

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

MINTEval: Evaluating Memory Under Multi-Target Interference in Long-Horizon Agent Systems

Hyunji Lee[1]* **Justin Chih-Yao Chen**[1]* **Joykirat Singh**[1] **Zaid Khan**[1]
Elias Stengel-Eskin[2] **Mohit Bansal**[1]
[1]UNC Chapel Hill [2]The University of Texas at Austin

Motivation

- Agents work over **long horizons** where information keeps changing.
- New facts revise old ones, so memories **interfere** with each other.
- Users ask about **the latest state** and also **earlier states**.
- Existing benchmarks mostly test **static, independent recall**.

Comparison With Prior Memory Benchmarks

Table 1: Comparison of MINTEVAL with prior memory benchmarks. We categorize benchmarks by (1) input context properties: interdependent inputs (**Interdep.**), dense interference ($\geq$10 depth; **Interference**), and multi-domain coverage (**M-Domain**); and (2) question properties: multi-target aggregation (**Aggr.**) and lookback to earlier context (**LookBack**).

	Input Context			Question Type	
Benchmark	Interdep.	Interference	M-Domain	Aggr.	LookBack
MemoryAgentBench [Hu et al., 2026]	✗	✗	✓	✓	✗
Mem-α [Wang et al., 2025]	✗	✗	✓	✗	✗
Locomo [Maharana et al., 2024]	✓	✗	✗	✓	✗
LongMemEval [Wu et al., 2025]	✓	✗	✗	✓	✗
BEAM [Tavakoli et al., 2026]	✓	✗	✗	✓	✗
StoryBench [Wan and Ma, 2025]	✓	✗	✗	✗	✗
OAKS [Kim et al., 2026]	✓	✗	✗	✓	✗
MINTEVAL (Ours)	✓	✓	✓	✓	✓

MINTEval Overview

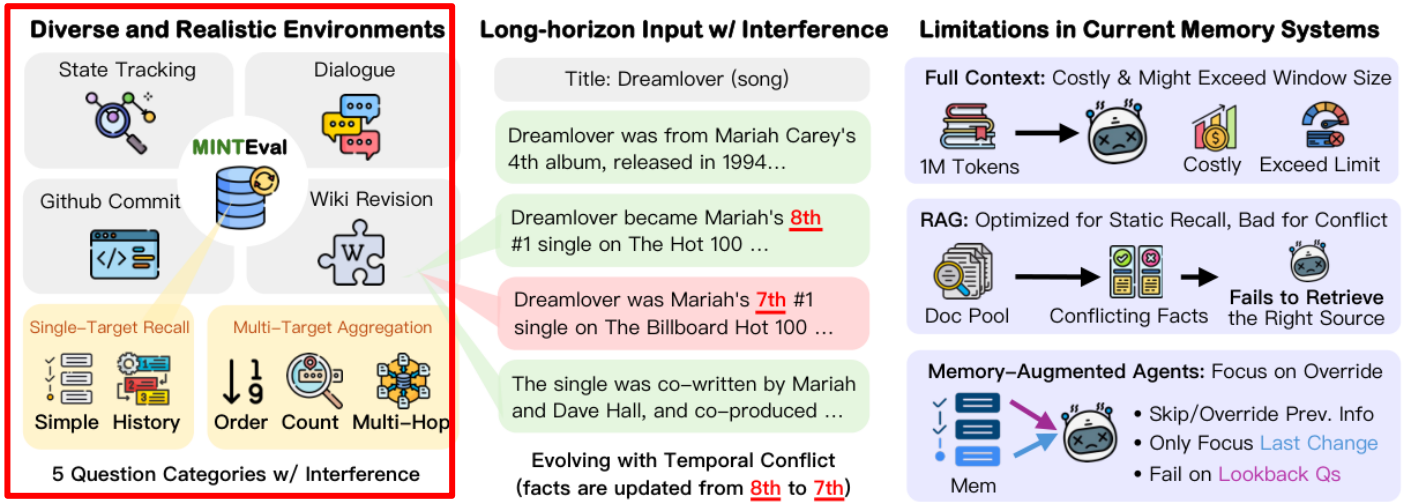

Figure 1: **Left:** MINTEVAL spans four realistic domains: state tracking, dialogue, GitHub commits, and Wikipedia revisions, with five question categories probing different aspects of memory behavior. **Middle:** The contexts are inherently dynamic and continuously evolving, naturally creating frequent destructive interference. **Right:** Existing memory systems show distinct failure modes: (1) full-context methods are computationally expensive and exceed context limits, (2) RAG systems often retrieve incorrect evidence due to conflicting information, and (3) memory-augmented agents overemphasize recent information and underuse historical context, hurting lookback-style queries.

MINTEval Overview

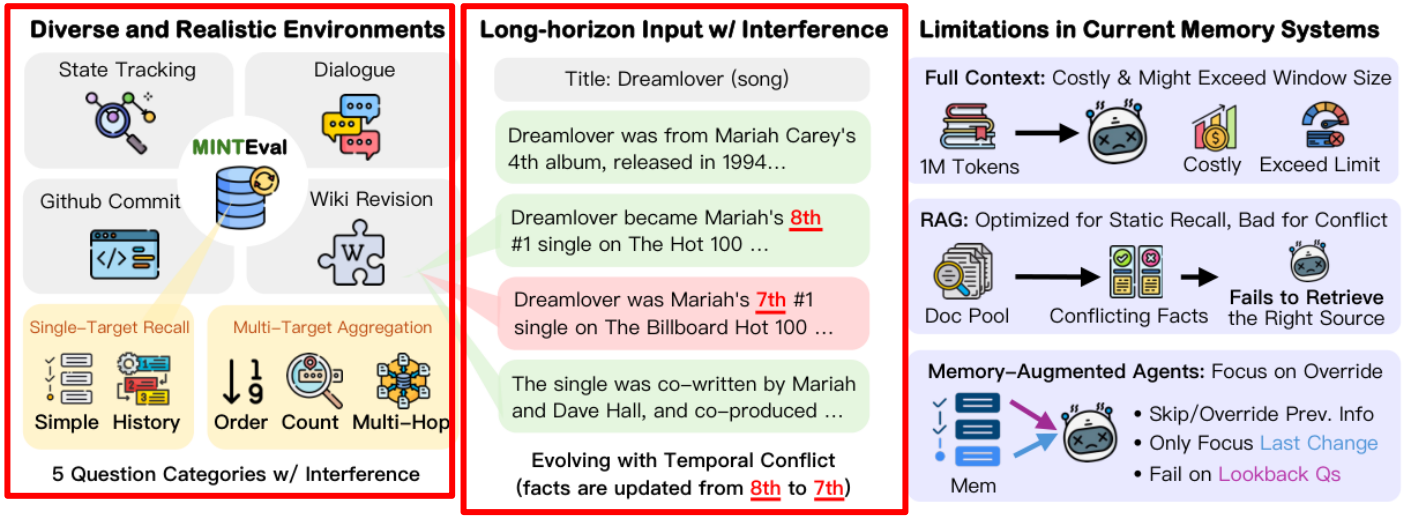

Figure 1: **Left:** MINTEVAL spans four realistic domains: state tracking, dialogue, GitHub commits, and Wikipedia revisions, with five question categories probing different aspects of memory behavior. **Middle:** The contexts are inherently dynamic and continuously evolving, naturally creating frequent destructive interference. **Right:** Existing memory systems show distinct failure modes: (1) full-context methods are computationally expensive and exceed context limits, (2) RAG systems often retrieve incorrect evidence due to conflicting information, and (3) memory-augmented agents overemphasize recent information and underuse historical context, hurting lookback-style queries.

Five Question Types

Category	Type	Example
Single-Target Recall	Simple	How many floors does the article state the building has?
	History	In the version two edits prior, which team is 1919 County Champion?
Multi-Target Aggregation	Ordering	In which order was the section added to the article?
	Counting	How many different individuals have been listed as the album's producer?
	Multihop	What was episode 5's title just before episode 4's third title change?

Main Result

- Average: **27.9%**
- Best system: **33.4%**
- Simple: **47.5%**
- History: **21.0%**
- Aggregation: **26.5%**

Table 4: Results on Qwen3.6-35B-A3B. We compare three categories of methods: Full Context, RAG, and Memory-Augmented Agents. Cells are color-coded by score to highlight performance patterns, transitioning from dark red (lowest) through light red and light green to dark green (highest).

		Full	RAG	HippoRAG	AtomMem	Mem-α	MemAgent
bAbI	Simple	57.4	66.7	70.0	65.2	82.6	85.7
	History	16.1	16.7	33.3	36.3	44.9	36.0
	Ordering	22.0	37.5	50.0	58.1	64.7	59.0
	Counting	40.5	80.0	80.0	43.8	70.4	24.3
	Multihop	30.8	40.0	41.7	30.7	61.0	51.7
Wiki Revisions	Simple	23.3	36.7	37.9	16.9	49.9	54.2
	History	14.5	30.2	31.1	15.7	20.6	28.8
	Ordering	10.9	6.9	4.1	2.4	13.5	38.3
	Counting	22.2	15.9	19.1	14.3	25.0	36.5
	Multihop	11.2	26.8	27.1	16.2	17.3	23.7
Git Commits	Simple	82.0	81.5	81.9	40.8	71.7	82.3
	History	27.1	30.1	30.6	19.0	4.8	24.0
	Ordering	17.7	40.6	44.5	13.8	17.0	55.9
	Counting	21.1	38.9	14.1	24.7	18.8	51.6
	Multihop	13.1	12.8	39.6	27.3	8.5	34.7
HorizonBench	Simple	11.3	11.6	12.1	4.4	7.5	7.5
	History	9.9	10.3	10.9	2.4	5.8	3.8
	Ordering	3.5	4.2	3.9	1.0	0.4	6.7
	Counting	0.8	2.8	4.2	2.9	1.6	1.8
	Multihop	11.9	29.7	33.9	30.0	24.0	28.1
Overall Avg.		21.0	29.5	32.3	22.1	28.0	33.4

Where Do the Errors Come From?

- Evidence found: only **58.3%**
- Memory gap: **41.7%** drop
- Answer gap: **25.2%** drop
- Bottleneck: **building memory**

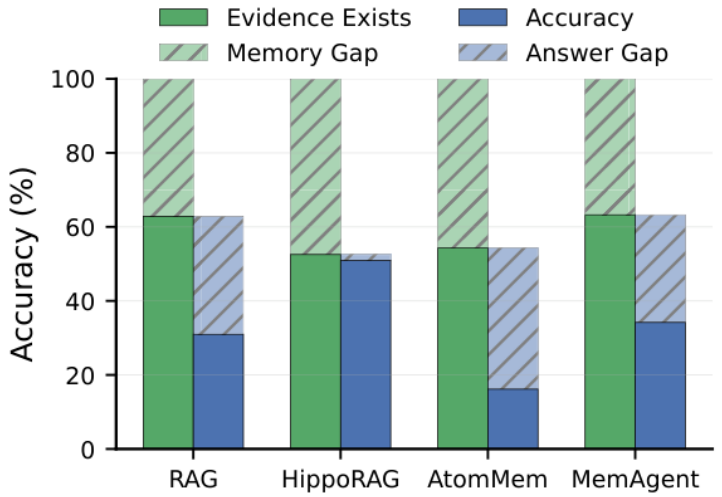

Figure 2: Error due to missing evidence in memory (green) or incorrect answers despite the evidence being present (green–blue gap). Only 58.3% of cases contain the required evidence, making retrieval/memory construction the main bottleneck; answering errors add a 25.2% drop. A perfect system would reach 100%.

Longer Lookback Hurts Performance

- History questions on Wiki
- Full context: **78%** to **12%**
- Memory agents: flatter, but low
- **Temporal cues** reduce the drop

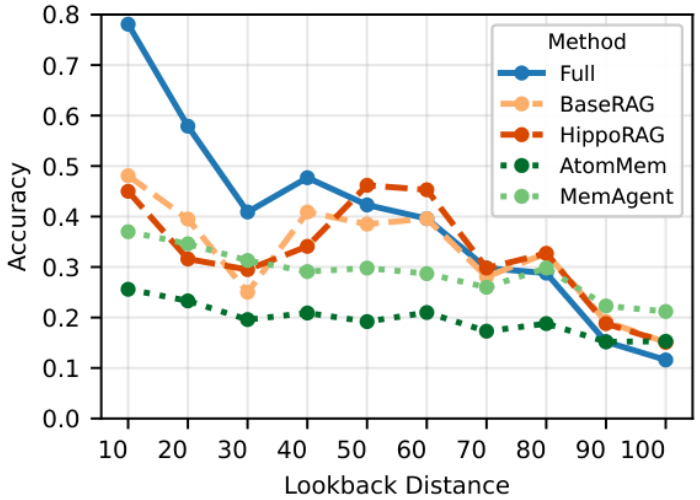

Figure 3: Performance (y-axis) vs. Lookback Distance (x-axis). Accuracy drops as lookback distance increases, with the largest drops in Full Context and retrieval methods (RAG, HippoRAG). Memory-based agents degrade less, suggesting greater robustness from better temporal encoding and compact historical representations.

Takeaways

- Current memory systems average only **27.9%** on MINTEval.
- **Lookback** and **aggregation** are much harder than simple recall.
- Most errors come from **building the memory**, not from answering.
- Agents **insert** 76.8% of the time and rarely update or delete.

Code and data: github.com/amy-hyunji/MINTEval