
Text-native	"*Another country, besides the one he didn't mention, Nuuk*"	×
LIRA (Latent Memory)	"*Canada, Greenland*"	✓

Analysis. This question requires aggregating two cross-turn evidence spans (Toronto on July 9, Nuuk on July 22) and resolving each city to its country. Retrieval surfaces only the Nuuk turn and misses the Toronto mention. Text-native readout inadvertently reveals its own blind spot: its output ("*another country, besides the one he didn't mention*") registers that a second country exists but cannot name it, because the Toronto evidence is unreachable through the retrieved text alone. LIRA recovers the missing country from its latent memory of the earlier turn and produces the complete multi-entity answer.

Case Study 3: Temporal reasoning with misaligned timeline.

Question. *When did Gina lose her job at Door Dash?*

Oracle evidence.

```
[4:04 pm, 20 January 2023] Gina:  Sorry about your job Jon, but
starting your own business sounds awesome!  Unfortunately, I also
lost my job at Door Dash this month.  What business are you thinking
of?
```

Retrieved text.

```
[2:35 pm on 16 March, 2023] Gina:  Hey Jon!  Great to hear from you.
Been having some tough times lately.
[2:35 pm on 16 March, 2023] Jon:  Oof, sorry to hear that.  Whats up?
Is there anything I can do to help?
[2:35 pm on 16 March, 2023] Gina:  Thanks, Jon!  Appreciate your offer.
Since I lost my job at Door Dash, things have been tough.  But here's
```

`some good news - I've got something to share!...`

(Retrieval contains a later conversation discussing the aftermath of losing the job; the original January passage where Gina says "this month" is missed.)

Predictions.

Method	Output	
Gold answer	*January 2023*	—
Text-native	"*16 March 2023*"	✓
LIRA (Latent Memory)	"*January 2023*"	✓

Analysis. For the temporal-grounding question, the gold answer is encoded as the relative phrase "this month" anchored to the timestamp [20 January 2023]. Retrieval surfaces a later conversation about the relative event but misses the original January passage. Text-native memory reads only the retrieved March content and reports the timestamp it sees (*16 March 2023*). LIRA recovers the original January utterance from its latent memory and resolves "this month" against the correct anchor, producing *January 2023*.

C.3 Effectiveness of Calibrated Localization

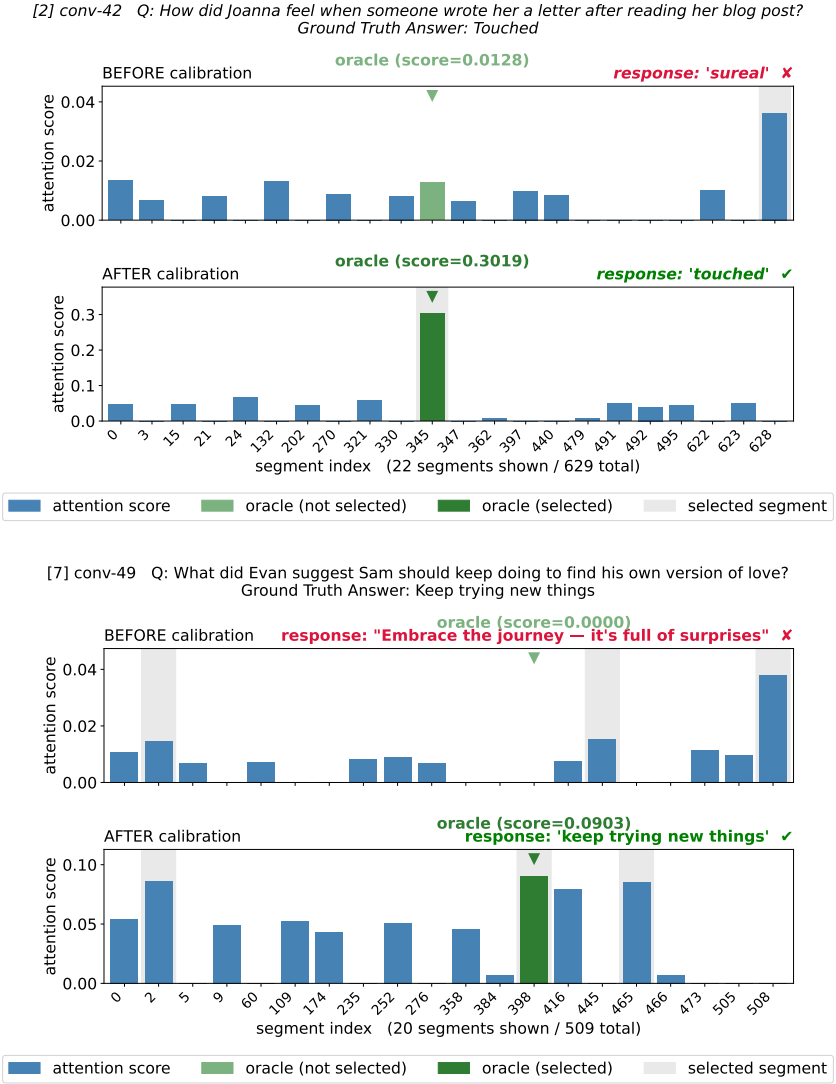


Figure 7: **Attractor calibration on two LoCoMo case studies.** Each panel shows segment-level localization scores along the dialogue history for one query.

Figure 7 shows two case studies on LoCoMo, with raw question-conditioned attention a_i on top and the calibrated score $s_i = [a_i - b_i]_+$ below. Both cases exhibit the same pattern: under raw attention the oracle supporting segment receives a magnitude indistinguishable from background noise — 0.0128 in conv-42 and 0.0000 in conv-49, and is dominated by query-independent attractors such as the most recent turn, leaving the segment outside the top-N_{anchor} anchors and the reader without access to the supporting evidence. After subtracting the neutral-probe baseline, the same oracle segment becomes a competitive or dominant peak in the calibrated distribution and is admitted into the readout plan, after which the reader recovers the gold answer in both cases.

D Full Results on Loong

Table 7 reports the full aggregate results on Loong Set1 and Set2 across all three backbones. LIRA consistently outperforms Full Context, Dense-RAG, and MemArt in every setting. Among all baseline methods, LIRA achieves the best accuracy in five of the six columns; the only exception is Set1 with Qwen2.5-7B, where SnapKV is slightly higher. Performance drops from Set1 to the longer Set2 for all methods, but LIRA remains the strongest method across all three backbones on Set2, indicating that the proposed readout planning remains effective under longer multilingual multi-document contexts.

Table 7: Accuracy on Set 1 and Set 2 of Loong across three model backbones. **Bold** marks the best per column.

	Set 1			Set 2		
Method	LLaMA-3.1-8B	Qwen2.5-7B	Qwen3-30B-A3B	LLaMA-3.1-8B	Qwen2.5-7B	Qwen3-30B-A3B
Full Context	41.23	33.13	59.14	29.32	15.81	48.13
Dense-RAG	15.40	16.13	17.58	11.54	11.14	14.23
PyramidKV	43.84	39.09	62.11	32.14	25.65	50.76
SnapKV	44.32	**39.64**	62.92	32.12	27.10	50.88
MemArt	30.78	28.85	41.42	20.70	19.19	22.78
LIRA	**47.51**	38.41	**63.09**	**34.09**	**28.24**	**50.98**

E Prompt Template

Prompt Template: QA on LoCoMo

```
You are an intelligent memory assistant answering questions from Dialogue D.
RULES:
    1. Answer using ONLY information from Dialogue D.
    2. Identify WHO the question asks about. Answer based on what THAT person said or
       did. Do not confuse different speakers.
    3. For ‘‘when’’ questions: compute concrete dates from timestamps. Format: ‘‘D
       Month YYYY’’, ‘‘Month YYYY’’, or ‘‘YYYY’’.
    4. For questions asking about multiple items: include ALL items mentioned,
       separated by commas.
    5. If the dialogue contains contradictory information, use the most recent
       evidence by timestamp.
    6. Keep answers concise. For list questions, use commas, no full sentences.
    7. Output ONLY the final answer, no explanation or elaboration.
```

Prompt Template: QA on Loong

```
You answer questions based on reference documents. Use exactly ONE format:
    1. Single value (default -- numbers, names, dates): output EXACTLY ONE value and
       STOP.
    2. List of items: output ONE list only and STOP.
    3. JSON dict: output ONE complete JSON object and STOP.
```

Absolute rules:

- Extract values verbatim (currency signs, parentheses for negatives, etc.).
- No reasoning, no preamble, no trailing commentary.

Prompt Template: Binary Judge on LoCoMo

Your task is to label a generated answer to a question as either "CORRECT" or "WRONG".

You will be given the following data:

1. A question posed by one user to another user.
2. A gold answer, which is the ground-truth answer.
3. A generated answer, which you should judge.

The question asks about something one user should know about the other user based on their prior conversations. The gold answer is usually concise and short. For example:

Question: Do you remember what I got the last time I went to Hawaii?
Gold answer: A shell necklace

Judging rules:

- Be generous when judging semantic correctness.
- The generated answer may be longer than the gold answer.
- If the generated answer touches on the same topic as the gold answer, label it "CORRECT".
- For time-related questions, accept different surface forms if they refer to the same date or time period.
- For example, "May 7th" and "7 May" should be considered the same date.
- Otherwise, label the answer as "WRONG".

Input:
Question: {question}
Gold answer: {gold_answer}
Generated answer: {generated_answer}

Output rule:
Return exactly one JSON object with the key "label" and value either "CORRECT" or "WRONG".
Do not include any explanation, reasoning, preamble, or trailing commentary. Do not include both "CORRECT" and "WRONG".

Valid outputs:
{"label": "CORRECT"}
{"label": "WRONG"}

Prompt Template: Score Judge on LoCoMo

[Question]
{question}

[Gold Answer]
{gold_answer}

[The Start of Assistant's Predicted Answer]
{generated_answer}
[The End of Assistant's Predicted Answer]

[System]
We would like to request your feedback on the performance of the AI assistant in response to the user question displayed above according to the gold answer. Please use the following listed aspects and their descriptions as evaluation criteria:

- Accuracy and Hallucinations: The assistant's answer is semantically consistent with the gold answer. The numerical value and order need to be accurate, and there should be no hallucinations.

```
    • Completeness: Referring to the reference answers, the assistant's answer
      should contain all the key points needed to answer the user's question.
      Further elaboration on these key points can be omitted.

Please rate whether this answer is suitable for the question. Please note that the gold
answer can be considered as a correct answer to the question.

The assistant receives an overall score on a scale of 1 to 100, where a higher score
indicates better overall performance. If the assistant's answer and the gold answer
fully meet the above criteria, its overall rating should be the full marks, 100.

Please first provide a comprehensive explanation of your evaluation, avoiding any
potential bias. Then, output a line indicating the score of the Assistant.

Output format:
Please output with the following format, where the score is a scale of 1 to 100 and the
format must be strictly followed:

<start output>
Evaluation evidence: your evaluation explanation here, no more than 100 words
Rating: [[score]]
<end output>

For example:
Rating: [[100]]

Now, start your evaluation:
```

F Limitations and Broader Impacts

Limitations. LIRA is designed for long-context question answering with an amortized latent-memory store, and our empirical conclusions should be interpreted within this setting. First, the evaluation is limited to two benchmarks, LoCoMo and Loong, and three open-weight backbones. Although these benchmarks cover long-term conversational memory, multilingual multi-document QA, and several reasoning categories, they do not exhaust the diversity of real deployments, such as interactive agents, tool-using systems, streaming updates, or domains with strict factuality and privacy requirements. Second, our Loong experiments cover Set1 and Set2 because of effective context-length and computational constraints; therefore, we do not claim that the same performance trends necessarily hold for the longest Loong settings or for histories beyond the tested length range. Third, LIRA operates on reusable full-prefix latent states, which requires access to latent states and standard positional bookkeeping during readout assembly. This design is compatible with the RoPE-based autoregressive backbones evaluated in this work, and extending it to other cache layouts or positional encodings mainly requires implementation-specific adaptation. Finally, LIRA adds a lightweight per-query planning and assembly stage before decoding from the compact latent readout. The resulting latency profile depends on the workload: the one-time full-prefix build is naturally amortized when many queries share the same history, while short-answer QA leaves less opportunity for compact decoding to offset retrieval and assembly overhead. Optimizing this systems path is orthogonal to the proposed readout objective and may further improve practical latency.

Broader impacts. This work may have positive impacts by improving the reliability and budget efficiency of long-context memory systems. Compact latent readouts can reduce the amount of active context needed at answer time, potentially lowering inference cost and making long-context QA more accessible for applications such as document analysis, conversational assistants, and personal knowledge management. The distinction between evidential coverage and inferential resilience may also help future memory systems become more robust to imperfect retrieval, reducing failures caused by missing or partially selected evidence. However, latent-memory systems also introduce risks. Because LIRA stores and reuses full-prefix latent states, the memory may encode sensitive information even when the corresponding text is not explicitly retrieved or displayed. This raises privacy concerns in deployments over personal conversations, enterprise documents, medical records, or other sensitive histories. Latent states are also less interpretable than textual memories, which can make auditing and debugging more difficult when the system produces an incorrect or privacy-sensitive answer.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction are supported by the method description in Section 3 and the empirical results in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses several limitations and practical constraints, presented in Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present formal theoretical results, theorems, or proofs. The equations and algorithmic descriptions are used to specify the proposed method rather than to establish theoretical guarantees.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper discloses the information needed to reproduce the main experimental results to the extent relevant to the paper's claims. Section 3 provides the full algorithmic specification; Section 4.1 specifies the datasets, backbones, baselines, evaluation metrics, and shared evaluation protocol; Appendix B and E provides the hyperparameters and prompt templates.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper uses existing public benchmarks and open-weight backbone models, but the current draft does not provide an anonymized code.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting is specified in Section 4.1, including the LoCoMo and Loong evaluation sets, the three open-weight backbones, the compared baselines, the budget-matched evaluation protocol, and the reported metrics. Additional implementation details are provided in the appendix B and E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to the computational cost of the LLMs, it is hard to run the statistical significance for our experiments.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computer resources information is provided in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses existing benchmarks and open-weight language models, does not introduce new human-subject data collection, and is presented as an anonymized submission.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discussed both positive and negative broader impacts in Appendix F.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a new high-risk pretrained model, image generator, scraped dataset, or other asset with elevated misuse risk.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper properly cites the datasets and tools (e.g., GPT-4.1) used in the experiments, ensuring proper credit and adherence to licensing terms.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce or release a new dataset, benchmark, pretrained model, or other standalone asset.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not conduct new crowdsourcing experiments or new research with human subjects. All experiments are performed on existing benchmarks and model outputs.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve new human-subject experiments, new participant recruitment, or new crowdsourced data collection. Therefore, IRB approval or equivalent review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs are central to the research: the proposed method operates on latent states of open-weight language-model backbones, and the experiments evaluate Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct, and Qwen3-30B-A3B. The paper also reports GPT-4.1-judged accuracy and provides the judge prompt templates in Appendix D.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.