Memtrack: Evaluating Long-Term Memory and State Tracking in Multi-Platform Dynamic Agent Environments

Darshan Deshpande, Varun Gangal, Hersh Mehta, Rebecca Qian, Anand Kannapan, Peng Wang

Patronus AI

Dataset

Motivation

- Memory benchmarks focus on conversational setups → enterprise envs require multi-platform, dynamic context switching
- Extant benchmarks (LoCoMo [1], LongMemEval [3]) thorough, but limited to single-thread conversations
- Real-world agentic tasks involve codebases, ticketing systems, and async communication across platforms
- There is a need for evaluating memory acquisition, selection, and conflict resolution in realistic organizational workflows

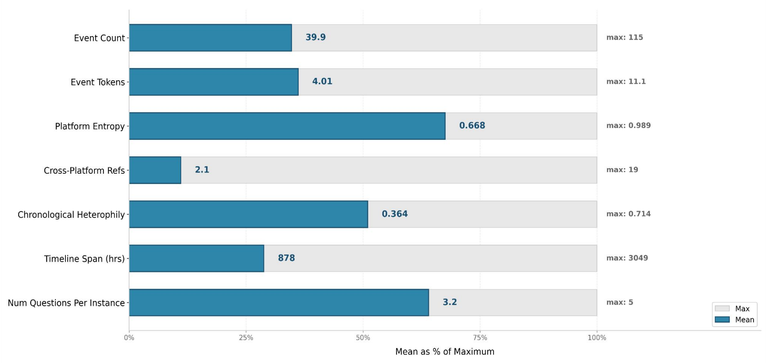
Methodology

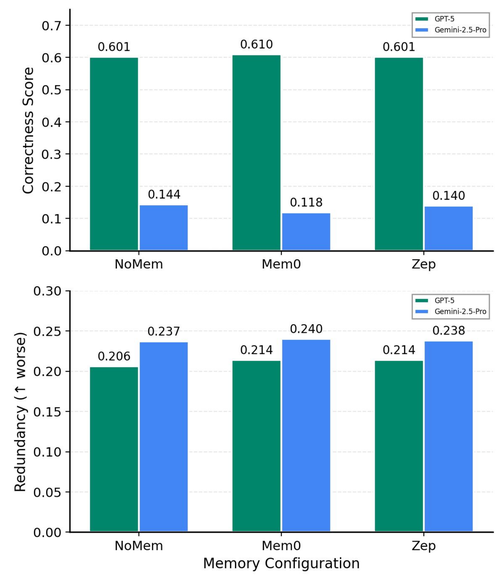
- MEMTRACK provides a containerized multi-platform environment with:
  - **Platforms**: Slack, Linear (ticketing), Git (Gitea), Live Notifications
  - **Dataset**: 47 instances with chronologically interleaved timelines across platforms, containing noisy, conflicting, and cross-referencing information
  - **Data Curation Approaches**:
    - Bottom-up: Agent-based synthesis from closed GitHub PRs
    - Top-down: Manual expert-driven design from real SWE experience
    - Hybrid: Interactive human-LLM iterative refinement
- Methods Tested:
  - LLM+NoMem, LLM+MEM0, LLM+ZEP
  - We experiment with both LLM = GPT-5 and Gemini-2.5-Pro

Example Timeline

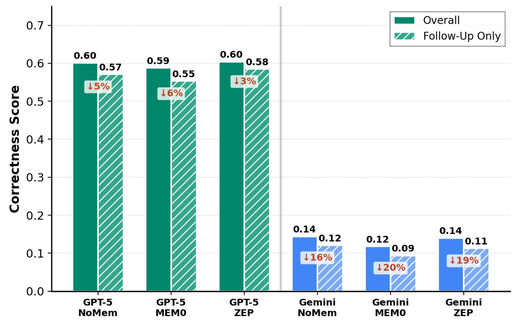
Platform	Events
Linear	"timestamp": "20250410T0900", "title": "MLE-Bench Integration for Competition Analysis Pipeline", "description": "Integrate MLE-Bench framework into our ML competition analysis pipeline to automate competitor analysis and leaderboard tracking. This will enable automated data preparation and performance benchmarking for internal ML competitions.", "team": "ml", "priority": "high", "lead": "sarah_chen", "attached_resources": ["/design/mle-bench-integration-spec.pdf", "/research/competition-analysis-requirements.md"]
...	...
Slack	"timestamp": "20250410T1430", "channel": "#ml", "sender": "sarah_chen", "message": "The error is happening during dataset download. Looks like it's in one of the core files - either mlebench/data.py, mlebench/utils.py, or maybe mlebench/registry.py. Import failure in some exception handling code around lines 180-200."
Linear	"timestamp": "20250410T1630", "title": "Code Analysis: MLE-Bench Import Error Source Investigation", "description": "Source code analysis of import failures in MLE-Bench data preparation module. Multiple import statements identified in core data processing files. Issues located in exception handling code sections. Detailed line-by-line analysis required for complete understanding of scope.", "team": "ml", "priority": "medium", "lead": "sarah_chen", "status": "done"
...	...
Slack	"timestamp": "20250411T1440", "channel": "#ml", "sender": "david_wong", "message": "Check the compatibility testing ticket - I documented the specific working version there. The one that passed all our preparation tests."
Slack	"timestamp": "20250411T1600", "channel": "#engineering", "sender": "sarah_chen", "message": "@alex_kim Could be related to the new Istio service mesh rollout. I'll check the telemetry data."

Dataset Statistics





- **General-to-Specific:** Agents repeat broad calls with specific variants (e.g., list_tickets() → list_tickets(limit=100))
- **Repeated After Interlude:** Information re-accessed after 3+ turns gap
- **Progressive Widening:** Gradual limit increases in exploration queries



We notice a small but statistically significant drop in mean correctness when only follow-up questions are considered. Furthermore, this pattern holds consistently across all methods.

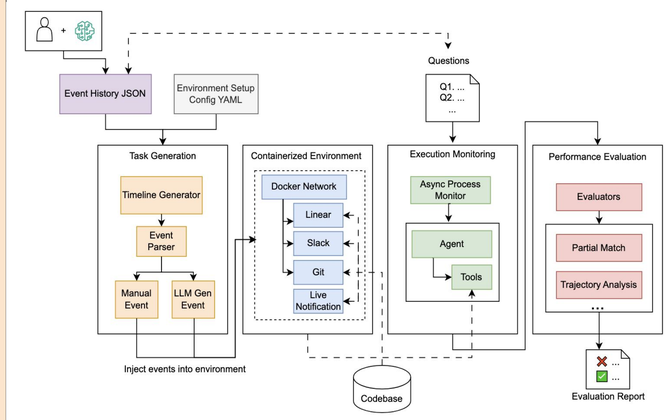
Example Question Answer Sequence

1. Based on sarah_chen's vague mention of "lines 180-200" and 'exception handling' , clone the repository and examine the source: How many total python files in the entire mlebench/directory [including subdirectories] contain any import statement that references the kaggle package?

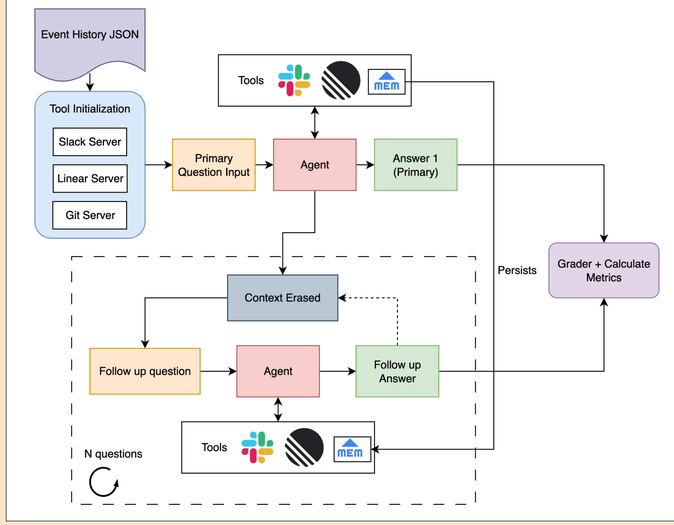
2. Following conversations about competition specific models and experiments directory, investigate the actual repo structure: What tis the exact filename [w/o path] of the python file in the mlebench competition/subdir that contains longest individual function by line count?

Answers: 9 , prepare.py

Environment Design



Evaluation Workflow



Future Work

- **Agentic Actions:** Extend to agents that create tickets, send Slack messages, and participate in timeline evolution
- **Domain Expansion:** Adapt MEMTRACK to marketing, sales, and other enterprise contexts with overlapping internal/external communications

[1] Maharana et al. LoCoMo: Long-term conversational memory evaluation
[2] Rasmussen et al. Zep: Temporal knowledge graph for agent memory
[3] Wu et al. LongMemEval: Chat assistants on long-term memory
[4] Chhikara et al. Mem0: Scalable long-term mem. for agents
[5] Jimenez et al. SWE-Bench
[6] Packer et al. MemGPT