When Splitting Beats Concentrating: Allocation as a Design Axis in LLM Agent Memory

Anonymous ACL submission

Abstract

Long-term LLM agents face a query-time allocation problem: a fixed context budget can be spent on one memory representation or divided across complementary views. We study this axis on LongMemEval-S by comparing recent-session, retrieval, summary, and split strategies under matched character budgets, including a concentration control that gives the full budget to the same BM25 chunk retriever used by the chunk half of the split. Our Dual-Track Split baseline is deliberately simple: it spends half the memory budget on BM25-selected session summaries and half on BM25-ranked raw-history chunks. Across a budget sweep, the preferred allocation changes: chunk-only concentration is higher when context is scarce, while the split becomes higher once both summaries and chunks receive enough room to contribute. Masking and span-coverage analyses suggest why: chunks carry local answer evidence, while summaries recover session-level state and chronology. These results frame query-time memory allocation as an evaluation variable: long-memory systems have budget-response curves, and single-budget rankings can obscure when the preferred policy shifts from concentration to diversification.

1 Introduction

Long-term conversational agents increasingly rely on memory to answer questions about earlier sessions, user preferences, commitments, and evolving facts (Wu et al., 2025; Maharana et al., 2024). At query time, however, the model can only consume a limited amount of serialized context. A memory system must therefore decide not only what to store, but also which memory text to inject into the prompt under a fixed query-time budget.

Prior work has largely focused on improving memory representations or retrieval mechanisms, such as recent context, retrieved chunks, session summaries, compression, or learned memory policies (Packer et al., 2023; Zhong et al., 2024; Xu et al., 2025; Fang et al., 2026; Yan et al., 2025; Zhang et al., 2026). These comparisons are often reported at one or a few fixed budgets. Yet the same budget can be spent in qualitatively different ways: it can be concentrated on one representation, such as retrieved raw-history chunks, or split across complementary representations, such as session-level summaries and retrieved chunks.

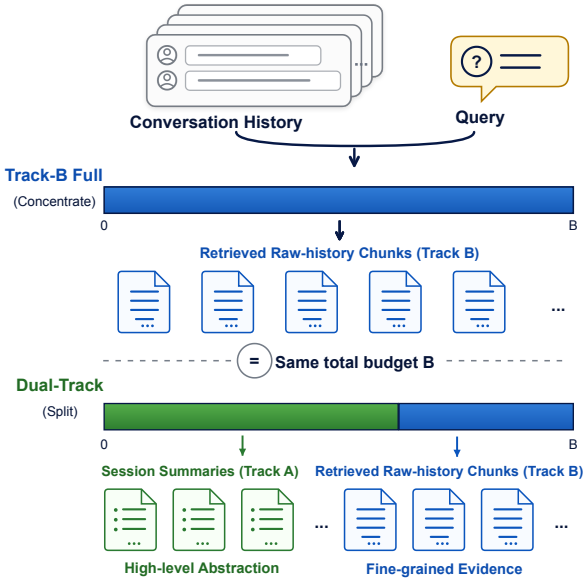

Figure 1: Query-time memory allocation under a fixed budget. BM25 Track-B Chunks concentrates the entire budget on retrieved raw-history chunks, while Dual-Track Split divides the same total budget between session summaries and retrieved chunks before using the same answer model.

Figure 1 illustrates the allocation axis we study: holding the total injected memory budget and answer model fixed, we vary whether context is concentrated in one representation or diversified across two complementary views. Concentrating the full budget on Track B emphasizes local lexical evidence from retrieved chunks. Splitting budgets between Track A and B gives the model both session-level structure and local evidence, but reduces the room available to each track. This creates a con-

crete hypothesis: splitting should hurt when the budget is too tight to feed both views, but can help once summaries and chunks each receive enough context to contribute complementary information.

Our experiments use LongMemEval-S (Wu et al., 2025) as the main testbed and sweep the query-time memory budget across character limits from 500 to 6000. Dual-Track Split has two fixed components. Track A ranks past sessions with BM25 (Robertson and Zaragoza, 2009) and inserts cached summaries of the selected sessions; Track B retrieves BM25-ranked raw-history chunks. We compare Dual-Track Split with recent-session, session-level, summary-only, dense-chunk, and sparse-chunk baselines. The sweep reveals that the ranking between concentration and splitting is not fixed. BM25 Track-B Chunks performs better when the memory budget is scarce, but Dual-Track Split overtakes it at larger budgets, where summaries and chunks can both contribute. Masking and span-coverage analyses support the same interpretation: chunks mainly provide local answer evidence, while summaries add session-level state and chronology. The implication is that long-memory methods should not be ranked at a single budget alone, instead, they should be evaluated through allocation policies and budget-response curves.

2 Related Work

Memory Systems and Budgets. Long-horizon LLM agents maintain memory of prior interactions through events, summaries, reflections, or learned management policies (Packer et al., 2023; Shinn et al., 2023; Park et al., 2023; Zhong et al., 2024; Xu et al., 2025; Fang et al., 2026; Yan et al., 2025), but they are typically evaluated as end-to-end pipelines that couple representation choice to retrieval and storage logistics. BudgetMem (Zhang et al., 2026) isolates representation under a budget, but governs construction rather than query-time injection; diagnostic work separating retrieval from utilization (Yuan et al., 2026) likewise does not allocate across representations. We hold indexing and retrieval fixed and ask how a single *query-time injection budget* should be split.

Multi-Representation Retrieval. Other lines of work obtain more from memory by reorganizing what is indexed, e.g., hierarchical summary trees or entity graphs (Sarthi et al., 2024; Edge et al., 2024), or by combining rankings from multiple queries or retrievers over the same units (Cormack et al., 2009; Rackauckas, 2024; Bruch et al., 2024). Our dual-track setup is orthogonal: both tracks share a flat index and the same BM25 retriever, and we vary only how the prompt budget is split between summary-track and chunk-track context.

3 Budgeted Memory Allocation

This section defines memory allocation as a query-time budgeting problem and specifies the matched control needed to separate allocation from retriever choice.

Allocation Protocol. Given a multi-session history $\mathcal{H}$, query q, and character budget B, a memory strategy π selects a serialized memory context $c = \pi(\mathcal{H}, q, B)$ with $|c| \leq B$ characters. The answer model then generates from (q, c). The budget applies to injected memory text after serialization, including section headers and separators, and excludes the user query and answer prompt. We use character budgets as a tokenizer-agnostic proxy for prompt length and evaluate $B \in \{500, 1000, 2000, 3000, 4000, 6000\}$ when available; Appendix A.3 reports token equivalents under the DeepSeek tokenizer.

Split Policy. Dual-Track Split is our allocation probe. It divides B equally across two views of the same history. **Track A** ranks raw sessions with BM25 (Robertson and Zaragoza, 2009) and injects cached summaries of the selected sessions in chronological order. **Track B** segments raw history into larger overlapping chunks of roughly 1024 characters with roughly 128-character overlap, indexes them with BM25 (k_1=1.5, b=0.75), retrieves up to five chunks subject to its half-budget, and presents them in conversation order. The 50/50 split is not an optimized allocation but a parameter-free diagnostic default; Appendix F checks whether the observed effect is isolated to this exact ratio.

Matched Control. The primary control is BM25 Track-B Chunks, which uses the same Track-B chunking, BM25 scoring, ordering, and serialization as the chunk side of Dual-Track Split, but receives the full budget upper bound rather than half of it. This matched control isolates the allocation choice: concentrating the query-time budget on one raw-history chunk representation versus splitting the same budget across summaries and chunks. We also include recency, session-level, summary-only, dense 64-word chunk, and BM25

Strategy	Qwen3.5-9B				DeepSeek-V4-Flash			
	1k	2k	4k	6k	1k	2k	4k	6k
Dual-Track Split (**ours**)	24.0	39.0	**48.0**	**53.0**	22.2	40.6	**48.4**	**51.4**
BM25 Track-B Chunks	**36.0**	**42.0**	47.0	49.6	**34.4**	**41.0**	43.4	42.8
BM25 64-Word Chunks	20.2	36.2	43.6	43.8	18.2	33.6	41.0	41.0
Dense 64-Word Chunks	20.4	33.8	41.0	42.6	20.0	31.6	38.6	38.6
BM25 Full Sessions	14.8	22.2	28.4	31.2	15.0	22.2	29.6	30.0
BM25 Session Summ.	13.0	16.6	22.4	27.6	13.2	18.6	26.8	30.0
Recent Full Sessions	6.2	7.0	8.4	9.4	6.2	7.2	8.4	9.0

Table 1: LongMemEval-S accuracy (%) under matched query-time budgets. Qwen3.5-9B was evaluated at 1k/2k/4k/6k. BM25 Track-B Chunks is the full-budget concentration control using Track B's chunking and BM25 retrieval.

64-word chunk baselines as reference points; Appendix A.1 lists their full definitions. The 64-word chunk baselines compare retriever choice under a conventional small-chunk setup and should not be conflated with BM25 Track-B Chunks, which is matched to Dual-Track Split for the allocation comparison. Appendix A.2 gives the prompts, serialization format, tokenization, summary caching, and reproducibility details.

4 Experiments and Analysis

4.1 Experiment Setup

The main answer and judge model is the hosted `deepseek-chat` non-thinking API, which corresponds to DeepSeek-V4-Flash under the provider's post-April-24-2026 legacy-alias mapping (DeepSeek-AI, 2026; DeepSeek, 2026). Because provider aliases are mutable and may be remapped over time, reproduction should use the released outputs or explicitly pin the evaluated API model. Using the same model family for answering and judging risks self-preference bias; we mitigate this with a blind independent GPT-4o-mini (OpenAI, 2024) judge audit that preserves the sign of all main comparisons (Appendix B.2). Generation uses temperature 0. We evaluate all methods on the LongMemEval-S evaluation split (Wu et al., 2025); cross-model and cross-benchmark checks use Qwen3.5-9B (Qwen Team, 2026) and LoCoMo (Maharana et al., 2024), with the LoCoMo results reported in Appendix G. Accuracy is the primary metric.

4.2 Main Findings

Table 1 reports the discrete budget sweep. At $B = 1000$, BM25 Track-B Chunks leads Dual-Track Split by 12.2 points (34.4% vs. 22.2%), because the split gives each track only half the budget. At $B = 2000$, the two policies are nearly tied. At

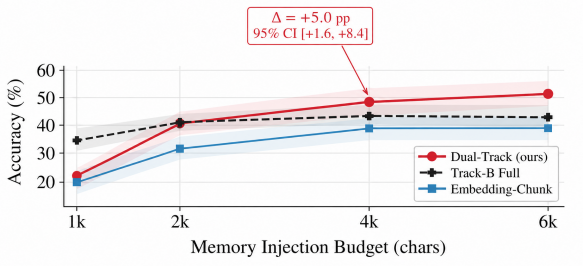

Figure 2: Accuracy-vs-budget curves on LongMemEval-S. Shaded regions are 95% bootstrap CIs for each strategy's accuracy. The split is weaker than full-budget Track-B retrieval at low budget but becomes stronger once both tracks receive a minimum viable amount of context; Figure 3 shows the paired crossover directly.

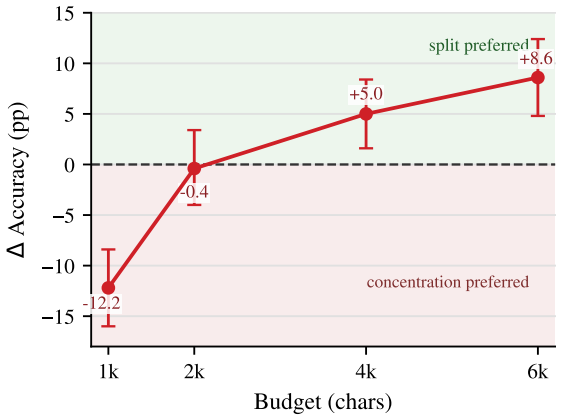

Figure 3: Paired crossover for the main matched comparison. Points show the item-level accuracy difference (Dual-Track Split minus BM25 Track-B Chunks) at each budget; error bars are 95% paired bootstrap CIs. Negative values favor concentration, while positive values favor splitting.

larger budgets, the ranking reverses: Dual-Track Split is higher by 5.0 points at $B = 4000$ and 8.6 points at $B = 6000$. The main finding is therefore not that splitting is uniformly better, but that the preferred allocation changes with budget.

Figure 2 shows the shape behind this reversal. BM25 Track-B Chunks improves quickly at first but changes little after $B = 2000$, ending at 42.8% at $B = 6000$. Dual-Track Split starts lower but continues rising to 51.4%, producing the crossover once both summaries and chunks have enough room to contribute. Figure 3 makes the same story explicit at the item level: the paired delta is negative at 1k, statistically tied at 2k, and positive at 4k/6k. The remaining checks bound alternative explanations rather than define the main allocation comparison. At $B = 4000$, Dual-Track Split is also above BM25 64-Word Chunks by 7.4 points and Dense 64-Word Chunks by 9.8 points, so the result is not just a comparison against a dense retriever.

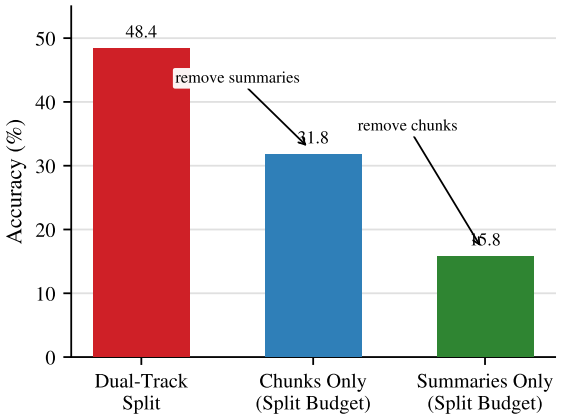

Figure 4: Masking ablation at $B = 4000$. Removing either view from the constructed Dual-Track Split prompt hurts accuracy, indicating that the split is not carried by only one representation.

These baselines use a different chunking scheme, however, so the matched allocation comparison remains Dual-Track Split versus BM25 Track-B Chunks. Appendix B.1 reports W/L/T counts and exact McNemar tests, and Appendix B.2 audits the matched comparison with an independent judge.

4.3 Analysis

This subsection interprets the crossover as a budget threshold and examines it through masking, span coverage, and split-ratio sensitivity.

Budget Threshold. Under small budgets, splitting underfeeds both views: Track A has too little room to summarize enough sessions, and Track B has too little room to retrieve enough local evidence. After each view receives enough space, the two tracks can play different roles: Track B supplies local answer evidence, while Track A restores session-level structure such as chronology, speaker preferences, and cross-session state.

Masking further supports this interpretation. At $B = 4000$, removing summaries drops accuracy from 48.4% to 31.8%, and removing chunks drops it to 15.8% (Figure 4). Chunks are therefore the stronger isolated source, but the full split outperforms either masked condition. Since the removed half-budget is not redistributed, these ablations measure contribution within Dual-Track Split, not optimal single-track allocation; Appendix C reports the task-type breakdown.

Evidence Coverage. Span coverage gives a complementary view of what each track contributes. Among the 242 items Dual-Track Split answered correctly at $B = 4000$, answer keywords are covered by both tracks in 44.2% of cases, by Track B only in 16.1%, and by Track A only in 7.4%; 15.3% are covered by neither track under a strict lexical rule. This is consistent with Track B carrying local answer spans while Track A supplies session framing that helps answer selection even without the gold answer verbatim. Appendix D reports the full table and task-type breakdown.

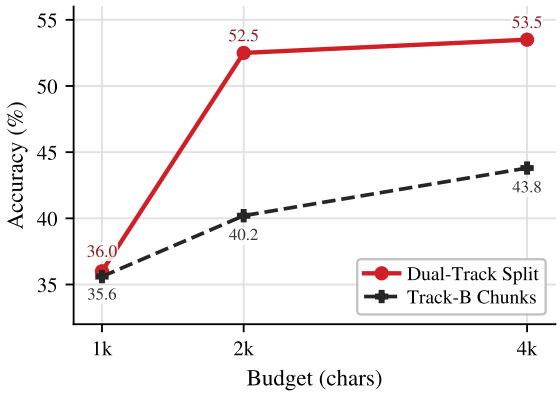

Figure 5: LoCoMo cross-benchmark comparison between Dual-Track Split and the matched Track-B concentration control.

Cross-Benchmark Check. A small LoCoMo replication suggests that the allocation effect is not specific to LongMemEval-S. As shown in Figure 5, Dual-Track Split is nearly tied with the matched Track-B concentration control at 1k (36.0 vs. 35.6), then leads by +12.3 points at 2k and +9.7 points at 4k. This suggests that the exact low-budget crossover pattern is benchmark-dependent, while split allocation can remain competitive or stronger on a second long-memory benchmark.

Ratio Sensitivity. Split-ratio robustness is reported in Appendix F; Appendix E gives additional descriptive diagnostics.

5 Conclusion

We studied query-time memory allocation for LLM agents. On LongMemEval-S, a static split is worse than full-budget Track-B retrieval at small budget, tied at medium budget, and stronger at larger budgets. This pattern suggests that the value of representation diversity depends on whether each view receives enough usable context. Memory systems should therefore treat allocation across representations as a first-order evaluation variable. A simple split is not a universal optimum, but it is a useful baseline for testing whether a benchmark rewards direct evidence, structural context, or both.

Limitations

Our evidence is strongest for English long-dialogue QA benchmarks and two model families. LoCoMo, Qwen3.5-9B, split-ratio, and judge checks are supporting rather than primary claims. We control query-time injected context rather than offline memory-construction cost; cached summaries require a separate summarization step. The 50/50 split is intentionally simple; learned allocation may perform better but introduces additional confounds. Track A assumes identifiable session boundaries, which may require extra segmentation logic in continuous interaction streams. Our full-budget concentration control is matched to Track B, while the 64-word `bm25_chunk_64w_top8` baseline instead matches `embedding_chunk`; these controls answer different questions and should not be conflated. Automatic judging may interact with the style or structure of injected memory; although the independent GPT-4o-mini audit preserves the same comparison directions, human evaluation would strengthen confidence further. Our budget sweep covers approximately 250–1500 tokens of injected memory; whether the crossover persists, widens, or reverses under high-budget regimes (e.g. 4k–20k tokens) remains an open question.

Ethical Considerations

This paper studies memory allocation on public English benchmarks and does not introduce a deployed system that stores private user conversations. Still, stronger long-context memory can increase the chance of retrieving sensitive or personally identifying details from past interactions. Real-world deployments should therefore pair memory mechanisms with data minimization, user deletion controls, access restrictions, and careful filtering of private content. Our experiments are limited to benchmark QA and should not be extrapolated to high-stakes domains such as medical, legal, or safety-critical assistance without domain-specific evaluation and human oversight.

References

Sebastian Bruch, Siyu Gai, and Amir Ingber. 2024. An analysis of fusion functions for hybrid retrieval. *ACM Trans. Inf. Syst.*, 42(1):20:1–20:35.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 758–759. ACM.

DeepSeek. 2026. DeepSeek API Docs: Change log. https://api-docs.deepseek.com/updates. Accessed 2026-05-26.

DeepSeek-AI. 2026. DeepSeek-V4-Flash. https://huggingface.co/deepseek-ai/DeepSeek-V4-Flash. Hugging Face model card. Accessed 2026-05-26.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph RAG approach to query-focused summarization. *CoRR*, abs/2404.16130.

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. 2026. LightMem: Lightweight and efficient memory-augmented generation. In *The Fourteenth International Conference on Learning Representations*.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. 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, Bangkok, Thailand. Association for Computational Linguistics.

OpenAI. 2024. GPT-4o mini: Advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. OpenAI release, July 18, 2024. Accessed 2026-05-26.

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

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *Preprint*, arXiv:2304.03442.

Qwen Team. 2026. Qwen3.5-9B. https://huggingface.co/Qwen/Qwen3.5-9B. Hugging Face model card. Accessed 2026-05-26.

Zackary Rackauckas. 2024. RAG-Fusion: a new take on retrieval-augmented generation. *CoRR*, abs/2402.03367.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on*

Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. RAPTOR: recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, pages 5776–5788. Curran Associates, Inc.

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

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

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schütze, Volker Tresp, and Yunpu Ma. 2025. Memory-R1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *Preprint*, arXiv:2508.19828.

Boqin Yuan, Yue Su, and Kun Yao. 2026. Diagnosing retrieval vs. utilization bottlenecks in llm agent memory. *Preprint*, arXiv:2603.02473.

Haozhen Zhang, Haodong Yue, Tao Feng, Quanyu Long, Jianzhu Bao, Bowen Jin, Weizhi Zhang, Xiao Li, Jiaxuan You, Chengwei Qin, and Wenya Wang. 2026. Learning query-aware budget-tier routing for runtime agent memory. In *Proceedings of the 43rd International Conference on Machine Learning*. Accepted at ICML 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. MemoryBank: Enhancing large language models with long-term memory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19724–19731.

A Method Details

A.1 Strategy Definitions

Strategy	Description
Recency and session-level baselines	
Recent Full Sessions	Most recent raw sessions
BM25 Full Sessions	BM25 over raw sessions
BM25 Session Summaries	BM25-selected cached session summaries
64-word chunk baselines	
Dense 64-Word Chunks	`all-MiniLM-L6-v2` over 64-word chunks (Wang et al., 2020; Reimers and Gurevych, 2019)
BM25 64-Word Chunks	BM25 over the same 64-word chunks; capped at top-8
Matched allocation comparison	
BM25 Track-B Chunks	Full-budget BM25 over Track-B chunks
Dual-Track Split	Session-summary track + BM25 Track-B chunk track, 50/50 split

Table 2: Full definitions of the budget-constrained memory strategies, grouped by retrieval unit and experimental role.

A.2 Implementation and Baseline Details

This appendix records implementation choices that are abbreviated in section 3: budget accounting, baseline construction, prompt templates, serialization, BM25 tokenization, summary caching, and released artifacts.

Budgeting and Ordering. Each selector returns memory text capped at B serialized characters, excluding the answer prompt and query. Selected chunks or summaries are inserted chronologically.

Baseline Definitions. Recent Full Sessions (`recent_sessions`) uses the most recent raw sessions. BM25 Session Summaries (`session_summary`) ranks raw sessions with BM25 and inserts their cached summaries. BM25 Full Sessions (`bm25_session`) ranks raw sessions directly. Dense 64-Word Chunks (`embedding_chunk`) uses 64-word chunks with `all-MiniLM-L6-v2`; BM25 64-Word Chunks (`bm25_chunk_64w_top8`) uses the same chunks with BM25 and a top-8 cap. BM25 Track-B Chunks (`bm25_trackb_full`) is the full-budget Track-B-only control: it uses Track B's approximately 1024-character chunks and adds ranked BM25 chunks up to the same budget upper bound B. Dual-Track Split (`dual_track`) splits B evenly between summaries of BM25-ranked raw sessions and BM25-ranked Track-B chunks.

Retrieval and Summarization Details. Dense 64-Word Chunks and BM25 64-Word Chunks use 64-word chunks, 16-word overlap, and 48-word stride. The dense retriever is `all-MiniLM-L6-v2` (384 dimensions, maximum sequence length 256) (Wang et al., 2020; Reimers and Gurevych, 2019). Track B uses approximately 1024-character chunks with about 128-character overlap. BM25 uses k_1=1.5 and b=0.75. Each completed session has one cached summary, generated offline by the corresponding answer model with temperature 0.1 and a 300-token cap.

Summary Prompt. Session summaries are generated with the following fixed prompt:

> You will receive a conversation session. Generate a memory summary for long-term retrieval.
>
> Requirements:
>
> 1. Preserve specific facts, dates, time cues, preferences, commitments, plans, and final states.
> 2. If old information was overridden by new information, write only the latest version explicitly.
> 3. Be specific and concrete. Avoid vague generalities.
> 4. Output 4–6 bullet points, each specific enough to directly support question answering.
>
> Session content: `{session_text}`

Serialization Format. Track A wraps selected session summaries as:

```
[Episodic Memory]
[Session 1] <bullet summary>
[Session 2] <bullet summary>
[/Episodic Memory]
```

Track B wraps retrieved chunks as:

```
[Semantic Memory]
- (turn=3) <raw dialogue text>
- (turn=7) <raw dialogue text>
[/Semantic Memory]
```

Both blocks are inserted chronologically; for Dual-Track Split they are concatenated with a blank line separator.

BM25 Tokenization. All BM25 indexes use the same tokenizer: lowercase, extract alphanumeric sequences of length $\geq$2 via the regex `[a-z0-9'_]+`. No stemming or stopword removal is applied.

Summary Caching. Summaries are generated once per completed session using the summary prompt above, cached to disk, and reused across all subsequent queries and budget levels. The summarization call uses temperature 0.1 and a 300-token output cap. All strategies share the same cached summaries.

Answer Prompt. All strategies use the following answer prompt (abstention questions receive a stricter variant instructing the model to say "I don't know" if unsupported):

> You have access to the following memory context. Answer the question based on this information. If the memory context does not contain sufficient information to answer the question, respond with "I don't know" or "Not enough information."
>
> `{memory_context}`
>
> Question: `{question}`
>
> Provide a concise answer directly. Do not explain your reasoning.

Judge Prompt. The binary semantic-match judge uses:

> Determine whether the model's answer matches the reference answer (semantic equivalence is sufficient; exact match is not required).
>
> Reference answer: `{gold}`
> Model answer: `{prediction}`
>
> Reply with only "YES" or "NO".

An abstention variant checks whether the model correctly declined to answer. Following LLM-as-judge practice (Zheng et al., 2023), the judge receives no strategy labels.

Reproducibility. Per-item predictions and judge decisions for all reported conditions are included in the supplementary release.

A.3 Character Budget and Token Cost

Table 3 reports the average input tokens (prompt tokens reported by the API) for the main comparison strategies. Because summaries are more token-efficient than raw dialogue, Dual-Track Split uses approximately 15–18% fewer tokens than BM25 Track-B Chunks at the same character budget. This means the concentration control receives *more* token-level information, so the token mismatch does not explain Dual-Track Split's accuracy advantage at larger budgets.

Strategy	1k	2k	4k	6k
Dual-Track Split	237	420	861	1207
BM25 Track-B Chunks	337	567	1025	1480
BM25 64-Word Chunks	—	—	751	—
Dense 64-Word Chunks	—	—	774	—

Table 3: Average input tokens (DeepSeek tokenizer) per strategy and character budget on LongMemEval-S (n=500). The split uses fewer tokens because summaries compress dialogue; the concentration control therefore receives more token-level information at the same budget.

B Statistical Validation of the Main Comparison

B.1 Paired Uncertainty for the Main Crossover

Budget	Δ pp	95% CI	W/L/T	p
1k	-12.2	$[-16.0,-8.4]$	20/81/399	$< 10^{-9}$
2k	-0.4	$[-4.0,+3.4]$	44/46/410	0.916
4k	$+5.0$	$[+1.6,+8.4]$	51/26/423	0.0059
6k	$+8.6$	$[+4.8,+12.4]$	71/28/401	< 0.001

Table 4: Paired uncertainty for the main crossover comparison, Dual-Track Split minus BM25 Track-B Chunks ($n = 500$). W/L/T counts items where Dual-Track Split wins, loses, or ties the concentration control. Intervals are 95% paired bootstrap CIs over item-level correctness differences; p is a two-sided exact McNemar test.

B.2 Independent Judge Validation

We audited the matched concentration comparison with an independent GPT-4o-mini judge. The audit samples the same 100 item ids for both strategies at each budget, using approximate proportional stratification by LongMemEval task type. Answers are generated first; the judge receives only the gold answer and prediction, without strategy labels.

Budget	Dual acc. (%)	Track-B acc. (%)	Δ pts	W/L/T	DeepSeek Δ pts
1000	17.0	27.0	-10.0	2/12/86	-11.0
4000	40.0	38.0	$+2.0$	7/5/88	$+4.0$
6000	42.0	39.0	$+3.0$	9/6/85	$+5.0$

Table 5: Independent judge validation for the matched comparison between Dual-Track Split and BM25 Track-B Chunks (100 stratified items per budget). Accuracies and deltas are percentages; Δ is Dual-Track Split minus BM25 Track-B Chunks. W/L/T counts items where Dual-Track Split alone is correct, BM25 Track-B Chunks alone is correct, or both agree under the GPT-4o-mini judge. The last column reports the same sampled items under the original DeepSeek judge.

C Masking Ablation Details

Table 6 expands the masking ablation summarized in Section 4.3 by task type. The purpose is diagnostic: these rows test whether each half of the constructed Dual-Track Split prompt contributes usable evidence, while keeping the removed half-budget unused.

D Orthogonality of Track A and Track B Span Coverage

For the 242 questions Dual-Track Split answered correctly at B=4000, we extracted Track A and

Condition	Overall (n=500)	Multi session (n=121)	Temporal (n=127)	Knowledge (n=72)	Single session (n=64)
Dual-Track Split	**48.4**	**26.4**	**18.9**	**68.1**	**87.5**
Chunks Only (Split Budget)	31.8	10.7	9.4	48.6	65.6
Summaries Only (Split Budget)	15.8	5.0	3.9	12.5	18.8

Table 6: Masking ablation accuracy (%) at $B = 4000$, broken down by task family. Each ablation removes one track from the constructed Dual-Track Split prompt *without redistributing its budget to the other track*; the remaining track still uses only its original half-budget. These conditions test marginal contribution within the split prompt, not optimal single-track allocation.

Task type	N	A-only	B-only	Both	Neither	Unknown
single-session-user	56	0.0%	39.3%	42.9%	1.8%	16.1%
single-session-assistant	44	2.3%	22.7%	70.5%	0.0%	4.5%
single-session-preference	7	28.6%	0.0%	14.3%	57.1%	0.0%
knowledge-update	49	6.1%	6.1%	53.1%	8.2%	26.5%
multi-session	32	3.1%	0.0%	12.5%	31.2%	53.1%
temporal	24	12.5%	4.2%	66.7%	16.7%	0.0%
abstention	30	26.7%	10.0%	16.7%	46.7%	0.0%
Overall	242	7.4%	16.1%	44.2%	15.3%	16.9%

Table 7: Span-coverage breakdown for the 242 items Dual-Track Split answered correctly at $B = 4000$. Each cell gives the percentage of correct items whose gold-answer keywords appear in Track A only, Track B only, both, neither, or are too short to evaluate.

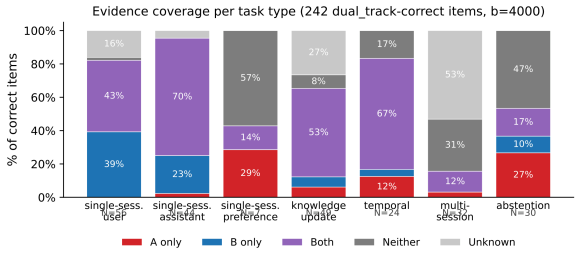

Figure 6: Per-task-type evidence-coverage breakdown for the 242 items Dual-Track Split answered correctly at $B = 4000$. Bars are normalized to 100% within each task type. The pattern is task-dependent: *single-session-user* is overwhelmingly Track-B-covered, *multi-session* has the highest *Neither* fraction, and *abstention* maxes out *Neither* as expected.

Track B contexts post-hoc (no answer or judge LLM calls) and applied a keyword-overlap check against the gold answer: each gold answer is reduced to content keywords (length ≥ 3, stopwords removed) and a track is said to "cover" the answer if at least $\lceil K/2 \rceil$ of its K keywords appear in that track's context after lower-case normalization and punctuation stripping. Table 7 reports the per-task-type breakdown; Figure 6 visualizes the same data as a stacked-bar plot. The *unknown* bucket collects gold answers with too few content keywords to apply the rule (e.g. abstention answers like "I don't know").

Split (A:B)	A chars	B chars	Acc. (%)
25:75	1,000	3,000	43.0
33:67	**1,320**	**2,680**	**54.0**
50:50	2,000	2,000	46.0
67:33	2,680	1,320	46.0
75:25	3,000	1,000	47.0
Dense 64-Word Chunks	4,000 (single track)		38.6

Table 8: Split-ratio sweep on an exploratory 100-item LongMemEval-S check, $B = 4000$, DeepSeek-V4-Flash. The Dense 64-Word Chunks row is the previously reported reference from the main sweep and is included for orientation only. The 50/50 split was fixed *prior* to this analysis; the sweep shows that performance is not isolated to a single ratio.

E Task-Type and Threshold Diagnostics

Figure 7 and Figure 8 are descriptive diagnostics rather than additional primary comparisons. They show where the split appears strongest by task type and how its advantage over a conventional dense chunk baseline changes with budget; the matched allocation claim remains the Dual-Track Split versus BM25 Track-B Chunks comparison in Table 1.

F Split-Ratio Robustness

A natural concern is whether the 50/50 split was selected to fit the LongMemEval distribution. We evaluate five split ratios (25/75, 33/67, 50/50, 67/33, 75/25) on an exploratory 100-item check at $B = 4000$ chars (Table 8; Figure 9). The 100 items were selected before inspecting split-ratio outcomes and are used only for sensitivity analysis.

Across the five tested ratios, accuracy ranges from 43.0% to 54.0%, and all five exceed the single-track Dense 64-Word Chunks reference (38.6%) by a substantial margin. The peak is 33:67 (54.0%), with 50:50 at 46.0%; the summary-leaning half of the grid (50:50 through 75:25) spans only 1 point (46.0–47.0%), indicating stable performance once both tracks receive substantial budget. The 33:67 peak should not be over-interpreted as a

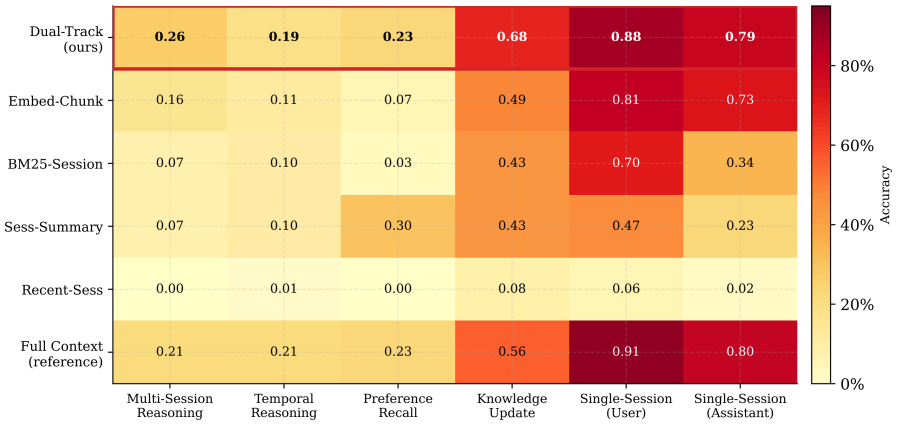

Figure 7: Task-type accuracy at $B = 4000$ for representative baselines and the full-context reference. The heatmap is used to inspect task heterogeneity: Dual-Track Split is strongest on knowledge-update and single-session categories, while all budgeted methods remain weak on multi-session and temporal reasoning. The matched Track-B concentration control is reported separately in Table 1.

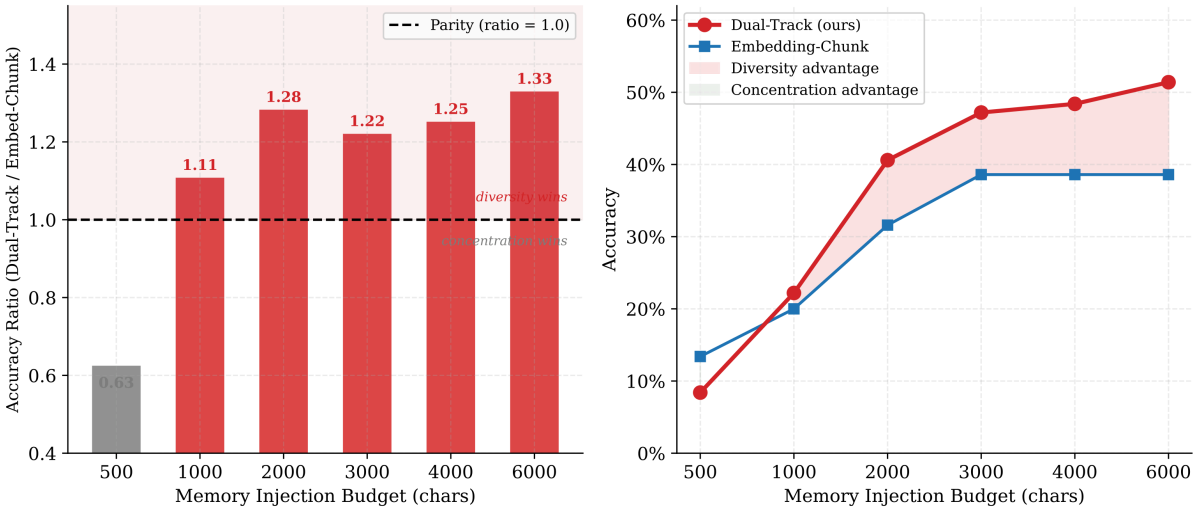

Figure 8: Budget-threshold diagnostic relative to the dense 64-word chunk baseline. Left: the Dual-Track Split / Dense 64-Word Chunks accuracy ratio crosses parity after the smallest budget and remains above 1.0. Right: Dual-Track Split continues improving as the budget grows, while the dense chunk baseline saturates. This figure is descriptive because Dense 64-Word Chunks is not the matched concentration control.

tuned optimum given the smaller exploratory sample ($n = 100$); on the full evaluation set, small differences at this scale are within expected sampling noise. The practical takeaway is narrower: the 50:50 point is not an isolated success case in this exploratory sample. We retain 50:50 as a parameter-free default rather than treating the 33:67 peak as a tuned optimum.

G LoCoMo Cross-Benchmark Check

Strategy	1k	2k	4k
Dual-Track Split (**ours**)	36.0	**52.5**	**53.5**
BM25 Track-B Chunks	**35.6**	40.2	43.8
BM25 Full Sessions	31.5	45.5	50.0
BM25 Session Summaries	28.5	31.0	45.0
Dense 64-Word Chunks	30.0	32.0	31.0
Recent Full Sessions	20.5	20.5	23.5

Table 9: LoCoMo cross-benchmark accuracy (%, 500 items). Unlike on LongMemEval-S, the split does not trail the concentration control at low budget; Dual-Track Split is never weaker than BM25 Track-B Chunks at any tested budget. The absence of a low-budget reversal suggests that the exact crossover pattern is benchmark-dependent, while the split can remain competitive or stronger under another benchmark.

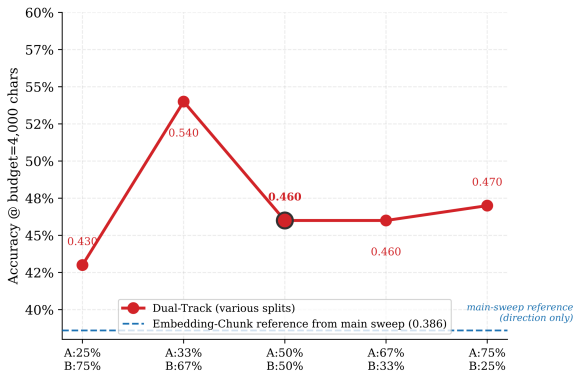

Figure 9: Split-ratio sweep on an exploratory 100-item LongMemEval-S check, $B = 4000$. The dashed line marks the previously reported Dense 64-Word Chunks reference from the main sweep; this cross-run comparison is directional only. The summary-leaning half of the grid (50:50 through 75:25) spans only 1 point, so the 33:67 peak should be interpreted as evidence that 50:50 is not isolated rather than a tuned optimum.