SLICE-MEM: SALIENCE-GATED LIGHTWEIGHT INFORMATION CURATION FOR EFFICIENT LLM AGENT MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term memory enables large language model (LLM) agents to maintain personalized and temporally coherent interactions, but repeated LLM inference for memory admission, extraction, and organization introduces substantial cost and latency. We present SLICE-Mem, a lightweight memory framework that combines salience-guided admission and predicate-centered information extraction. A compact classifier identifies user turns worth retaining, and a semantic-dependency-based extractor converts admitted turns into concise memory entries. This online pipeline requires no generative LLM calls, while cross-entry reconciliation is deferred to an offline consolidation stage. We evaluate SLICE-Mem on the weekly subset of Memora and LongMemEval-S using 8B and 30B Qwen backbones for memory construction and consolidation on both benchmarks. Compared with the generative-memory baseline MemOS under the Qwen3-30B configuration, SLICE-Mem's online pipeline achieves 46.60 versus 46.68 FAMA (Forgetting-Aware Memory Accuracy) on Memora Weekly and 56.8% versus 59.3% accuracy on LongMemEval-S, while reducing total LLM token consumption by 92.4% and 99.5%, respectively. These results demonstrate that SLICE-Mem can preserve useful evidence for long-term memory with substantially less reliance on LLM-based memory construction.

1 INTRODUCTION

Long-term memory is essential for large language model (LLM) agents that interact with users across sessions. Such agents must retain preferences, past events, commitments, and evolving constraints to provide coherent and personalized assistance. Yet the underlying LLM is effectively stateless across independent sessions, and a finite context window cannot serve as durable memory. Agent systems therefore externalize interaction histories into persistent stores (Packer et al., 2023; Park et al., 2023; Zhong et al., 2024). As these histories grow, memory construction becomes a bottleneck: many turns are transient or redundant, indiscriminate storage introduces retrieval noise, and repeatedly invoking LLMs for filtering, extraction, and organization incurs substantial latency and token cost. Efficient long-term memory must therefore determine, before expensive processing, which information merits retention and how it can be represented with minimal online computation.

Existing memory systems primarily optimize downstream retrieval or answer accuracy. Structured and hierarchical approaches improve organization (Kang et al., 2025; Xu et al., 2025), but often call a general-purpose LLM throughout the memory lifecycle to select interactions, extract facts, detect conflicts, and consolidate records. Retrieval-augmented generation supplies external evidence at answer time (Lewis et al., 2020), but long-context use remains challenging (Bai et al., 2024; Liu et al., 2024); neither directly controls admission to persistent memory. Hierarchical and graph-based stores likewise rely on LLM-generated summaries, graphs, or atomic units (Gutiérrez et al., 2024; Kang et al., 2025; Sarthi et al., 2024; Xu et al., 2025). Recent lightweight designs reduce repeated summarization or defer maintenance to an offline phase (Fang et al., 2026; Hu et al., 2025), but early admission and non-generative extraction remain underexplored as a unified construction strategy. This gap affects both efficiency and reliability: irrelevant or stale entries enlarge the retrieval space and may steer later agent behavior (Xiong et al., 2026).

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

We introduce **SLICE-Mem**, a **S**alience-gated **L**ightweight **I**nformation **C**uration framework for **E**fficient agent memory. Inspired by selective encoding and sleep-dependent consolidation (Atkinson & Shiffrin, 1968; Diekelmann & Born, 2010; Rasch & Born, 2013), SLICE-Mem separates lightweight online formation from deferred cross-entry reconciliation. A compact Salience Gate learns to identify user turns worth retaining from consensus labels produced by three heterogeneous LLM teachers and released memory-admission annotations. By considering each target turn together with its preceding dialogue context, the Gate filters low-value content before parsing and embedding. For admitted turns, a deterministic extractor uses semantic dependencies to identify predicates and their associated arguments and modifiers, then assembles the corresponding text spans into concise memory entries. This process preserves surface evidence and turn-level provenance without generative LLM calls during online admission or extraction. Offline consolidation subsequently addresses redundancy and conflicting information through relation classification and entry-status updates while preserving the original memory text. We evaluate SLICE-Mem on Memora Weekly and LongMemEval-S using 8B and 30B Qwen backbones for memory construction and consolidation, with answer generation and judging held fixed. In the 30B comparison, the online pipeline achieves 46.60 versus 46.68 FAMA on Memora Weekly and 56.8% versus 59.3% accuracy on LongMemEval-S relative to the generative-memory baseline MemOS (Li et al., 2025). It reduces total LLM token consumption by 92.4% and 99.5% and reported memory-construction time by 84.8% and 61.0%, respectively. Separate evaluation of the online and consolidated states reveals task- and backbone-dependent effects of consolidation. Together, these results demonstrate a favorable effectiveness–cost trade-off for selective, non-generative memory formation while highlighting the need for evidence-preserving consolidation.

Our main contributions are threefold:

- We develop a compact Salience Gate that learns context-aware memory admission from consensus-based LLM annotations and released admission labels, filtering user turns before extraction and embedding without generative LLM inference at deployment.
- We design a deterministic, semantic-dependency-guided extractor that converts admitted turns into concise memory entries while preserving surface evidence and turn-level provenance. Together with the Gate, it enables online memory formation without generative LLM calls.
- We evaluate SLICE-Mem on two long-term memory benchmarks with 8B and 30B Qwen backbones, demonstrating competitive answer quality at substantially lower token and time cost, and identifying the task-dependent benefits and limitations of offline consolidation.

2 Related Work

Memory architectures. Persistent-memory agents differ primarily in how they represent and organize past interactions. Generative Agents retrieve chronological memories using relevance, recency, and LLM-estimated importance, and periodically synthesize reflections (Park et al., 2023). MemoryBank maintains evolving user profiles with time-dependent forgetting (Zhong et al., 2024), whereas MemGPT treats the context window as working memory and moves information across virtual tiers (Packer et al., 2023). Later systems introduce more structured organization: HiAgent compresses completed subgoals (Hu et al., 2025), A-MEM maintains linked atomic notes (Xu et al., 2025), and MemoryOS applies tier-specific update policies (Kang et al., 2025). These designs improve retrieval and multi-session reasoning, but commonly rely on repeated LLM inference for selection, reflection, summarization, or note construction.

Efficient and adaptive management. Efficiency-oriented methods reduce either the context presented to an agent or the cost of maintaining its store. LightMem moves expensive consolidation outside the online path and avoids repeatedly processing the full history (Fang et al., 2026). Learned managers instead optimize memory operations: AgeMem trains a unified policy over storing, retrieving, updating, summarizing, and discarding (Yu et al., 2026), while Memory-R1 learns ADD, UPDATE, DELETE, and NOOP decisions jointly with memory-aware answering (Yan et al., 2026). Although adaptive, these approaches require policy training and often retain generative models in the management loop. Moreover, efficient retrieval cannot repair poor entries after ingestion, since agents may follow retrieved experiences even when they are erroneous (Xiong et al., 2026).

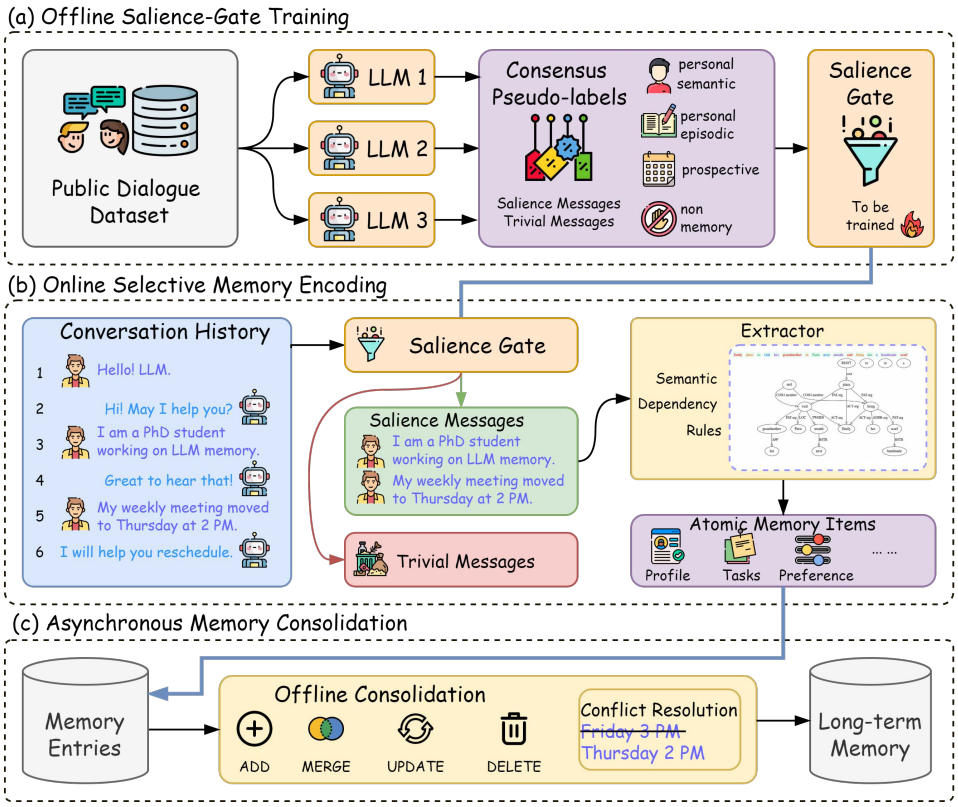

Figure 1: Overview of SLICE-Mem architecture. (a) Consensus-based salience gate training. (b) Dependency-guided extraction and turn-level memory storage. (c) Offline reconciliation of memory entries.

Salience-aware curation and lightweight extraction. Selective memory is usually implemented through scalar importance scores, heuristic retention rules, or prompt-based LLM judgments. Generative Agents combine importance with recency and relevance at retrieval time (Park et al., 2023), while MemoryBank couples importance with forgetting (Zhong et al., 2024). Such mechanisms may conflate different reasons for remembering, including future utility, persistent personal relevance, and social significance. Structured systems also commonly ask an LLM to convert interactions into summaries, profiles, or atomic notes before pruning or consolidation (Kang et al., 2025; Xu et al., 2025). In contrast, SLICE-Mem makes an explicit admission decision before extraction and uses deterministic linguistic processing for admitted turns, removing LLM calls from online memory construction. Its contribution is therefore complementary to hierarchical storage and learned lifecycle management. Related structured, temporal, linguistic, and admission-control designs include EnSIMem, TSM, Semantic Anchoring, and A-MAC (Meng et al., 2026; Su et al., 2026; Chatterjee & Agarwal, 2025; Zhang et al., 2026a); Appendix D.4 contrasts their mechanisms with ours.

3 METHOD

As illustrated in Figure 1, SLICE-Mem separates online memory formation from offline consolidation. Given a dialogue $U = (u_1, \ldots, u_T)$, with speaker, text, and timestamp for each turn, the online pipeline admits informative user turns, extracts predicate-centered statements, and inserts their embeddings with timestamp and provenance. Admission and extraction use a compact classifier and deterministic linguistic rules without generative LLM calls. Consolidation subsequently reconciles relations among stored entries without rewriting their text.

3.1 SALIENCE-GATED MEMORY ADMISSION

The Gate predicts whether retaining a user turn may support future continuity, personalization, factual consistency, or plan completion. For a target user turn u_t, let $H_t^{(k)}$ contain up to k immediately preceding dialogue messages. We construct $z_t = \mathrm{Pack}(H_t^{(k)}, u_t)$, using explicit markers to distinguish the context from the target. The Gate encodes this input and predicts admission through a single-logit classification head:

$$p_t = \sigma\big(\mathbf{w}^\top \mathrm{Enc}_\theta(z_t) + b\big), \qquad g(u_t) = \mathbf{1}[p_t \geq \tau_g]. \tag{1}$$

Here, $\mathrm{Enc}_\theta(z_t) \in \mathbb{R}^d$ is the d-dimensional pooled input representation produced by the encoder with parameters θ, and $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are the learnable classification-head parameters. The sigmoid function σ maps the logit to p_t, the estimated probability that the target turn merits retention given its context. The admission threshold τ_g determines whether the turn proceeds to extraction: the indicator function $\mathbf{1}[\cdot]$ returns 1 when the condition holds and 0 otherwise.

The Gate is trained on binary memory-admission labels obtained from consensus among multiple LLM teachers and released annotations. It learns admission directly rather than predicting a semantic memory type, using class-weighted binary cross-entropy. At inference time, preceding messages provide context for the admission decision, while only the admitted target user turn is passed to the extractor. Supervision construction and the training objective are detailed in Appendices A.1–A.2; training settings and context-window selection are reported in Appendices B.1 and C.1.

3.2 SEMANTIC DEPENDENCY-GUIDED MEMORY EXTRACTION

We use HanLP (He & Choi, 2021), an NLP toolkit, to obtain tokenization, part-of-speech tags, named entities, and semantic dependencies for each admitted user turn. Based on these analyses, deterministic rules divide the turn into clauses and select predicates with informative arguments and modifiers. For a candidate predicate p, the score is

$$S(p) = \sum_{r \in \mathcal{R}} \alpha_r I_r(p) + B(p) - P(p), \tag{2}$$

where $I_r(p)$ indicates the presence of an outgoing relation in semantic role group r, α_r is its fixed weight, and B and P encode content-cue bonuses and structural penalties. Role groups cover patients or objects, actors, time, purpose/reason/condition, and location. Appendix A.3 details selection and realization; numerical settings are collected in Appendix B.1.

For each selected predicate, the extractor follows semantic dependencies to identify its arguments and modifiers, extending the text span when needed to retain important entities and quantities. It then takes the corresponding contiguous span from the source, preserving the original word order. Deterministic rules convert first-person expressions into user-attributed statements and restore missing subjects or omitted source content where possible. These operations require no generative LLM calls. Related or overlapping units may be merged to preserve context, and units exceeding the per-turn limit are combined rather than discarded. A coverage check measures how much content from the cleaned source remains in the extracted text. If coverage is insufficient or no usable unit is produced, the extractor attempts to retain a normalized version of the cleaned source instead. This fallback reduces information loss, although some turns may still yield no entry. The extracted units are concatenated in order into $m_t = \mathrm{Concat}(f_{t,1}, \ldots, f_{t,n_t})$, where $f_{t,i}$ denotes the i-th unit extracted from turn t and n_t is the number of units. The resulting text is embedded and stored as a single memory entry with its timestamp and source-turn provenance. Only the target user turn supplies memory content; preceding dialogue informs the Gate's admission decision but is not included in the extraction input.

3.3 OFFLINE MEMORY CONSOLIDATION

After online memory construction, SLICE-Mem performs offline consolidation to identify redundant or outdated information among stored entries (Diekelmann & Born, 2010; Rasch & Born, 2013). It updates entry statuses to guide subsequent retrieval while preserving the original text, embeddings, and timestamps, without deleting or rewriting memories.

The procedure begins with deterministic rules for clear duplicates and changes in mutually exclusive personal attributes. Duplicate detection considers whether entries refer to the same occurrence, so similar descriptions of separate events remain available. State changes are processed in timestamp order, allowing a return to an earlier value to be retained as a new transition. For entries that remain active, the system retrieves similar candidates whose message timestamps are no later than the target's. A LLM then classifies the relation between the target and each candidate. The four possible decisions are `ignore`, which leaves the entries unchanged, `dedup`, which marks the later duplicate as redundant, `supersede`, which marks the older entry as outdated and links it to its replacement, and `qualify`, which records a compatible relation while retaining both entries. Before applying a decision that would exclude an entry from ordinary retrieval, the system checks the stated relation, its scope, and supporting text from the target. Unsupported decisions are ignored. Additional checks enforce consistent status changes and replacement links. The prompt also directs the model to retain complementary information and avoid replacing an entire entry when only part of its content conflicts. Subsequent retrieval normally uses active entries. Questions matching predefined temporal or state-change cues can additionally retrieve outdated entries and follow their replacement links, with historical information explicitly marked. Because excluding an entry may also hide useful evidence, we evaluate the online and consolidated stores separately. Appendix A.4 provides the detailed rules, candidate-selection procedure, validation checks, and historical retrieval conditions.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks and baselines. We evaluate on Memora Weekly (Uddin et al., 2026) and LongMemEval-S (Wu et al., 2025), comparing SLICE-Mem with LangMem (LangChain, 2025), A-MEM (Xu et al., 2025), Mem0 (Chhikara et al., 2025), MemOS (Li et al., 2025), and LightMem (Fang et al., 2026). The quarterly subset of Memora contributes Gate supervision, while the weekly subset is used for end-to-end evaluation. Supervision construction and split handling are described in Appendix A.1; model configurations and evaluation settings are detailed in Appendix B.1.

Experimental settings. We evaluate memory construction and consolidation using 8B and 30B Qwen backbones, while fixing answer generation to Qwen3-30B-A3B-Instruct-2507 and judging to GPT-OSS-20B. The construction backbones are used only by LLM-dependent components and their paired consolidators. SLICE-Mem Online and the variant without the Salience Gate do not use a generative construction backbone. For each benchmark, we reuse one fixed evaluation per applicable method across the 8B and 30B blocks. Consolidation and the generative ablations are evaluated separately with their respective construction backbones. The Gate uses 2000 hard-labeled examples, with 1600 for training and 400 for validation, and is independently evaluated on 150 human-annotated turns. Its context window is selected using validation loss. We evaluate a nine-point grid of admission thresholds, $\tau_g \in \{0.1, 0.2, \ldots, 0.9\}$, using the complete memory-and-QA pipeline on Memora Weekly. The setting $\tau_g = 0.2$ attains the highest overall FAMA in this sweep and is used for the main comparisons; Appendix C.3 reports all nine settings. The same Gate configuration is applied to LongMemEval-S without dataset-specific tuning. All five baseline frameworks are evaluated on the benchmark inputs. Detailed configurations, data-overlap handling, and component evaluations are provided in Appendices B–C.

Evaluation metrics. On Memora Weekly, we report Forgetting-Aware Memory Accuracy (FAMA), which evaluates remembering, reasoning, and recommending while penalizing reliance on information that should be forgotten. On LongMemEval-S, we report answer accuracy (Acc.) across questions involving temporal reasoning, multi-session evidence, knowledge updates, and information from individual speakers. We measure operational cost using LLM token consumption, construction-stage LLM calls, and reported memory-construction runtime. Token counts cover input and output tokens for summarization, updates and answer generation (QA), and call counts cover memory construction and consolidation. For consolidated systems, both online construction and offline consolidation are included in the reported costs. Metric definitions and detailed accounting procedures are provided in Appendix B.2.

4.2 Main Results

Tables 1 and 2 compare effectiveness and efficiency on Memora Weekly and LongMemEval-S under the 30B and 8B construction settings, with answer generation and judging held fixed. We report the online and consolidated states separately. All costs reported for the consolidated state include both online memory construction and offline consolidation; total LLM tokens additionally include answer generation. The online scores are from separate QA runs rather than matched reruns of the consolidated states; the comparison protocol and its limits are described in Appendix B.2.

Memora Weekly. At the $\tau_g = 0.2$ operating point selected by the nine-threshold end-to-end sweep (Appendix C.3), SLICE-Mem's separate main-table evaluation under the shared 30720-token QA context achieves 46.60 FAMA in both the 30B and 8B comparisons. This is 0.08 points below MemOS in the 30B block and 3.22 points above MemOS in the 8B block. These results are obtained without generative LLM calls for online admission or extraction. In the 30B comparison, SLICE-Mem uses 0.536M total LLM tokens and reports 0.362 h of parsing-plus-embedding time, compared with 7.009M tokens and 2.378 h for MemOS, reducing token consumption by 92.4% and the compared time values by 84.8%. Relative to LightMem, the online pipeline reduces total tokens by 79.2% and 83.4% and reported construction time by 64.5% and 76.0% in the 30B and 8B comparisons, respectively. These are comparisons of recorded implementation-stage timings, not uniformly instrumented end-to-end latency; Appendix B.2 details the accounting boundary.

After consolidation, FAMA is 46.74 with the 30B consolidator and 47.29 with the 8B consolidator. The corresponding combined construction-and-consolidation times are 0.874 and 0.819 h, with 2532 consolidation calls in each setting. Total token consumption is 3.356M and 3.340M, respectively. Both consolidated states have lower reported runtime than LightMem, but both use more total tokens than LightMem's 2.584M and 3.231M in the paired settings. Relative to the shared online score of 46.60, FAMA increases by 0.14 after 30B consolidation and by 0.69 after 8B consolidation. Thus, the strongest efficiency gains arise from the online pipeline, while consolidation adds cost and has scale-dependent effectiveness.

LongMemEval-S. The online pipeline achieves 56.8% accuracy in the 30B comparison, while it matches A-MEM and is 2.5 percentage points below MemOS, the highest-scoring baseline in Table 2 at 59.3%. In the 8B comparison, it exceeds the strongest listed 8B baseline, A-MEM at 56.2%, by 0.6 percentage points. The online result is shared across these comparisons because SLICE-Mem's online construction does not depend on a generative backbone. Its 0.913M total tokens consist entirely of answer-generation input and output, and its reported construction time is 31.679 h. Relative to MemOS in the 30B comparison, these represent reductions of 99.5% in total tokens and 61.0% in construction time. Relative to A-MEM in the 8B comparison, the reductions are 99.6% and 27.3%, respectively. The Gate, dependency parser, and embedding model still incur computation; eliminating generative construction calls therefore does not imply zero construction cost.

The consolidated state reaches 56.0% accuracy with both backbones, compared with 56.8% for the online evaluation. The 30B consolidated state uses 57.650M total tokens and 59762 construction/consolidation calls, 44.7% and 51.9% below LightMem, respectively, while achieving 5.2 percentage points higher accuracy. The 8B consolidated state uses 57.728M total tokens and requires 37.984 h for online construction and offline consolidation combined. Compared with A-MEM in the 8B setting, it uses 74.0% fewer tokens and 12.8% less reported construction time, with accuracy 0.2 percentage points lower. Substantial cost savings therefore remain possible after consolidation, although neither consolidated LongMemEval-S run exceeds the online score. Additional category-wise results are provided in Appendix D.

Across both benchmarks and the 8B and 30B backbone comparisons, SLICE-Mem's online pipeline maintains competitive task performance while substantially reducing reported memory-construction time and generative LLM cost relative to the strongest listed baseline in each setting.

4.3 Admission and Extraction Quality

Does the Gate retain useful evidence? At the deployed operating point, the Gate retains 28.1% of Memora Weekly user turns, filtering approximately 72% of candidates before parsing and em-

Table 1: Effectiveness and stage-wise cost on Memora Weekly. Summary, Update, and QA report input/output tokens in millions (M); QA denotes answer generation. Bold and underlined values indicate the best and second-best effectiveness or cost within each block. SLICE-Mem Online and w/o Salience Gate share one fixed evaluation per method across blocks. – indicates no value for the metric and + Consolidation includes the online and offline stages.

Method	FAMA ↑	Summary (M)		Update (M)		QA (M)		Total (M) ↓	Calls ↓	Time (h) ↓
		In	Out	In	Out	In	Out			
Qwen3-30B-A3B-Instruct-2507										
LangMem	24.05	5.339	0.675	6.484	1.067	1.929	0.057	15.551	3039	10.639
A-Mem	35.67	0.858	0.193	4.985	0.984	2.999	**0.048**	10.067	2732	3.777
Mem0	43.08	13.780	0.562	**0.234**	**0.010**	**0.371**	0.068	15.025	1551	1.856
MemOS	46.68	5.279	1.028	–	–	0.633	0.069	7.009	5499	2.378
LightMem	36.93	1.035	0.141	0.876	0.043	0.428	0.061	2.584	2939	1.019
SLICE-MEM	46.60	**0**	**0**	–	–	0.474	0.062	0.536	**0**	**0.362**
+ Consolidation	46.74	**0**	**0**	2.301	0.517	0.474	0.064	3.356	2532	0.874
w/o Salience Gate	43.61	**0**	**0**	–	–	0.457	0.059	**0.516**	**0**	0.761
w/o Extractor	**48.02**	0.631	0.115	–	–	0.409	0.062	1.217	1351	0.442
w/o Both	44.64	1.156	0.143	–	–	0.407	0.061	1.767	1555	0.542
Qwen3-8B										
LangMem	29.84	5.190	0.744	5.543	0.580	1.224	0.052	13.333	3108	8.830
A-Mem	35.03	0.981	0.143	5.086	0.483	2.999	**0.048**	9.740	3092	3.265
Mem0	41.49	13.680	0.585	**0.169**	**0.008**	**0.275**	0.063	14.780	1551	2.583
MemOS	43.38	4.859	0.909	–	–	0.532	0.067	6.368	5841	4.594
LightMem	37.60	1.056	0.255	1.358	0.062	0.441	0.059	3.231	3686	1.509
SLICE-MEM	46.60	**0**	**0**	–	–	0.474	0.062	0.536	**0**	0.362
+ Consolidation	**47.29**	**0**	**0**	2.311	0.492	0.474	0.064	3.340	2532	0.819
w/o Salience Gate	43.61	**0**	**0**	–	–	0.457	0.059	**0.516**	**0**	0.761
w/o Extractor	44.05	0.629	0.100	–	–	0.431	0.061	1.221	1345	**0.358**
w/o Both	46.48	1.171	0.239	–	–	0.430	0.059	1.899	1566	0.835

bedding. The shared online evaluation achieves 46.60 FAMA in both comparison blocks (Table 1). Retention alone does not establish evidence quality: the end-to-end threshold scan in Table 5 directly tests how more aggressive filtering affects downstream performance. Together, these results show that substantial filtering can preserve competitive answer quality, while the sensitivity analysis identifies the cost of discarding additional turns.

Independent validation of salience judgments. On a disjoint, human-annotated set of 150 turns, the selected Gate configuration achieves an AUPRC of 0.898 ± 0.021 over three training seeds. Human and teacher admission decisions agree on 140 examples (93.3%). The context window is selected using Gate validation loss, not the human-audit scores. This provides independent evidence for the learned admission signal, although the audit uses one annotator and stratified sampling rather than the natural stream distribution. Context-dependent turns remain harder: a representative model obtains AUPRC 0.844 on those turns versus 0.935 on self-contained turns. Training, sampling, and uncertainty estimates appear in Appendix C.

Sensitivity and extraction reliability. The earlier threshold-sweep runs reveal a coverage–cost trade-off. Their QA evaluations are separate from the shared online evaluation used in the main tables. Raising τ_g from 0.2 to 0.9 reduces retained turns from 28.1% to 13.3% and reported parsing-plus-embedding time from 0.362 to 0.173 h, but lowers overall FAMA from 48.54 to 42.10 and Reasoning FAMA from 46.67 to 32.67. Within this Memora Weekly sweep, moderate filtering yields higher FAMA than more aggressive rejection. The sweep characterizes the trade-off on the data used for threshold selection. At the deployed $\tau_g = 0.2$ operating point (3327 admitted turns), the extractor produces empty outputs on 0.51% of admitted turns and none of the positive-labeled

Table 2: Effectiveness and stage-wise cost on LongMemEval-S. Summary, Update, and QA report input/output tokens in millions (M); QA denotes answer generation. Bold and underlined values indicate the best and second-best effectiveness or cost within each block.

Method	Acc. (%) ↑	Summary (M)		Update (M)		QA (M)		Total (M) ↓	Calls ↓	Time (h) ↓
		In	Out	In	Out	In	Out			
Qwen3-30B-A3B-Instruct-2507										
LangMem	23.6	127.598	11.619	142.472	24.825	3.252	0.060	309.827	47634	300.459
A-Mem	<u>56.8</u>	58.789	4.226	136.010	6.753	12.385	0.050	218.213	35311	32.883
Mem0	13.8	262.676	<u>4.078</u>	**4.431**	**0.058**	**0.280**	**0.032**	271.557	<u>23848</u>	**13.683**
MemOS	**59.3**	141.753	32.948	–	–	0.859	0.055	175.615	89341	81.252
LightMem	50.8	<u>24.270</u>	12.028	63.460	<u>4.003</u>	<u>0.486</u>	0.048	104.295	124255	50.053
SLICE-MEM	<u>56.8</u>	**0**	**0**	–	–	0.865	0.048	**0.913**	**0**	<u>31.679</u>
+ Consolidation	56.0	**0**	**0**	<u>50.348</u>	6.389	0.865	<u>0.047</u>	<u>57.650</u>	59762	38.878
Qwen3-8B										
LangMem	16.8	121.464	9.273	139.104	15.605	2.459	<u>0.040</u>	287.945	47265	334.503
A-Mem	<u>56.2</u>	58.884	<u>2.610</u>	143.716	<u>4.320</u>	12.385	0.051	221.966	36061	43.548
Mem0	10.8	260.507	3.277	**4.137**	**0.050**	**0.222**	**0.030**	268.222	<u>23857</u>	**21.244**
MemOS	55.0	98.856	18.622	–	–	0.727	0.051	118.257	60117	112.907
LightMem	53.4	<u>24.058</u>	8.682	258.222	15.230	<u>0.565</u>	0.046	306.804	414413	103.821
SLICE-MEM	**56.8**	**0**	**0**	–	–	0.865	0.048	**0.913**	**0**	<u>31.679</u>
+ Consolidation	56.0	**0**	**0**	<u>50.587</u>	6.228	0.866	0.048	<u>57.728</u>	59762	37.984

turns. The heuristic fragment rate is 0.96%, whole-turn fallback occurs on 6.64%, and realized strings retain 82.80% of content tokens from positive turns. These automatic diagnostics characterize the admitted stream at the deployment threshold and support the feasibility of deterministic realization, but they do not establish proposition-level factuality, atomicity, or downstream retrieval utility (Appendix C.4). The following ablations provide the more direct evidence for the modules' downstream contribution.

4.4 Ablation and Consolidation Analysis

Contributions of Gate and Extractor. Here, *w/o Gate* processes all candidate user turns with the deterministic extractor; *w/o Extractor* retains the Gate but replaces deterministic extraction with LLM-based extraction; *w/o Both* omits the Gate and uses LLM-based extraction. Thus, removing the Extractor does not mean storing raw turns. In the 30B Memora block, removing both components yields 44.64 FAMA and retaining only the Gate yields 48.02. The shared non-generative evaluations obtain 43.61 with the dependency extractor alone and 46.60 with both components. In the 8B comparison, neither component alone improves on the 46.48 no-component configuration, whereas their combination reaches 46.60. Thus, the effectiveness gains from individual components are not uniform across the paired comparisons. The cost effects are clearer: the Gate-only variants still invoke generative extraction, while the extractor-only variants remove these calls but process all candidate turns. Combining both preserves the zero-call online path and reduces reported runtime from 0.761 to 0.362 h relative to the extractor-only variant. Full training, selection, and diagnostic results are in Appendix C.

Joint effectiveness and efficiency. Against the configuration without either component, the combined online pipeline has 1.96 points higher Memora FAMA in the 30B comparison while reducing reported construction time from 0.542 to 0.362 h. In the 8B comparison, FAMA changes by only 0.12 points, but runtime falls from 0.835 to 0.362 h and generative construction calls fall from 1566 to zero. The benefit is therefore not solely an increase in answer score: a similar score can accompany a substantial change in the computational path. Conversely, the 8B Gate-only ablation is slightly faster than the combined pipeline (0.358 versus 0.362 h), but has lower FAMA and still re-

quires generative extraction. These comparisons distinguish the cost of filtering turns from the cost of representing admitted content.

Task dependence of consolidation. The 8B Memora results illustrate why an aggregate improvement should be interpreted cautiously. Relative to the online evaluation, the consolidated state raises Recommending FAMA by 0.77 points and Reasoning by 2.00, while Remembering decreases by 0.71; overall FAMA rises by 0.69. With the 30B consolidator, overall FAMA increases by 0.14, with improved Remembering, unchanged Reasoning, and lower Recommending. The choice of online versus consolidated memory should therefore consider both the target task and the added cost. These are descriptive comparisons across runs, without an estimate of repeated-run variation.

Consolidation and evidence coverage. In the 30B Memora comparison, Recommending FAMA changes from 50.23 to 46.34 after consolidation, while Reasoning remains at 43.33. On LongMemEval-S, Multi-Session accuracy changes from 66.92% to 66.17% with either consolidator. Knowledge-Update remains at 74.36% with 30B but changes from 74.36% to 73.08% with 8B; Single-Assistant changes from 25.00% to 23.21% with 30B and stays at 25.00% with 8B. These descriptive differences do not isolate which relation decisions caused an answer to change. The low Single-Assistant score also highlights the user-turn-only admission boundary. Appendix D gives the detailed results.

5 Discussion and Limitations

SLICE-Mem separates the cost of forming useful memories online from the cost and risks of later reconciliation. Against the five baseline frameworks in Tables 1 and 2, the results show that selective admission and deterministic extraction can preserve competitive downstream performance with substantially fewer generative calls and tokens. They do not establish a universal runtime advantage, and the total cost of a consolidated system must include its offline stage. Component training and teacher annotation are additional preparation costs outside the reported operational comparisons.

Consolidation is not uniformly beneficial. Status-based suppression can restrict retrieval of useful historical evidence even when the original text remains stored. The Memora 30B and 8B results differ, motivating policies that preserve complementary episodes and defer uncertain relation decisions. The evaluation measures downstream effectiveness and operational cost, but does not isolate the effects of individual relation decisions or quantify changes in store compactness.

The present pipeline admits only user turns, limiting coverage of assistant-authored evidence. The one-annotator Gate audit and automatic extractor diagnostics at the deployed $\tau_g = 0.2$ do not establish proposition-level factuality or complete coreference resolution. The fixed HanLP parser and extraction rules have not been systematically stress-tested under domain shift, disfluency, or controlled parser errors. The threshold is selected from a full nine-point end-to-end sweep on Memora Weekly, also used for reporting. Exact-string de-duplication does not rule out semantic overlap. Evaluation is limited to two dialogue-memory benchmarks, without heterogeneous-source testing such as RHELM (Zhang et al., 2026b).

6 Conclusion

We presented SLICE-Mem, a lightweight memory framework that separates online memory formation from offline consolidation. Learned salience-based admission and deterministic dependency-guided extraction eliminate generative LLM calls from online admission and extraction while preserving useful conversational evidence. Experiments on Memora Weekly and LongMemEval-S with 8B and 30B Qwen backbones for memory construction and consolidation demonstrate a favorable effectiveness–cost trade-off. Against MemOS in the 30B comparison, the online pipeline achieves 46.60 versus 46.68 FAMA on Memora and 56.8% versus 59.3% accuracy on LongMemEval-S, with 92.4% and 99.5% fewer total LLM tokens, respectively. Reported Memora construction runtime decreases by 84.8%. The consolidated state has task- and backbone-dependent effects on Memora and scores below the online state on LongMemEval-S. These findings support selective, non-generative memory formation as an efficient foundation for persistent-memory agents. Future work will focus on evidence-preserving consolidation and extending coverage to assistant-authored information.

AI USE STATEMENT

Generative AI tools were used to edit the manuscript for grammar and readability and to assist with code generation. The authors manually reviewed and verified the AI-assisted code for correctness. The authors reviewed all AI-assisted wording and take full responsibility for the final implementation and manuscript, including its claims and results.

REPRODUCIBILITY STATEMENT

The Method and Experiments sections describe the SLICE-Mem pipeline, benchmarks, baselines, evaluation metrics, and main comparisons. Appendices A–D provide the annotation and extraction rules, model and hardware settings, evaluation and cost-accounting protocols, parameter-selection analyses, and detailed results. The complete code project has been uploaded as supplementary material to enable readers to reproduce the reported experiments.

REFERENCES

Richard C. Atkinson and Richard M. Shiffrin. Human memory: A proposed system and its control processes. In Kenneth W. Spence and Janet T. Spence (eds.), *Psychology of Learning and Motivation*, volume 2, pp. 89–195. Academic Press, 1968. doi: 10.1016/S0079-7421(08)60422-3.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, 2024. doi: 10.18653/v1/2024.acl-long.172.

Bill Byrne, Karthik Krishnamoorthi, Saravanan Ganesh, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. Taskmaster-2. Google Research dataset, 2020. URL https://github.com/google-research-datasets/Taskmaster/tree/master/TM-2-2020.

Maitreyi Chatterjee and Devansh Agarwal. Semantic anchoring in agentic memory: Leveraging linguistic structures for persistent conversational context. *arXiv preprint arXiv:2508.12630*, 2025. URL https://arxiv.org/abs/2508.12630.

Yi-Pei Chen, An-Zi Yen, Hen-Hsen Huang, Hideki Nakayama, and Hsin-Hsi Chen. LED: A dataset for life event extraction from dialogs. In *Findings of the Association for Computational Linguistics: EACL 2023*, pp. 384–398, 2023. doi: 10.18653/v1/2023.findings-eacl.29. URL https://aclanthology.org/2023.findings-eacl.29.

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.

Susanne Diekelmann and Jan Born. The memory function of sleep. *Nature Reviews Neuroscience*, 11(2):114–126, 2010. doi: 10.1038/nrn2762.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. LightMem: Lightweight and efficient memory-augmented generation. In *International Conference on Learning Representations*, 2026.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Han He and Jinho D. Choi. The stem cell hypothesis: Dilemma behind multi-task learning with transformer encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5555–5577, 2021. doi: 10.18653/v1/2021.emnlp-main.451.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. HiAgent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32779–32798, 2025. doi: 10.18653/v1/2025.acl-long.1575.

Jihyoung Jang, Minseong Boo, and Hyounghun Kim. Conversation chronicles: Towards diverse temporal and relational dynamics in multi-session conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13584–13606, 2023. doi: 10.18653/v1/2023.emnlp-main.838.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25961–25970, 2025. doi: 10.18653/v1/2025.emnlp-main.1318.

LangChain. LangMem. Software repository, 2025. URL https://github.com/langchain-ai/langmem.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

Zhiyu Li, Shichao Song, Hanyu Wang, Simin Niu, Ding Chen, Jiawei Yang, Chenyang Xi, Huayi Lai, Jihao Zhao, Yezhaohui Wang, Junpeng Ren, Zehao Lin, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhiqiang Yin, Qingchen Yu, Bo Tang, Hongkang Yang, Zhi-Qin John Xu, and Feiyu Xiong. MemOS: An operating system for memory-augmented generation (MAG) in large language models. *arXiv preprint arXiv:2505.22101*, 2025.

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. doi: 10.1162/tacl_a_00638.

Siyang Liu, Chujie Zheng, Orianna Demasi, Sahand Sabour, Yu Li, Zhou Yu, Yong Jiang, and Minlie Huang. Towards emotional support dialog systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3469–3483, 2021. doi: 10.18653/v1/2021.acl-long.269.

Xuanyu Meng, Xing Fan, Xinyi Fan, Chenlei Guo, Yixuan Xie, and Jiawei Han. EnSIMem: Entity-structured indexing for long-term agent memory. *arXiv preprint arXiv:2609.27279*, 2026. URL https://arxiv.org/abs/2609.27279.

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

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023. doi: 10.1145/3586183.3606763.

Björn Rasch and Jan Born. About sleep's role in memory. *Physiological Reviews*, 93(2):681–766, 2013. doi: 10.1152/physrev.00032.2012.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, 2024.

Miao Su, Yucan Guo, Zhongni Hou, Long Bai, Zixuan Li, Yufei Zhang, Guojun Yin, Wei Lin, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. Beyond dialogue time: Temporal semantic memory for personalized LLM agents. *arXiv preprint arXiv:2601.07468*, 2026. URL https://arxiv.org/abs/2601.07468.

Md Nayem Uddin, Kumar Shubham, Eduardo Blanco, Chitta Baral, and Gengyu Wang. From recall to forgetting: Benchmarking long-term memory for personalized agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 26814–26841, 2026. URL https://aclanthology.org/2026.findings-acl.1337/.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2526–2547, 2025. doi: 10.18653/v1/2025.acl-long.127.

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

Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Zirui Liu, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. How memory management impacts LLM agents: An empirical study of experience-following behavior. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 623–645, 2026. doi: 10.18653/v1/2026.acl-long.27.

Jing Xu, Arthur Szlam, and Jason Weston. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5180–5197, 2022. doi: 10.18653/v1/2022.acl-long.356.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schuetze, Volker Tresp, and Yunpu Ma. Memory-R1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825, 2026. doi: 10.18653/v1/2026.acl-long.583.

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21457–21483, 2026. doi: 10.18653/v1/2026.acl-long.981.

Guilin Zhang, Wei Jiang, Xiejiashan Wang, Aisha Behr, Kai Zhao, Jeffrey Friedman, Xu Chu, and Amine Anoun. Adaptive memory admission control for LLM agents. *arXiv preprint arXiv:2603.04549*, 2026a. URL https://arxiv.org/abs/2603.04549.

Han Zhang, Zihao Tang, Xin Yu, Xiao Liu, Yeyun Gong, Haizhen Huang, Yan Lu, Weiwei Deng, Feng Sun, Qi Zhang, and Hanfang Yang. Beyond static dialogues: Benchmarking realistic, heterogeneous, and evolving long-term memory. *arXiv preprint arXiv:2605.31086*, 2026b. URL https://arxiv.org/abs/2605.31086.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024. doi: 10.1609/aaai.v38i17.29946.

A Methodology Details

A.1 Salience Annotation and Supervision Construction

Memory admission. We distinguish a turn's semantic type from its value for persistent memory. The four auxiliary types are personal semantic information (identity, preferences, habits, and enduring constraints), prospective information (plans, intentions, deadlines, and commitments), autobiographical episodic information (bounded personal events), and non-memory information. The binary admission label indicates whether retaining the target turn's contribution may improve later continuity, personalization, factual consistency, or plan completion beyond the immediate exchange. A turn may mention personal information without contributing useful persistent evidence, whereas a short context-dependent reply may revise a preference or plan. Admission therefore depends on the target turn's potential future utility rather than its semantic category alone. Under the annotation protocol, however, the `non_memory` type requires a negative admission label. Auxiliary semantic types support corpus balancing and annotation checks; the student learns only binary admission.

Consensus supervision. Each of three teachers receives the same rubric and the target turn together with up to K_a preceding messages. The request separates `[CONTEXT]` from `[TARGET]` and includes anonymous speaker identifiers. Context is used only to resolve references, ellipsis, alternatives, contrasts, and relative time in the target; facts appearing solely in context are not attributed to the target. The annotation window K_a and the student window k serve different purposes and need not be equal. Each response contains a primary memory type, context-dependence label, binary admission decision, confidence, one to three verbatim target excerpts, and a short rationale. Local validation checks field types, admissible labels, confidence in $[0, 1]$, consistency of `non_memory` with rejection, and target-grounded evidence. Case-only evidence mismatches are normalized before validation. Invalid responses can receive a correction request containing validation feedback and the target text. An example enters the consensus pool only if all three teachers return valid outputs and agree on binary admission; agreement on auxiliary fields is not required. Released Memora labels are used directly, without teacher relabeling. The actual teacher rubric is Chinese; Appendix A.5 presents the system prefix and an English translation of the rubric.

Gate supervision data. We construct the Gate supervision corpus from two sources of hard labels, totaling 2000 examples with 1600 for training and 400 for validation. The first source contains 1500 examples from five conversational datasets with complementary domains: Multi-Session Chat (MSC) (Xu et al., 2022), ESConv (Liu et al., 2021), Conversation Chronicles (Jang et al., 2023), Life Event Dialog (LED) (Chen et al., 2023), and Taskmaster-2 (Byrne et al., 2020). Each dataset contributes 300 examples, split into 240 training and 60 validation examples. Candidate user turns are independently annotated by DeepSeek-V4-Pro, GLM-5.2, and Kimi-K2.6 using the protocol above. The second source contains 500 hard-labeled examples derived directly from the released `share_memory` annotations in the quarterly subset of Memora (Uddin et al., 2026). We sample 50 examples for each of ten personas, with 40 for training and 10 for validation, and keep each conversation group within a single split. User turns of at most 60 characters that exactly match a user utterance in the weekly or monthly subsets are discarded and replaced from the remaining pool to reduce the over-representation of shared boilerplate. The overlap filter targets short exact matches across Memora subsets; it does not constitute a complete audit of semantic overlap. The quarterly subset is used for supervision and the weekly subset for downstream evaluation.

A.2 Gate Input and Training Objective

The student uses the marked context–target format in Equation 1, with up to k immediately preceding messages and no future messages. Speakers are mapped to anonymous identifiers. Let $y_i \in \{0, 1\}$ denote a training label from either unanimous teacher supervision or released Memora annotations, and let p_i denote its predicted admission probability. The objective and positive-class weight are

$$\mathcal{L}_{\text{gate}} = -\frac{1}{N}\sum_{i=1}^{N}\left[w_+ y_i \log p_i + (1 - y_i)\log(1 - p_i)\right], \qquad w_+ = \text{clip}\left(\frac{N_-}{N_+}, 1, w_{\max}\right), \quad (3)$$

where N is the training-set size and N_+ and N_- are its positive and negative counts. The bounded weight compensates for class imbalance. The thresholded deployment rule is given in Equation 1; training configuration and validation-based window selection appear in Appendices B.1 and C.1.

A.3 Dependency-Based Extraction Rules

The extractor applies deterministic selection and realization to HanLP tokenization, part-of-speech tags, named entities, and semantic dependencies.

Predicate selection. Each role group contributes at most once to Equation 2. Bonuses favor content cues, identifiable actors, and first-person actors. Penalties apply to nominally embedded predicates, objectless light verbs, predicates without informative outgoing relations, and selected matrix predicates with nested actions. Candidates are processed by decreasing score, then decreasing span length, then increasing predicate-token index. Hard filters additionally remove restrictive or nominal-modifier candidates, objectless light verbs, candidates lacking informative arguments, and negative-scoring candidates. Integrity checks reject very short realizations, incomplete copular fragments without content cues, discourse-only fragments, and realizations omitting a dollar amount present in the clause. Up to K_c propositions are selected per clause. Overlapping candidates can expand an existing span. A first-person attitude predicate from *want, like, love, prefer*, or *hate* can replace an already selected nested action, preserving the attitude–action relation. For example, *I want to learn Japanese* becomes *User wants to learn Japanese*, retaining the intention rather than asserting that the user already learns the language. Appendix B.1 gives all selection weights and limits.

Span construction and realization. Permitted semantic relations guide argument, nested-clause, and noun-phrase expansion. Spatial, manner, accompaniment, and related modifiers provide additional attachments. Named entities, proper nouns, numbers, monetary patterns, and selected disconnected content tokens can widen the span. Realization uses the contiguous interval from the minimum to maximum retained token index, including intervening tokens. Token order is preserved within each span; separate units selected from the same clause may follow score order. First-person expressions are normalized to user-attributed statements. Subject detection distinguishes complete subjects from possessives, incomplete verb phrases, and object-control constructions; a default subject is supplied when needed. Requests can be rendered as *User asks to…*, and otherwise unreconstructable fragments as *User noted:…*. Recovery can append missing entities, quoted material, or monetary expressions. These operations use fixed surface rules and make no generative LLM calls.

Merging, coverage, and storage. Clauses are processed in source order. If multiple units coexist with an anaphoric or contrast cue, they are merged into one. Exceeding the turn limit K_t also merges all accumulated units rather than retaining only the highest-scoring K_t. Otherwise, up to K_t units remain. The adapter joins returned units in extraction order and stores one embedded entry per source turn. Coverage is measured against the source surface after discourse-prefix removal. A token occurrence counts as retained when its lowercase form appears as a substring of the joined lowercase realization. Repeated occurrences count separately; matching uses neither lemmatization nor word boundaries. If the retained fraction is below ρ and the cleaned surface has at least $L_{\min}$ whitespace-separated words, a normalized realization of that surface replaces the extraction. The same surface fallback is attempted when no usable clause unit is produced. Empty or recognized discourse-only inputs can still yield no memory, and the production adapter returns no units after an extraction exception. The safeguard therefore reduces content loss without guaranteeing an entry for every admitted turn. No generative fallback is used.

A.4 Offline Consolidation Procedure

Consolidation retains entry text and vectors while changing status and relation metadata. The active, outdated, and redundant states correspond to the implementation labels `active`, `superseded`, and `dropped`; the last label does not imply physical deletion.

Deterministic alignment. Exact-text comparison strips an initial date prefix, applies Unicode NFKC normalization and lowercasing, collapses whitespace, and trims boundary punctuation. Equal

normalized bodies are grouped by shared source-unit provenance when available, or by message timestamp otherwise; event-like entries lacking source-unit provenance are not deduplicated by this rule. This avoids collapsing separate purchases, step records, or meetings merely because they use the same wording. The rule-based state pass handles single-clause residence, relationship status, and digit-valued children counts in timestamp order. A repeated current value is deduplicated, whereas a changed value supersedes its predecessor only when positive timestamps establish a strict order. Processing the sequence preserves a return to an earlier value as a new state transition. Preferences are not treated as mutually exclusive by this pass.

Candidate construction and decisions. After rule processing, each active target with an embedding retrieves up to K_u nearest-neighbor hits with timestamps no later than its own. Self matches, inactive entries, candidates below the cosine-similarity threshold, and unordered pairs already assigned to a job are removed. At most N_u remaining neighbors are retained by decreasing similarity. Thus K_u limits initial retrieval and N_u limits the filtered candidate list.

The classifier returns `ignore`, `dedup`, `supersede`, or `qualify`. Deduplication retains the earlier copy and marks the later one redundant; supersession marks the older entry outdated and records reciprocal replacement links. Qualification relates compatible entries without hiding either. A proposed `dedup` or `supersede` must also specify its basis, a concrete entity-and-event or attribute scope, and a short verbatim anchor from the target. The validator checks that the anchor occurs in the target and that the basis supports the requested action. An episodic duplicate additionally requires a same-event basis plus either shared source provenance or an explicit shared event date; supersession requires an explicit event correction or an identifiable current-state change with a strict message-time order. Unsupported hiding decisions are ignored. The prompt in Appendix A.6 distinguishes independent episodes from changes to the same state or event and instructs the model to ignore uncertain or partial conflicts.

Candidate jobs are collected before classification in the store's returned entry order; they are not explicitly sorted chronologically or rebuilt after status changes. Action-specific checks skip redundant entries and inapplicable supersessions. Unparseable JSON yields no relation decisions for that request, and the failure is counted; this occurred twice in the 8B LongMemEval-S update. Unknown or forbidden rewriting actions map to `ignore`, IDs outside the candidate list are skipped, and aliases such as `duplicate` are normalized to `dedup`. Model-service failures remain errors rather than silently invoking rule-only consolidation. Status writes are checked for cycles, unintended reactivation, and inconsistent replacement links, then validated against the final stored state. The output contains relation metadata, never rewritten memory text.

Historical retrieval. A case-insensitive lexical pattern recognizes temporal or state-change queries, including expressions such as *when*, *previously*, *used to*, *how long*, *currently*, and *changed*. Purely aggregative queries do not trigger history expansion unless they also match this pattern. For requested retrieval budget R, the routine fetches up to $\max(3R, R + 16)$ similarity-ranked hits, always excludes redundant entries, and excludes outdated entries unless the trigger matches. It retains at most R eligible hits in similarity order; historical-query packing does not explicitly prioritize active entries. Triggered queries additionally follow direct supersession links from the initially packed entries, with at most $\max(R, 8)$ linked-entry lookup attempts. Expansion is one hop rather than recursive, and included IDs are not duplicated. Historical text is marked `[OUTDATED; superseded]`. The routine returns at most $R + \max(R, 8)$ entries for triggered queries and at most R otherwise.

A.5 Teacher Annotation Prompt

Task: You are an independent annotation teacher. Follow the rubric below and return JSON only.

Conversational Memory Annotation Guideline:

Decide what core proposition the TARGET contributes for a general conversational memory system. CONTEXT may be used only to resolve deixis, ellipsis, contrasts, options, or relative time in TARGET. Information that appears only in CONTEXT is not part of TARGET's contribution. Do not rely on dataset labels, test-set idiosyncrasies, class quotas, sentence length, or keywords, and do not make unsupported inferences.

Make three judgments separately: 1. primary_memory_type: what type of proposition TARGET contributes; 2. context_dependency: whether the same proposition is recoverable from TARGET alone; 3. candidate_memory: whether storing the proposition may help later turns or sessions.

Matching a memory type does NOT automatically imply candidate_memory=true. The student gate is trained ONLY on candidate_memory.

Decision procedure:

1. Recover TARGET's proposition

Identify what TARGET asserts, confirms, corrects, selects, commits to, or updates. Judge TARGET's contribution, not the whole dialogue topic. If TARGET's main act is asking, thanking, greeting, acknowledging, or reacting emotionally, treat attached clauses as secondary by default unless they clearly assert a storable fact, preference, plan, or commitment.

2. Choose exactly one 'primary_memory_type'

Only these four types are allowed:

- personal_semantic: stable or generalizable knowledge about the TARGET speaker (identity, relationships, preferences, habits, abilities, values, long-term constraints/states). Usually not tied to a single occasion. - autobiographical_episodic: a bounded personal event/experience at a specific occasion, situation, time, or place. Describes what happened once; the event need not be worth storing. - prospective: future intentions, plans, goals, commitments, appointments, deadlines, or expected actions attributable to the TARGET speaker. Corrections of future plans remain prospective. Shared future arrangements the speaker commits to also count as prospective when the main update is a future commitment. - non_memory: TARGET contributes no storable personal proposition (greetings, thanks, acknowledgments, generic questions, small talk, pure restatement, unsupported guesses, rumor-like chatty evaluations aimed at questioning/chitchat, or only confirming/repeating already-settled CONTEXT content without adding durable personal facts or future commitments).

Operational split: stable person knowledge → personal_semantic; one bounded event → autobiographical_episodic; speaker's future action/commitment → prospective; else → non_memory.

If multiple types seem to apply, pick by primary future value with priority: 1) speaker's future plan/commitment (prospective); 2) else stable person knowledge (personal_semantic); 3) else bounded event (autobiographical_episodic). Do not upgrade a one-off experience to personal_semantic unless TARGET itself states a generalizable preference/habit/value/long-term state. Do not treat casual interest, rumors, or "I heard X is good" chitchat as stable preferences.

3. "Remove CONTEXT" test

- self_contained: TARGET alone yields the same core proposition and the people/events/options/time needed to store it correctly. - context_dependent: without CONTEXT, the core proposition is incomplete, ambiguous, or clearly changes. If correct storage needs CONTEXT to resolve pronouns, ellipsis, which plan/order/option is meant, or who "they/you/it" refers to, label context_dependent. Pronouns/ellipsis/short replies are strong cues but not automatic. If CONTEXT is (none), do not invent missing meaning.

4. Decide 'candidate_memory'

Set true only if storing TARGET's contribution may improve later continuity, personalization, factual consistency, relevant follow-ups, or completion of unfinished plans/commitments, AND that benefit goes beyond the next few minutes of the current exchange.

Set false for: - small talk, thanks, greetings, acknowledgments, and generic questions; - question-dominated turns even if an interest topic is mentioned; - rumor/shallow evaluations that only sustain chitchat; - actions that only serve the next few minutes and then expire (e.g., pickup/delivery for this order, "I'm almost there", momentary commands); - trivial one-off events and experiences with no lasting value; - vague events whose key content is underspecified; - facts that appear only in CONTEXT; - interpretations that require unsupported inference.

Travel/work/daily anecdotes default to false unless TARGET already frames them as durable person knowledge or an unfinished commitment.

5. Calibration examples

- (none) / I usually prefer quiet hotels. → personal_semantic, self_contained, true - Are you still moving on Friday? / No, next Monday. → prospective, context_dependent, true - (none) / I had

toast this morning. → autobiographical_episodic, self_contained, false - For future trip plans, shall we use metric units? / Yes, let's keep metric units. → prospective, context_dependent, true - Have you been to Paris? / Not yet; I went to Brittany and liked the countryside. Going with friends? → autobiographical_episodic, self_contained, false - Any new books? / I heard Kristin Hannah is very good; have you heard of her? → non_memory, self_contained, false - Your order will be ready in about 20 minutes. / Great. I'll be there to get it. → prospective, context_dependent, false - Do your parents pressure you? / I just don't want to disappoint them. They work so hard for me. → personal_semantic, context_dependent, true

6. Output

Return exactly one JSON object (no Markdown, no extra fields):

{ "primary_memory_type": "prospective", "context_dependency": "context_dependent", "candidate_memory": true, "confidence": 0.91, "evidence": ["verbatim substring copied from TARGET"], "reason": "One short TARGET-based reason, using CONTEXT only if needed." } Memory type must be one of the four names above. context_dependency must be self_contained or context_dependent. candidate_memory must be a JSON boolean. confidence is a number in [0, 1]. evidence has 1–3 non-empty substrings copied character-for-character from TARGET (including case and punctuation). reason is one short evidence-based sentence. Reason internally; output JSON only.

A.6 Consolidation Relation Prompt

You classify relations between memories of ONE user's conversations. Never rewrite, merge, or delete memory text. Output JSON only.

TARGET has a message timestamp; CANDIDATES are earlier or equal-time messages. Message time is NOT necessarily event time. Similar topic, different amount, or later timestamp alone does not imply replacement.

Actions:

ignore: unrelated, uncertain, different entities/scopes/events, or only part of a multi-fact memory conflicts.

qualify: compatible information; keep both.

dedup: clearly the SAME fact/event retold, not another occurrence. Two purchases, daily step records, meetings, or attendance lists normally coexist even with identical wording. A return to a previous state is a new change, not a duplicate.

supersede: TARGET explicitly corrects the SAME event, or changes the SAME current-state attribute of the SAME entity and scope. Budgets, goals, project fields, plans and schedules may change. Different days' spending/steps are independent events. Daily spending is not a budget. Preserve historical evidence. If hiding a whole entry loses unrelated facts, ignore.

For ignore/qualify output only id and action. For dedup/supersede also output:

basis: same_fact—same_event—event_correction—state_change

scope: a short specific entity+attribute/event identifier (include project/category/period when relevant)

anchor: a SHORT verbatim quote from TARGET supporting the identity or explicit change; do not fabricate a quote.

Use dedup only with same_fact/same_event; supersede only with event_correction/state_change. When identity or scope is ambiguous, ignore. Do not infer an event correction from a new purchase or a different day.

Schema: { "decisions": [{ "id": "candidate id", "action": "ignore" }, { "id": "candidate id", "action": "supersede", "basis": "state_change", "scope": "user weekly lunch budget", "anchor": "change my weekly lunch budget" }] }

Keep output compact; no explanations outside these fields.

B EXPERIMENTAL DETAILS

B.1 MODELS, HARDWARE, AND HYPERPARAMETERS

Gate training. We instantiate the Gate encoder with ModernBERT-base (Warner et al., 2025). Speakers are mapped to anonymous identifiers, and the k messages immediately preceding the target turn are taken from the current dialogue. Sequences are left-truncated to 512 tokens so that the target and its nearest context are retained. We set $w_{\max}$=10, giving a positive-class weight of approximately 1.40, use AdamW with a learning rate of 2×10^{-5}, weight decay 0.01, warmup ratio 0.1, batch size 4, and four-step gradient accumulation. Models are trained for at most 10 epochs, evaluated every 50 steps, and checkpointed according to validation binary cross-entropy.

Teacher requests. The annotation context contains at most $K_a = 5$ preceding messages. Requests use temperature 0, JSON-object output, disabled thinking, and a maximum generated length of 400 tokens. The request loop permits two retries after the initial attempt. The student context window is selected separately as described in Appendix C.1.

Extractor configuration. For Equation 2, the weights for patient/object, actor, temporal, purpose/reason/condition, and locative roles are 3, 2, 2, 2, and 1, respectively. A configured content cue contributes a single bonus of 3.5; an identifiable actor adds 0.5 and a first-person actor adds another 0.5. Penalties are 1.5 for a nested matrix predicate, 6 for restrictive or nominal-modifier embedding, 2 for an objectless light verb, and 4 for missing informative arguments. The associated hard filters are detailed in Appendix A.3. The per-clause and per-turn limits are $K_c = 2$ and $K_t = 2$; exceeding K_t merges the accumulated units. Nested-clause and noun-phrase traversal depths are capped at 5 and 4. Near-duplicate suppression uses an overlap threshold of 0.85 relative to the smaller token set. Coverage fallback uses $\rho = 0.55$ and $L_{\min} = 8$ whitespace-separated words in the discourse-stripped source. Coverage counts token occurrences through case-insensitive substring matching, as defined in Appendix A.3. Additional splitting at comma–conjunction boundaries applies to segments with at least 16 words.

Consolidation configuration. Initial neighbor retrieval uses $K_u = 20$, followed by a minimum cosine similarity of 0.45 and a retained-candidate limit of $N_u = 10$. A request therefore contains one target and at most ten eligible neighbors. Relation-classifier requests use temperature 0, top-p 1, seed 42, JSON-object output, and disabled thinking. The configurable generation cap defaults to 1024 output tokens and bounds the relation-decision JSON for each request; it is not a length limit on the memory store or on rewritten memory text.

End-to-end models and hardware. Published baseline numbers are not reused because their hardware, model endpoints, and accounting conventions differ from ours. All methods are rerun on the same benchmark inputs, using official implementations and method-specific configurations where available. We use Qwen3-30B-A3B-Instruct-2507 as the answer-generation model for every system and GPT-OSS-20B as the fixed judge. For LLM-dependent memory construction, we evaluate Qwen3-30B-A3B-Instruct-2507 and Qwen3-8B; offline consolidation uses the model scale paired with the corresponding construction setting. The Salience Gate and dependency extractor are unchanged across these settings. We standardize the evaluation prompt and, where architectures permit, the retrieval budget, while preserving method-specific construction logic. API calls, token usage, and construction-stage runtimes are collected from the evaluation-run records. Experiments are conducted on a server with two NVIDIA RTX 4090 GPUs and two NVIDIA A100 GPUs. The Gate window experiments use the three seeds specified in Appendix C.1; all other randomized data processing uses seed 42.

B.2 EVALUATION PROTOCOL AND COST ACCOUNTING

Metrics. On Memora, Forgetting-Aware Memory Accuracy (FAMA) combines memory retention and forgetting at the question level. For question q, let $N_{p,q}$ and $N_{f,q}$ be the numbers of memory-presence and forgetting-absence criteria, and let $C_{p,q}$ and $C_{f,q}$ count satisfied criteria. Define $\mathrm{MPA}_q = C_{p,q}/N_{p,q}$ and $\mathrm{FAA}_q = C_{f,q}/N_{f,q}$. The evaluator sets MPA to zero when there are

no presence criteria and FAA to one when there are no forgetting criteria. It computes

$$\mathrm{FAMA}_q = \max(0, \mathrm{MPA}_q - \lambda_q(1 - \mathrm{FAA}_q)), \qquad \lambda_q = \frac{N_{f,q}}{N_{p,q} + N_{f,q}}, \tag{4}$$

with $\lambda_q = 0$ when both counts are zero. We average question-level FAMA within each persona, then average the ten persona scores and multiply by 100. Task-specific FAMA uses the same procedure restricted to questions of that task. The diagnostic MPA and FAA instead pool the corresponding criteria within each persona before averaging persona scores. On LongMemEval-S, answer accuracy is the proportion of correctly answered questions. Open-ended answers are graded by GPT-OSS-20B using a fixed evaluation prompt.

We report three operational cost measures. *Calls* counts generative LLM invocations during memory construction and consolidation, excluding answer generation and judging. Token consumption is separated into Summary, Update, and QA input/output tokens. Main-table token counts are in millions (M), summed across the evaluated personas or groups; Total includes all three stages and excludes judging. Totals are calculated from unrounded counts. LightMem's construction costs include its online and offline update stages. *Runtime* reports construction-stage time recorded in the evaluation runs, with offline reconciliation added for consolidated rows. These measurements describe the recorded memory-processing stages rather than the latency of answer generation and judging. In particular, Memora SLICE-Mem Online reports parsing-plus-embedding time, whereas baseline timings follow their recorded construction workflows; these are not uniformly instrumented end-to-end wall-clock measurements. Using the same server does not eliminate implementation, batching, or scheduling effects. For LongMemEval-S, we report 31.679 h for the online pipeline but do not report a parser/embedding/I/O breakdown, so the dominant bottleneck cannot be inferred from the aggregate. Time reductions are calculated from the original unrounded timings and displayed hours are rounded to three decimals. Teacher annotation and Gate training are preparation costs outside these operational totals. QA token consumption is reported separately from construction calls and runtime; zero construction calls therefore does not imply zero answer-generation cost. Retention ratios characterize admission selectivity, while downstream effectiveness is measured by FAMA or answer accuracy. Main-table comparisons do not include independent repeated-run variance or significance tests. The three-seed Gate experiments are not repeated end-to-end evaluations; small answer-score differences should therefore be read descriptively.

C Component Evaluation and Parameter Selection

The same Gate, admission threshold, and extraction rules are used for both downstream benchmarks. We distinguish training-set supervision, validation-based model selection, independent human auditing, and operating-point characterization on Memora Weekly.

C.1 Training and Context-Window Selection

Context-window selection and Gate evaluation. Using at most five preceding messages and identical data and optimization settings, we compare $k \in \{1, 3, 5\}$ using seeds $\{20260726, 20260727, 20260728\}$. Classification reports use a fixed threshold of 0.5. At inference we score only user turns and admit a turn when $p_t \geq \tau_g$. The window is selected using mean validation loss; operating-threshold selection is reported separately in Appendix C.3. Table 3 reports results on the 400-example Gate validation set used for early stopping and checkpoint selection. The mean validation loss selects $k = 1$. The human results in Appendix C.2 are reported as an independent evaluation rather than as a selection criterion.

C.2 Independent Human Evaluation

We construct an independent human-annotated set of 150 user turns from the five conversational datasets, with 30 examples per source. Sampling is stratified by context dependence and teacher agreement. The set is disjoint from the Gate training and validation data. One annotator independently applies the same admission criteria, yielding 84 positive examples. This set is used only for component-level evaluation and is excluded from training, early stopping, context-window selection, and operating-threshold selection. We report AUPRC and fixed-threshold precision, recall, F1,

Table 3: Salience Gate performance on the validation set ($n = 400$). Results are mean $\pm$ standard deviation over three seeds at threshold 0.5.

k	Loss	F1	Precision	Recall
1	0.382±0.015	0.853±0.013	0.841±0.015	0.866±0.027
3	0.399±0.033	0.845±0.022	0.823±0.051	0.870±0.011
5	0.428±0.035	0.828±0.017	0.863±0.052	0.800±0.063

and macro-F1. Confidence intervals for window comparisons use paired bootstrap resampling of the same 150 examples (10000 resamples), with predictions averaged across seeds.

Independent human evaluation. Table 4 reports performance on the independent 150-example human-annotated set. Values are means and standard deviations over three training seeds, and the reporting threshold is fixed at 0.5. The human annotations are not used for training, early stopping, context-window selection, or operating-threshold selection. Using paired bootstrap resampling on the same 150 examples and predictions averaged across seeds, $k = 1$ improves AUPRC over $k = 3$ by 0.043, with a 95% confidence interval of $[0.007, 0.085]$. Relative to $k = 5$, the AUPRC difference is 0.026 and its confidence interval includes zero, whereas the F1 difference is 0.100 and its confidence interval excludes zero. Human and teacher admission decisions agree on 140 of 150 examples (93.3%). For a representative $k = 1$ seed, AUPRC is 0.935 on context-self-contained turns and 0.844 on context-dependent turns. The human set has 84 positive examples (56%) because it is stratified for auditing rather than sampled to match the natural class distribution. Its fixed-threshold scores therefore characterize this audit distribution and are not estimates of accuracy on the full conversational stream.

Table 4: Salience Gate performance on the independent human-annotated set ($n = 150$). Results are mean $\pm$ standard deviation over three seeds at the fixed reporting threshold 0.5.

k	AUPRC	Precision	Recall	F1	Macro-F1
1	0.898±0.021	0.820±0.043	0.746±0.050	0.780±0.026	0.763±0.028
3	0.851±0.029	0.826±0.030	0.750±0.074	0.785±0.044	0.770±0.037
5	0.876±0.006	0.833±0.041	0.595±0.062	0.692±0.030	0.703±0.012

C.3 Admission-Threshold Selection and Sensitivity

False rejections remove evidence from all subsequent processing, so the sweep includes high-recall operating points. We evaluate the complete memory-and-QA pipeline at all nine thresholds $\tau_g \in \{0.1, 0.2, \ldots, 0.9\}$ on the same Memora Weekly stream of 11860 candidate user turns. We use $\tau_g = 0.2$ in the main comparisons because it yields the highest overall FAMA in this end-to-end sweep. LongMemEval-S uses the same threshold without separate tuning. Reporting all nine settings makes the selection transparent but does not provide an independent test of the selected operating point on Memora Weekly; a separate QA run on the same benchmark does not create a held-out dataset. Table 5 reports downstream performance and computational cost for all nine settings. At $\tau_g = 0.2$, the sweep obtains 48.54 FAMA, retains 28.1% of candidate turns, and records 0.362 h of parsing-plus-embedding time. For Tables 1 and 6, we re-evaluate this selected configuration and obtain 46.60 FAMA for the online state. The $\tau_g = 0.8$ point reuses a prior full run rather than an independent rerun within the sweep. Higher thresholds reduce runtime further but lower overall FAMA; at $\tau_g = 0.9$, Reasoning FAMA drops from 46.67 to 32.67. Thus, the selected threshold prioritizes answer quality while substantially reducing the stream processed by the extractor, rather than minimizing runtime alone.

Table 5: End-to-end sensitivity of memory curation performance and computational cost to the Salience Gate threshold τ_g. Rem., Rea., and Rec. denote task-specific FAMA for remembering, reasoning, and recommending, respectively. Keep is the percentage of candidate user turns retained for memory construction.

τ_g	FAMA	Rem.	Rea.	Rec.	MPA	FAA	Keep (%)	Runtime (h)
0.1	46.81	45.28	47.33	47.82	60.14	49.47	34.6	0.432
0.2	48.54	50.27	46.67	48.67	60.70	53.73	28.1	0.362
0.3	48.07	46.72	48.67	48.82	61.44	53.52	25.0	0.312
0.4	47.30	49.05	46.67	46.19	60.31	55.77	22.8	0.296
0.5	46.99	46.73	46.00	48.23	59.34	54.40	20.9	0.263
0.6	46.89	49.87	43.33	47.46	60.22	56.34	19.3	0.247
0.7	44.87	45.02	41.33	48.26	59.54	54.41	17.6	0.225
0.8	44.13	47.94	38.00	46.44	59.68	52.47	15.8	0.195
0.9	42.10	45.20	32.67	48.42	56.00	57.96	13.3	0.173

C.4 Extractor Diagnostics

At the deployment operating point $\tau_g = 0.2$, the Gate admits 3327 Memora Weekly user turns, including 1608 with positive memory-admission labels. Using HanLP v4.2.1 with the main-pipeline extraction configuration, empty outputs occur on 0.51% of admitted turns and none of the positive turns. A heuristic fragment flag occurs on 0.96% of admitted turns (0.25% of positive turns), and 6.64% of turns use whole-turn fallback. Content-token retention on positive turns is 82.80% over 18986 source content tokens, with 1.08 intermediate units per admitted turn on average. These diagnostics characterize extractor behavior at the deployment threshold, but they are automatic proxy metrics rather than proposition-level human judgments and do not establish factuality, atomicity, cross-turn reference resolution, retrieval utility, or answer correctness.

D Additional Results and Analysis

The following breakdowns complement the aggregate comparisons in Section 4.2. Memora FAMA and LongMemEval-S accuracy use different definitions and should not be compared numerically across benchmarks.

D.1 Task-wise Results on Memora

Table 6 reports the shared online score of 46.60 FAMA in both blocks. With the 30B consolidator, overall FAMA reaches 46.74. Remembering changes from 46.24 to 50.55, Reasoning remains at 43.33, and Recommending changes from 50.23 to 46.34. With the 8B consolidator, overall FAMA reaches 47.29. Remembering changes from 46.24 to 45.53, Reasoning from 43.33 to 45.33, and Recommending from 50.23 to 51.00. These cross-run comparisons do not isolate individual consolidation decisions or estimate repeated-run variability.

D.2 Category-wise Results on LongMemEval-S

The online SLICE-Mem state obtains 43.61% on Temporal, 66.92% on Multi-Session, and 74.36% on Knowledge-Update questions. In the 30B comparison, it leads the listed baseline frameworks on Multi-Session and Knowledge-Update, while MemOS has the highest Temporal accuracy (47.37%). Single-Assistant accuracy is 25.00%, consistent with the user-turn-only admission boundary. The 30B consolidated state reaches 56.0% overall. Relative to the online results, Multi-Session changes from 66.92% to 66.17%, Temporal from 43.61% to 42.86%, and Single-Preference from 36.67% to 33.33%; Single-User and Knowledge-Update remain unchanged, while Single-Assistant changes

Table 6: Task-wise results on Memora Weekly. Rem., Rea., and Rec. denote Remembering, Reasoning, and Recommending. Scores are reported on a 0–100 scale. SLICE-Mem Online results are shared across blocks.

Method	Overall FAMA	Rem.			Rea.		Rec.		
		FAMA	MPA	FAA	FAMA	MPA	FAMA	MPA	FAA
Qwen3-30B-A3B-Instruct-2507									
LangMem	24.05	20.63	27.89	86.88	10.67	17.78	40.85	54.61	68.72
A-Mem	35.67	62.68	74.71	76.39	10.00	16.67	34.33	49.01	61.61
Mem0	43.08	51.23	60.87	66.02	23.33	36.67	54.68	82.83	35.42
MemOS	46.68	58.35	64.74	60.89	25.33	40.00	56.37	76.98	42.76
LightMem	36.93	39.82	47.50	66.67	22.67	37.78	48.30	76.10	39.16
SLICE-MEM ($\tau_g = 0.2$)	46.60	46.24	56.56	67.30	43.33	50.00	50.23	77.69	35.05
+ Consolidation	46.74	50.55	58.70	67.79	43.33	52.22	46.34	78.01	29.33
Qwen3-8B									
LangMem	29.84	34.43	40.06	72.24	10.67	17.78	44.43	53.64	65.32
A-Mem	35.03	60.47	74.10	71.76	9.33	15.56	35.30	51.65	65.21
Mem0	41.49	50.83	57.77	74.22	27.33	41.11	46.32	75.66	32.83
MemOS	43.38	51.39	57.97	61.81	27.33	41.11	51.42	76.70	37.56
LightMem	37.60	42.90	51.73	65.94	21.33	35.56	48.57	76.06	34.72
SLICE-MEM ($\tau_g = 0.2$)	46.60	46.24	56.56	67.30	43.33	50.00	50.23	77.69	35.05
+ Consolidation	47.29	45.53	56.89	68.76	45.33	53.33	51.00	79.11	34.45

Table 7: Category-wise accuracy (%) on LongMemEval-S in the 30B and 8B construction comparisons. Parentheses report benchmark category sizes. Answer generation and judging are fixed across methods. SLICE-Mem Online results are shared across blocks.

Method	Acc.	Temporal ($n = 133$)	Multi-Session ($n = 133$)	Knowledge-Update ($n = 78$)	Single-User ($n = 70$)	Single-Assistant ($n = 56$)	Single-Preference ($n = 30$)
Qwen3-30B-A3B-Instruct-2507							
LangMem	23.6	12.03	15.79	47.44	37.14	23.21	16.67
A-Mem	56.8	36.84	51.13	69.23	77.14	98.21	13.33
Mem0	13.8	6.02	8.27	16.67	15.71	44.64	3.33
MemOS	59.3	47.37	56.06	73.08	80.00	67.86	26.67
LightMem	50.8	42.86	56.39	61.54	75.71	16.07	40.00
SLICE-MEM	56.8	43.61	66.92	74.36	77.14	25.00	36.67
+ Consolidation	56.0	42.86	66.17	74.36	77.14	23.21	33.33
Qwen3-8B							
LangMem	16.8	9.77	12.78	29.49	24.29	23.21	3.33
A-Mem	56.2	36.84	50.38	67.95	74.29	98.21	16.67
Mem0	10.8	1.50	7.52	11.54	7.14	44.64	10.00
MemOS	55.0	44.36	45.11	70.51	88.57	66.07	6.67
LightMem	53.4	43.61	62.41	67.95	71.43	19.64	40.00
SLICE-MEM	56.8	43.61	66.92	74.36	77.14	25.00	36.67
+ Consolidation	56.0	43.61	66.17	73.08	75.71	25.00	33.33

from 25.00% to 23.21%. The category counts sum to 500, and their weighted scores reproduce 56.8% online and 56.0% after consolidation to rounding precision.

In the 8B comparison, the online state leads the listed baseline frameworks on Multi-Session (66.92%) and Knowledge-Update (74.36%). MemOS leads on Temporal and Single-User, A-MEM on Single-Assistant, and LightMem on Single-Preference. The 8B consolidated state reaches 56.0% overall. Relative to the online result, Temporal and Single-Assistant remain at 43.61% and 25.00%; Multi-Session changes to 66.17%, Knowledge-Update to 73.08%, Single-User to 75.71%, and Single-Preference to 33.33%. The two consolidators have the same overall accuracy here but different category-level outcomes. The reported category sizes and accuracies imply 284 correct answers online and 280 after either consolidation, a net loss of four out of 500. For 30B, the net loss is one answer each in Temporal, Multi-Session, Single-Assistant, and Single-Preference; for 8B, it is one each in Multi-Session, Knowledge-Update, Single-User, and Single-Preference. These are aggregate net changes, not counts of changed predictions or causal attributions to individual consolidation decisions.

D.3 Extended Ablation and Consolidation Analysis

The online ablations in Table 1 separate admission filtering from non-generative extraction. Relative to the extractor-only variant, combining both reduces reported construction runtime by 52.4% while retaining zero generative construction calls. The Gate-only variant still requires 1351 and 1345 construction calls in the 30B and 8B comparisons, respectively. Its 8B runtime of 0.358 h is slightly below the combined pipeline's 0.362 h, but the latter achieves higher FAMA without generative extraction calls. For the consolidated runs, measured offline-update time is approximately 0.511 h with 30B and 0.456 h with 8B. Adding the reported online formation time yields the reported 0.874- and 0.819-h combined totals. These costs accompany different task-level effects across the two scales.

Decision reliability and untested failure mechanisms. The validators in Appendix A.4 check format, references, and textual support; they do not establish the semantic correctness of relation labels. Two unparseable 8B LongMemEval-S decisions are reported there, but this is not an estimate of the semantic error rate. Entry-level suppression may hide complementary content in a multi-fact turn, and lexical historical-query triggering may miss implicit temporal or aggregation needs. These are mechanisms suggested by the design, not audited causes of the observed score changes. Establishing causality would require tracing changed answers to retrieved evidence and the corresponding relation decisions.

D.4 Positioning and Evaluation Scope

EnSIMem organizes entity–property indexes with source-evidence provenance and adaptive retrieval (Meng et al., 2026); SLICE-Mem instead targets inexpensive turn-level admission and extraction, without an equivalent entity-indexed retrieval design. TSM models event semantic time and durative states (Su et al., 2026); message timestamps, status links, and lexical historical retrieval in SLICE-Mem do not implement semantic-time normalization or durative-memory construction. Semantic Anchoring combines dependency structure, discourse information, and coreference with vector memory (Chatterjee & Agarwal, 2025); our target-turn extractor uses a narrower deterministic realization mechanism. A-MAC explicitly models admission using interpretable factors and an LLM-assisted utility assessment (Zhang et al., 2026a); explicit admission itself is therefore not unique to SLICE-Mem. Our distinction is a compact learned Gate paired with deterministic extraction, with no generative online construction calls. These are architectural contrasts, not empirical superiority claims over those systems.

RHELM combines dialogues with heterogeneous sources such as documents and emails and tests evolving, cross-source memory (Zhang et al., 2026b). Our two-benchmark evaluation does not establish robustness in that setting. Cross-source ingestion, assistant-authored evidence, and implicit state reconstruction remain outside the tested coverage.