NEURAL INFORMATION PROCESSING SYSTEMS

# Rethinking How to Remember: Beyond Atomic Facts in Lifelong LLM Agent Memory

Jingwei Sun · Jianing Zhu · Jiangchao Yao · Tongliang Liu · Bo Han

https://tmlr-trimem.github.io/ & https://github.com/tmlr-group

# Outline

- Background and Motivation
- System Design of Our Method
- Experimental Results
- Conclusion

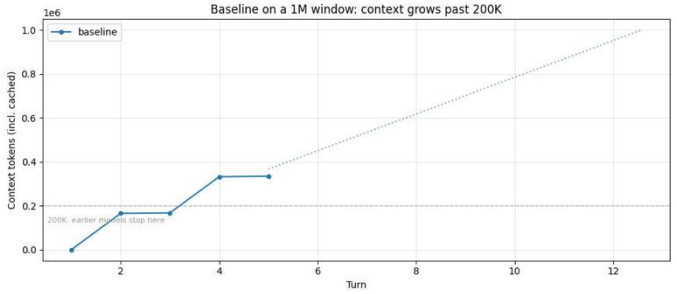
# Outline

- Background and Motivation
- System Design of Our Method
- Experimental Results
- Conclusion

# Why Do Agents Need Memory?

## 1 Agent tasks are longer

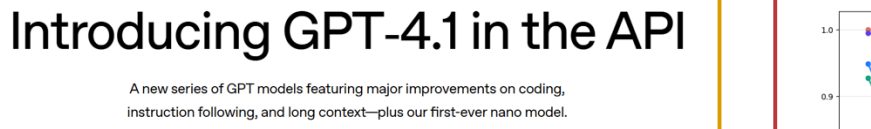
```
┌ [baseline]
| turn  1  ctx=  1,058
|          read_file       /research/celegans_review.md  → ~27,465 tok
|          read_file       /research/drosophila_review.md  → ~27,435 tok
|          read_file       /research/yeast_review.md  → ~27,320 tok
|          read_file       /research/killifish_review.md  → ~27,349 tok
| turn  2  ctx=166,043
|          record_finding  "## BATCH 1 NOTES: High-Throughput Model Organisms ..."
| turn  3  ctx=168,242
|          read_file       /research/mouse_review.md  → ~27,390 tok
|          read_file       /research/zebrafish_review.md  → ~27,363 tok
|          read_file       /research/nmr_review.md  → ~27,287 tok
|          read_file       /research/rhesus_review.md  → ~27,285 tok
| turn  4  ctx=332,821
|          record_finding  "## BATCH 2 NOTES: Low-Throughput Model Organisms —..."
| turn  5  ctx=335,279  (final answer)
└ completed: 5 turns, peak ctx 335,279, final ctx 335,279, 0 context event(s)
```




As the operation time increases, the context gradually expands.

Sources: Anthropic · Context Engineering

## 2 Windows keep expanding

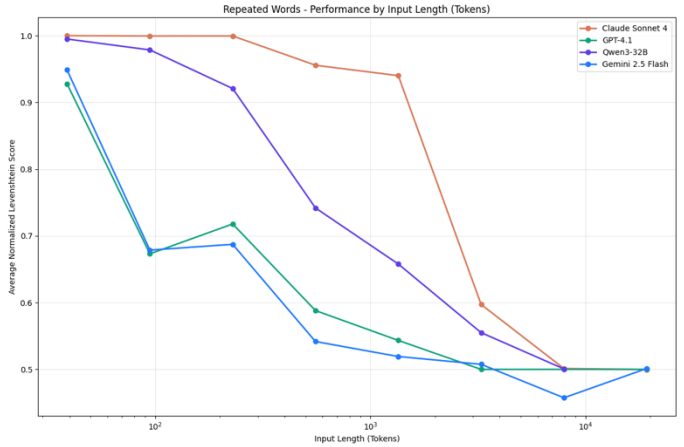



**128K** GPT-4o → **1M** GPT-4.1

The context length supported by the model is also getting longer and longer.

Source: OpenAI GPT-4.1 announcement

## 3 Reliability still degrades




Model performance still degrades as input length increases,

Source: Chroma technical report

**Takeaway: Larger windows are not enough, agents need memory module to manage context.**

https://tmlr-trimem.github.io/ & https://github.com/tmlr-group

# Where Do Agents Need Memory?

*Memory supports continuity across conversations, tasks, and changing environments.*

## Personal Assistants

Keep user preferences and past requests across sessions.

Prefers quiet cafés → Recommend a quiet café

## Coding Agents

Recall project decisions, previous edits and test results.

Parser test failed → Resume the fix

## Research Agents

Preserve sources, intermediate findings and open questions.

Read paper A previously → Insight to following project

## Social & Game Agents

Remember prior interactions, world states and learned skills.

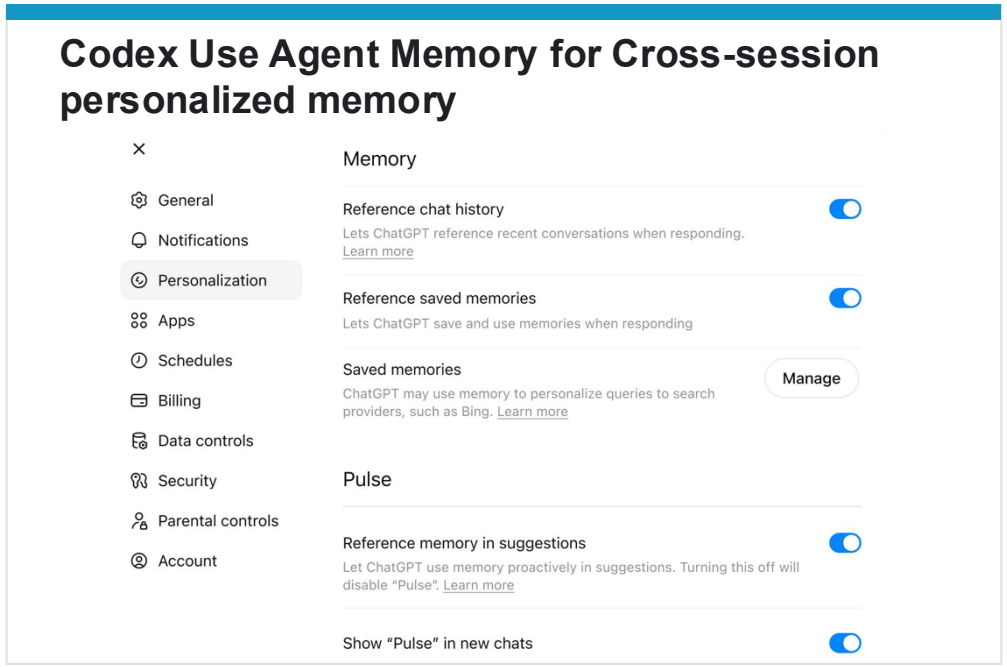
Plan a party → Follow up next day

Across these settings, agents must preserve, update, and retrieve useful information over time.

# Application: Personal Assistants

*Memory enables personalized responses across conversations.*

## Remember user preferences




*An illustrative cross-conversation example*

Earlier: "I prefer vegetarian meals."

↓

**Memory: vegetarian preference**

↓

Later: "Suggest dinner near my hotel."

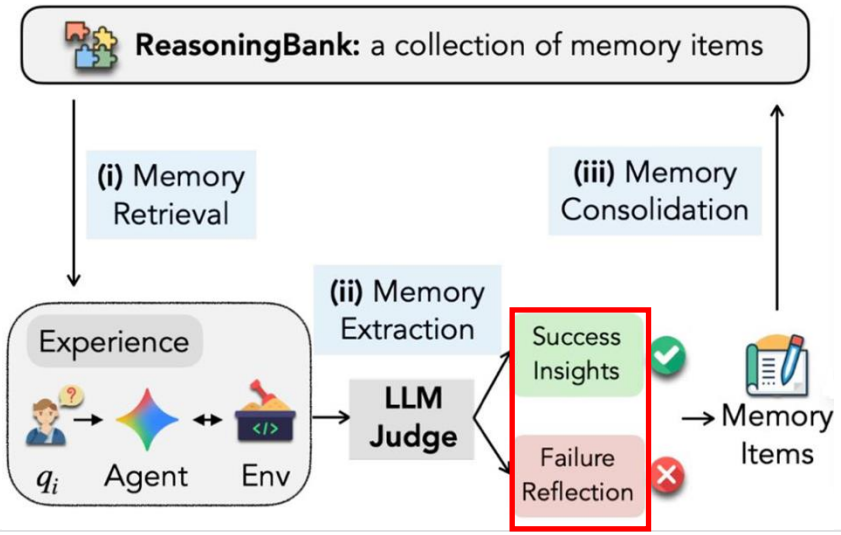
**The assistant can reuse a preference without asking the user to repeat it.**

# Application: Coding Agents

Agents can remember why a solution failed and avoid repeating the same mistake.

## Learn from past failures

**Google Research Store Previous Useful Experience in the Memory**




**Illustrative example: error-driven memory**

**Past failure: deploy before running tests**

↓

**Memory: cause + corrective rule**

"Run regression tests before deployment."

↓

**New task: check tests first → avoid recurrence**

**Memory turns past debugging mistakes or successful attempt into reusable rules for future tasks.**

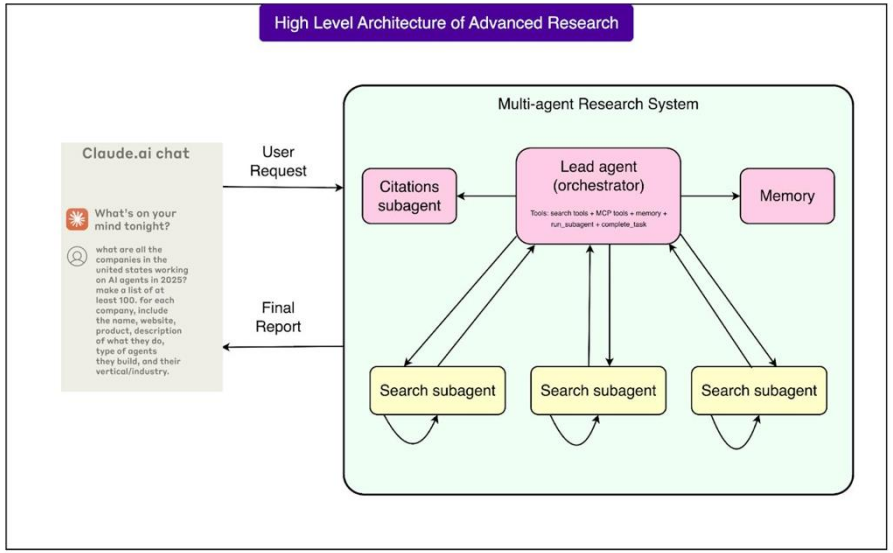
https://tmlr-trimem.github.io/ & https://github.com/tmlr-group

# Application: Research Agents

Agents can connect insights across papers to support future research.

## Build new ideas from prior research

**Anthropic uses memory to gather information from different research tasks.**




**Illustrative example: cross-paper insights**

**Paper A**

**Paper B**

**Research memory**

Connect and preserve findings from different papers

**New Research Insight**

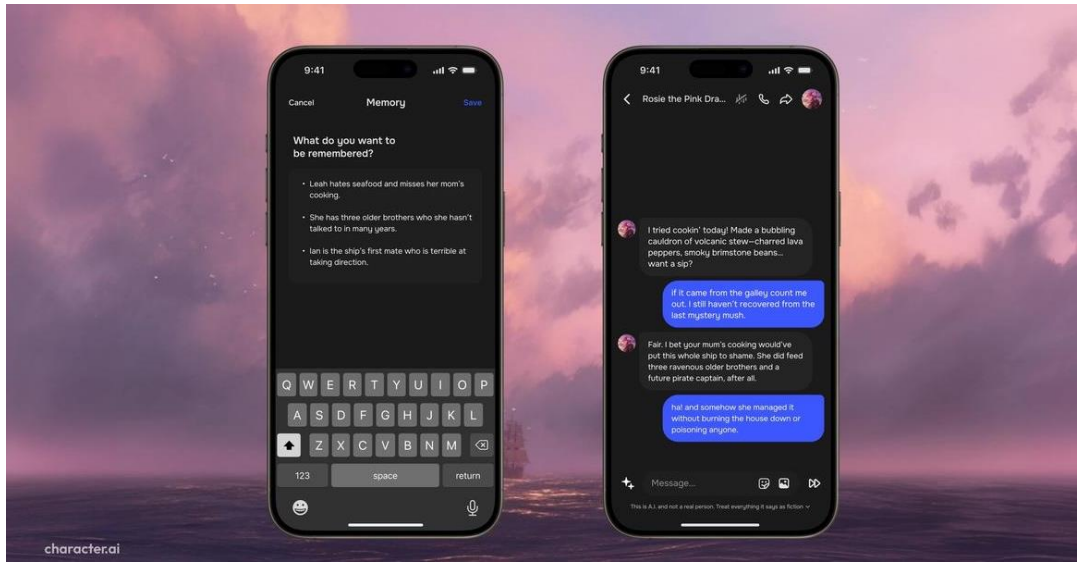
**Accumulated findings can inspire new research hypotheses and directions.**

https://tmlr-trimem.github.io/ & https://github.com/tmlr-group

# Application: Social and Game Agents

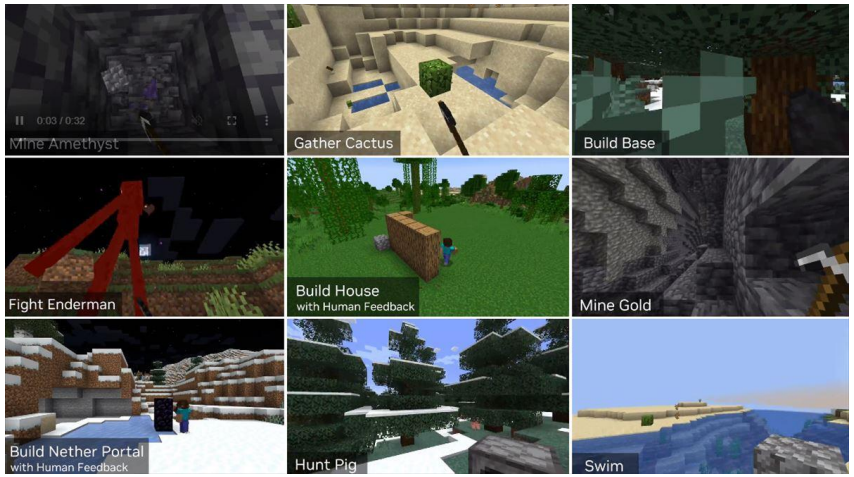
*Memory supports coherent social behavior and reusable skills.*

## Social agents




**Social agents help remember preferences of friends to support social interactions.**

## Game agents




**Explore in unfamiliar environment** → **Skill library**

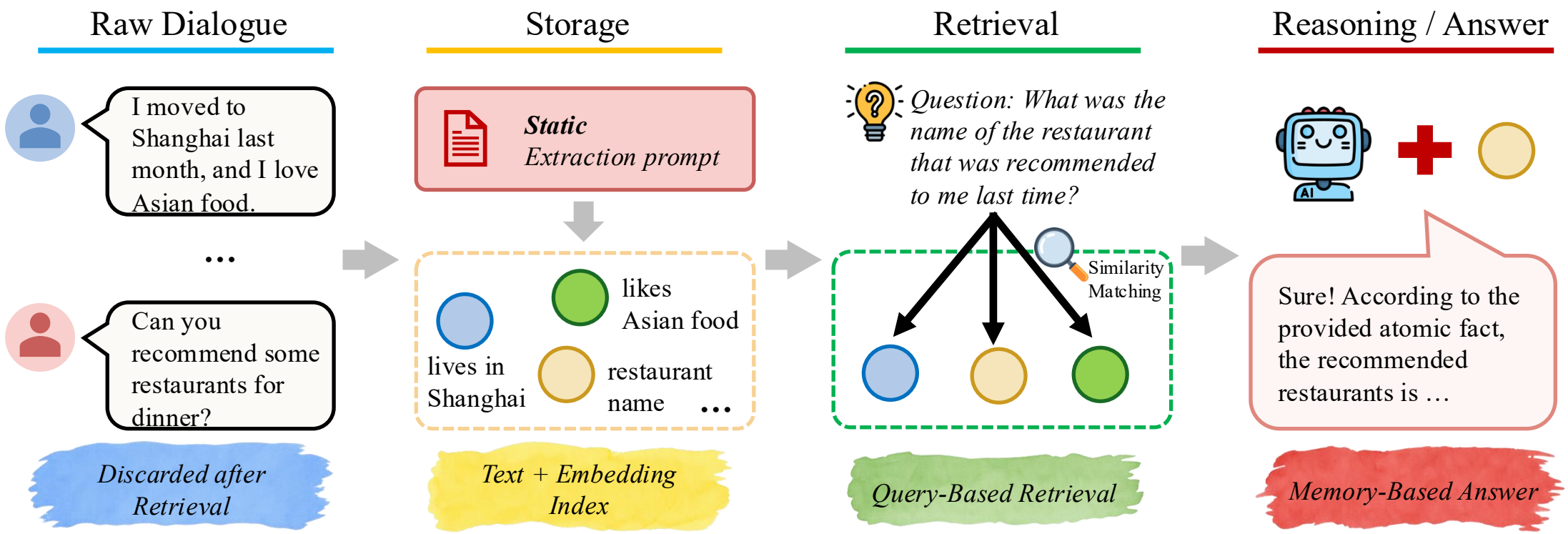
**Agents reuse learned skills for new goals.**

**Memory preserves relationships, events, and skills beyond the current action.**

https://tmlr-trimem.github.io/ & https://github.com/tmlr-group

# Agent Memory: A Typical Pipeline

*Store information from past interactions and retrieve relevant memories for each new query.*

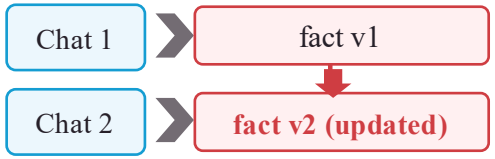



**Agent memory system typically use static prompt to establish a fact-centric pipeline**

# How different Methods Improve the Agent Memory System

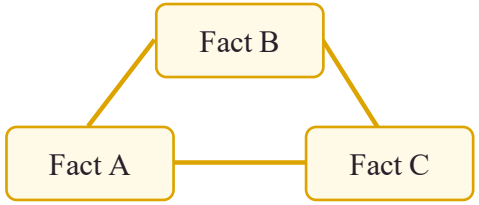
## Mem0

**Enable Update of Old Facts**



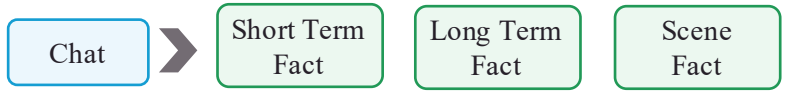

## A-Mem

**Construct Link between Facts**



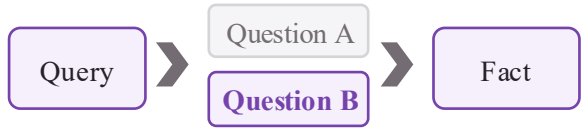

## LightMem

**Different Types of Fact**




## SimpleMem

**Generate Different Retrieval Questions**




**Previous method focus on how to better utilize the atomic facts extracted by static prompts**

# Can Atomic Facts Beneficially Affect all Three Stages of Agent Memory Systems?

**Given Question**

What career did Caroline choose?

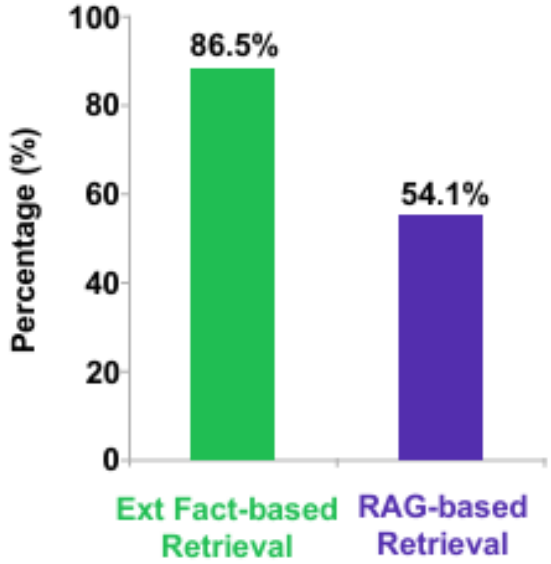
**Full-context RAG based retrieval return sentences contain similar words, making it prone to confusion during retrieval:**

career | jobs | occupation | ...

**Atomic-fact based retrieval enable more accurate matching**

Caroline helps trans people through **counseling.**

**What proportion of the relevant evidence for the questions can be successfully extracted?**



# Can Atomic Facts Beneficially Affect all Three Stages of Agent Memory Systems?

**Raw dialogue contain full information:**

Caroline helps **trans people** through counseling.

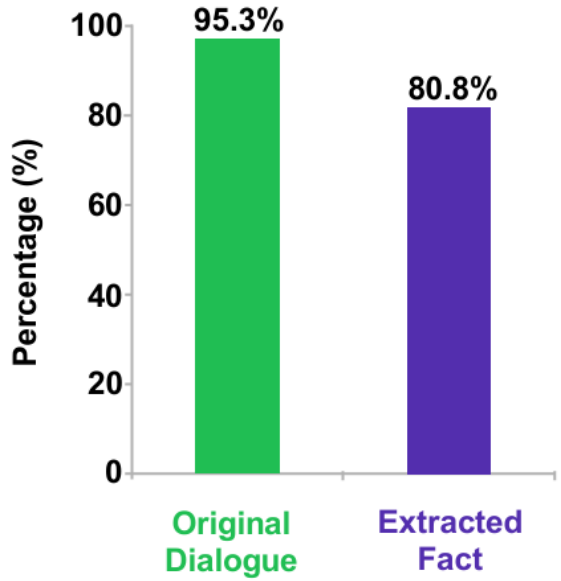
*Fact extraction*

**Stored atomic fact loss details:**

Caroline wants to pursue counseling…

**"trans people" lost**

**How many reference answer tokens are covered?**



# Can Atomic Facts Beneficially Affect all Three Stages of Agent Memory Systems?

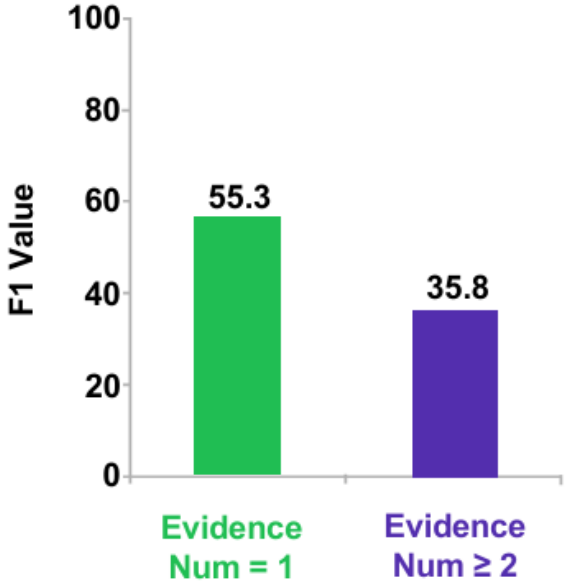
**Some understanding needs to be constructed based on scattered facts.**

**Fact 1**
Caroline went on a family road trip last weekend.

**Fact 2**
Her son had an accident last weekend.

**Caroline won't go to another trip recently.**

**How significant is the difference in the reasoning performance regarding two types of questions?**



# Can Static Fixed Prompt Adapt to Heterogeneous Real-World Dialogues?

## Explicit Expression

Jolene: I use the pomodoro technique to study.

↓ Extraction

**Jolene uses pomodoro technique**

✓ **Technique name preserved**

## Implicit Expression

Tim: I spend 25 minutes studying, then 5 minutes off.

↓ Extraction

**Tim uses a timed study technique**

✗ **"Pomodoro" not inferred**

**Reasoning Performance Comparison on Dialogues between Different Speakers (F1)**



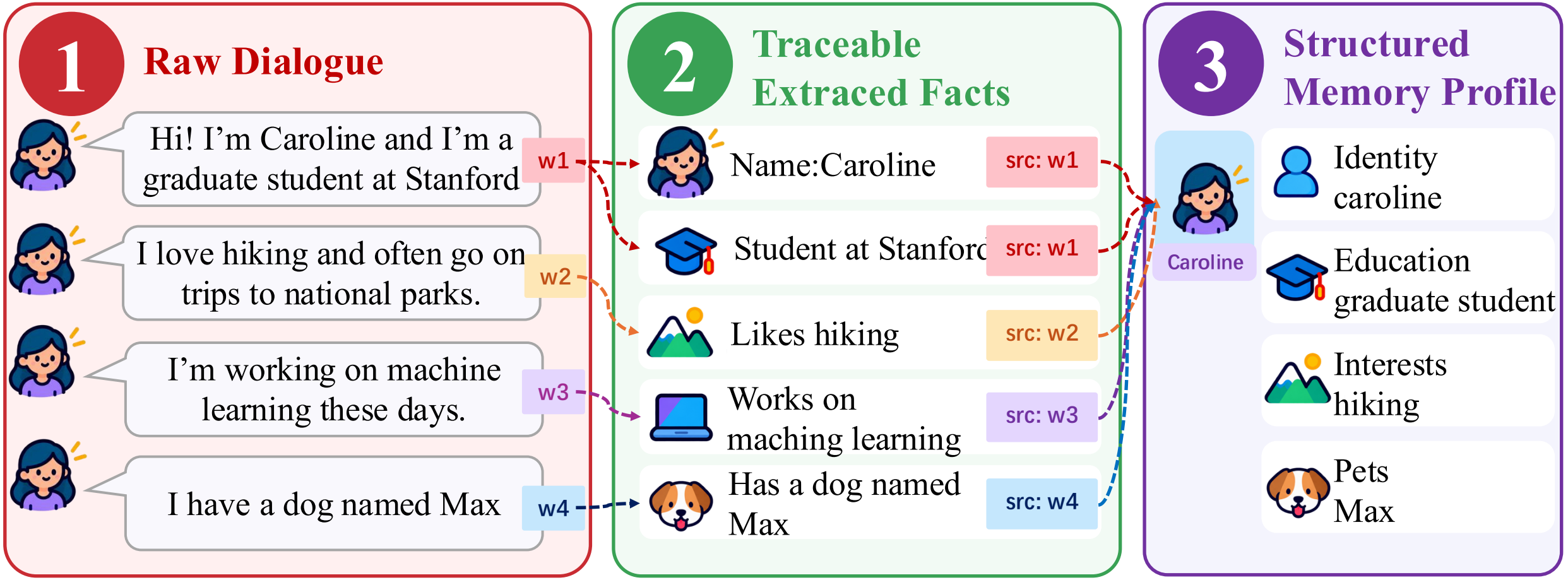

**Takeaway: Reasoning performance fluctuates as diverse conversation style makes fixed prompt unstable**

https://tmlr-trimem.github.io/ & https://github.com/tmlr-group

# Outline

- Background and Motivation
- System Design of Our Method
- Experimental Results
- Conclusion

# How TriMem Avoid Lossy Storage and Shallow Reasoning




*Maintain source identifier and integrate scattered facts into structured profile*

https://tmlr-trimem.github.io/ & https://github.com/tmlr-group

# How TriMem Maintain Efficient Retrieval

## 1 Required Information Analysis



Where does Caroline like to go on trips



**Query Analysis**

To answer this question, we need to search:

destination | trips | travel

hiking | caroline | ...

## 2 Key-Fact Matching

1  likes hiking on national park

2  national parks

3  student at stanford

| src | fact | profile |
|---|---|---|
| I love hiking and often go on trips to national parks. | likes hiking | caroline profile |

*TriMem retrieves efficiently by matching over key facts, Raw dialogue and profile can also be obtained by the pre-established link*

# How TriMem Continuously Improve the System Prompts

## 1 Answer vs Reference

| | Answer | Reference | |
|---|---|---|---|
| When did Caroline go to national park? | 2021 | 2023 | ✗ |
| Where did Caroline go yesterday? | Park | LGBTQ group | ✗ |
| What is Caroline's major? | Machine learning | Machine learning | ✓ |

## 2 Reflect and Prompt Refinement

- Wrong time information (2021 vs 2023)
- Wrong location information (the park vs LGBTQ group)



Reflection Agent

- Add clear rules for time expression
- Include key entities and constraints
- Add more personal backgrounds

*Utilize an additional reflection agent to reflect and optimize the prompt*

# Outline

- Background and Motivation
- System Design of Our Method
- Experimental Results
- Conclusion

# LoCoMo Benchmark

*Long conversations across many sessions*

## Dataset and Example

10 long conversations; about 600 turns and 16,000 tokens each, over up to 32 sessions.

**After Each Conversation:**

| | |
|---|---|
| **May 2023** | Caroline attended an LGBTQ+ support group. |
| **June 2023** | She spoke at a school about LGBTQ+ inclusion. |
| **Later** | She also took part in a Pride parade. |

**Several Related Questions are Asked:**

**Question: Which LGBTQ+ activities did Caroline join?**

**Answer: Support group, school talk, Pride parade.**

## Question Categories

**Single-hop**

Recall a fact from one message.

**Multi-hop**

Combine facts from separate messages.

**Temporal**

Identify when events happened or their order.

**Open-domain**

Use dialogue context and general knowledge.

**Adversarial**

Recognize unsupported or misleading questions.

*Challenge: How to extract different kinds of information from the dialogue*

# Performance on LoCoMo with High-Capability Models

Table 1: **Performance on the LoCoMo benchmark with High-Capability Models.** TriMem achieves outstanding performance across different models while maintaining high efficiency.

| Model | Method | MultiHop | | Temporal | | OpenDomain | | SingleHop | | Average | | Token |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | BLEU | F1 | BLEU | F1 | BLEU | F1 | BLEU | F1 | BLEU | F1 | Cost |
| **GPT-4.1-mini** | LoCoMo | 8.00 | 17.26 | 10.17 | 14.89 | 8.29 | 16.28 | 17.43 | 19.36 | 13.62 | 17.85 | 16863 |
| | Naïve RAG | 11.49 | 13.24 | 20.52 | 28.80 | 11.79 | 10.75 | 22.85 | 30.29 | 19.59 | 25.64 | 1119 |
| | Mem0 | 28.81 | 31.44 | 35.41 | 46.24 | 18.51 | 17.93 | 31.25 | 35.34 | 30.88 | 35.81 | 1153 |
| | MemoryOS | 16.46 | 24.02 | 34.78 | 46.52 | 14.89 | 19.58 | 36.18 | 43.92 | 30.95 | 39.30 | 936 |
| | A-Mem | 15.11 | 20.66 | 41.57 | 50.94 | 11.18 | 13.20 | 38.25 | 43.72 | 33.01 | 39.10 | 1276 |
| | LightMem | 32.93 | 40.33 | 47.53 | 55.23 | 18.31 | 21.91 | 37.68 | 48.39 | 37.66 | 46.69 | 695 |
| | SimpleMem | 32.40 | 39.33 | 43.69 | 58.01 | 19.56 | 24.50 | 43.41 | 53.99 | 39.97 | 50.30 | 587 |
| | **Ours** | **35.20** | **42.59** | **49.56** | **64.72** | **36.86** | **43.88** | **45.25** | **55.36** | **43.79** | **54.26** | 1217 |
| **GPT-4o** | LoCoMo | 19.64 | 19.20 | 9.50 | 13.95 | 11.87 | 16.60 | 13.81 | 16.12 | 13.86 | 16.26 | 16863 |
| | Naïve RAG | 14.36 | 15.35 | 11.48 | 16.17 | 9.03 | 9.09 | 26.67 | 35.03 | 20.15 | 25.88 | 1119 |
| | Mem0 | 25.52 | 32.36 | 32.48 | 42.70 | 14.50 | 18.50 | 30.02 | 39.84 | 28.74 | 37.74 | 1195 |
| | MemoryOS | 22.52 | 31.76 | 38.31 | 47.08 | 12.91 | 18.06 | 38.26 | 43.67 | 33.81 | 40.60 | 944 |
| | A-Mem | 20.90 | 26.12 | 35.39 | 48.64 | 10.74 | 12.33 | 37.11 | 42.08 | 32.14 | 38.67 | 1152 |
| | LightMem | 35.30 | 45.16 | 43.60 | 58.57 | 10.56 | 23.20 | 36.72 | 46.60 | 36.26 | 47.37 | 677 |
| | SimpleMem | 31.34 | 35.58 | 35.78 | 46.96 | 18.96 | 17.01 | 37.11 | 43.94 | 34.64 | 41.36 | 627 |
| | **Ours** | **40.36** | **46.00** | **51.39** | **60.41** | **39.27** | **50.15** | **40.61** | **47.78** | **42.73** | **50.23** | 1272 |
| **GPT-5-nano** | LoCoMo | 20.45 | 19.04 | 12.69 | 16.56 | 13.83 | 20.85 | 13.50 | 15.23 | 14.62 | 16.56 | 16863 |
| | Naïve RAG | 10.13 | 13.29 | 8.78 | 13.09 | 9.25 | 12.24 | 20.29 | 28.44 | 15.34 | 21.46 | 1119 |
| | Mem0 | 22.55 | 28.58 | 35.52 | 48.82 | 18.33 | 16.75 | 28.99 | 35.65 | 28.51 | 35.92 | 1074 |
| | MemoryOS | 10.74 | 23.50 | 32.50 | 39.71 | 10.02 | 20.30 | 34.28 | 40.34 | 28.09 | 35.88 | 952 |
| | A-Mem | 15.54 | 20.11 | 27.23 | 32.43 | 10.86 | 12.55 | 27.26 | 31.91 | 24.09 | 28.65 | 1175 |
| | LightMem | 28.63 | 38.21 | 39.72 | 55.51 | 18.79 | 22.74 | 31.19 | 42.01 | 31.73 | 42.93 | 723 |
| | SimpleMem | 25.42 | 33.28 | 32.15 | 45.75 | 20.77 | 24.31 | 39.65 | 46.71 | 34.30 | 42.65 | 655 |
| | **Ours** | **34.86** | **45.25** | **42.45** | **57.05** | **33.55** | **40.52** | **54.26** | **62.88** | **46.96** | **57.04** | 1256 |

- Our method consistently outperforms prior approaches when integrated with various **high-capability models.**
- The low average tokens consumption of retrieved contexts also demonstrate **the high information density** of TriMem.

https://tmlr-trimem.github.io/ & https://github.com/tmlr-group

# Compatibility on LoCoMo with Efficient Models

Table 2: **Performance on the LoCoMo benchmark with Efficient Models.** TriMem still achieves superior performance-efficiency balance with representative efficient models.

| Model | Method | MultiHop | | Temporal | | OpenDomain | | SingleHop | | Average | | Token |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | BLEU | F1 | BLEU | F1 | BLEU | F1 | BLEU | F1 | BLEU | F1 | Cost |
| **Qwen3-8B** | LoCoMo | 12.67 | 20.54 | 12.32 | 18.55 | 10.59 | 14.39 | 19.76 | 23.78 | 16.34 | 21.51 | 16863 |
| | Naïve RAG | 13.75 | 15.83 | 8.86 | 13.53 | 9.68 | 10.77 | 21.57 | 28.76 | 16.75 | 22.10 | 1119 |
| | Mem0 | 28.32 | 30.07 | 23.15 | 26.15 | 11.79 | 15.15 | 30.75 | 34.97 | 27.54 | 30.10 | 1140 |
| | MemoryOS | 14.38 | 22.72 | 18.67 | 22.79 | 11.06 | 13.52 | 25.65 | 33.52 | 21.22 | 28.06 | 911 |
| | A-Mem | 16.02 | 21.08 | 28.10 | 37.51 | 14.01 | 14.19 | 33.60 | 40.77 | 28.01 | 34.83 | 1180 |
| | LightMem | 22.84 | 32.54 | 37.62 | 48.37 | 18.05 | 19.02 | 23.03 | 31.37 | 25.73 | 34.36 | 740 |
| | SimpleMem | 23.39 | 30.39 | 24.66 | 34.51 | 14.04 | 15.39 | 35.73 | 41.26 | 29.81 | 36.25 | 608 |
| | xMemory | 28.44 | 39.13 | 28.65 | 35.41 | 17.76 | 21.57 | 40.66 | 50.57 | 34.49 | 43.51 | 2230 |
| | **Ours** | **33.09** | **41.22** | **38.71** | **53.13** | **30.59** | **37.64** | **45.10** | **52.52** | **40.66** | **49.65** | 1339 |
| **Llama-3.1-8B-Ins** | LoCoMo | 13.73 | 23.36 | 13.15 | 20.30 | 11.54 | 19.42 | 18.64 | 25.86 | 16.15 | 23.84 | 16863 |
| | Naïve RAG | 14.31 | 16.18 | 7.63 | 12.28 | 9.09 | 11.58 | 22.99 | 31.02 | 17.33 | 23.18 | 1119 |
| | Mem0 | 13.27 | 16.40 | 8.26 | 12.62 | 7.45 | 8.45 | 21.75 | 31.28 | 16.49 | 23.24 | 1085 |
| | MemoryOS | 13.57 | 22.63 | 19.18 | 23.31 | 10.59 | 13.01 | 23.46 | 31.05 | 19.95 | 26.77 | 964 |
| | A-Mem | 15.80 | 22.84 | 23.79 | **36.19** | 11.19 | 12.51 | 31.19 | 37.86 | 25.58 | 33.18 | 1340 |
| | LightMem | 13.19 | 19.64 | 16.93 | 28.06 | 17.39 | 20.68 | 27.62 | 41.06 | 22.11 | 33.16 | 758 |
| | SimpleMem | 18.81 | 26.22 | 21.15 | 30.44 | 15.81 | 18.77 | 26.79 | 31.23 | 23.47 | 29.37 | 674 |
| | xMemory | 21.89 | 31.24 | 21.78 | 26.84 | 12.37 | 16.62 | 27.75 | 41.36 | 24.47 | 34.94 | 2375 |
| | **Ours** | **25.42** | **34.56** | **25.98** | 32.36 | **28.40** | **32.71** | **35.76** | **43.20** | **31.37** | **38.70** | 1388 |

- Our system still achieves substantial performance improvements with **open-source efficient models**, demonstrating the extensive superiority and broad applicability of our method

# PerLTQA Benchmark

*Personal memories represented in different forms*

## Dataset and Example

8,593 questions; 141 profiles, 1,339 social links, 4,501 events and 3,409 dialogues.

**Apart from the dialogue, corresponding background is provided, including relationship, events:**

| | |
|---|---|
| **Relationship** | Liu Ming is Liu Liang's brother. |
| **Event 1** | Liu Liang and Liu Ming run a weekly study group. |
| **Dialogue** | Liu Liang: "Li Hua, let's play games at home." |

**Several Related Questions are Asked:**

**Question: What are Liu Liang and his brother doing at home?**

**Answer: Organizing a weekly study group**

## Memory Categories

### Semantic Memory

| | |
|---|---|
| **World knowledge** | General facts used to interpret a question. |
| **Profiles** | Personal details such as jobs or interests. |
| **Social relationships** | Family, friends and other person-to-person links. |

### Episodic Memory

| | |
|---|---|
| **Events** | Past activities and their participants. |
| **Dialogues** | Historical conversations and their context. |

*Challenge: Memory system needs to take into account both the dialogue and the relevant background*

# LongMemEval Benchmark

*Long histories of user–assistant interactions*

## Dataset and Example

500 questions in dated chat histories. The shorter setting still has about 40 sessions and 115,000 tokens.

**Given a long conversion:**

| | |
|---|---|
| **Last month** | Hawaii family trip, with snorkeling together. |
| **Last week** | The family visited Paris just last week. |
| **Today** | The family go to the restaurant for dinner. |

**Only One Related Question is asked:**

**Question: Where is the most recent trip planned?**
**Answer: Paris**

## Five Memory Abilities

**Information extraction**
Recall details stated by the user or assistant.

**Multi-session reasoning**
Combine evidence from different sessions.

**Temporal reasoning**
Use dates, durations and sequence of events.

**Knowledge updates**
Prefer the latest information when facts change.

**Abstention**
Do not guess when history has no answer.

*Challenge: The information density is low due to each long conversation contains only a few pieces of information relevant to a single question.*

# Generalization on Different Datasets.

Table 3: Performance comparison on the PerLTQA benchmark. All values report the LLM-judged correctness score (%), where `gpt-4.1-mini` evaluates whether each predicted answer matches the ground truth. TriMem achieves the best performance across different sub-tasks.

| Method | Qwen3-8B | | | | Llama-3.1-8B-Ins | | | |
|---|---|---|---|---|---|---|---|---|
| | Profile | Social Relationship | Events | Dialogues | Profile | Social Relationship | Events | Dialogues |
| Full-Context | 65.80 | 56.72 | 52.75 | 18.51 | 52.46 | 54.58 | 47.54 | 17.27 |
| Mem0 | 89.56 | 76.46 | 66.48 | 27.59 | 73.04 | 72.29 | 57.14 | 26.31 |
| LightMem | 64.93 | 78.00 | 73.03 | 47.01 | 53.85 | 74.08 | 69.21 | 44.72 |
| SimpleMem | 88.12 | 82.40 | 79.87 | 42.09 | 84.64 | 76.46 | 70.39 | 37.90 |
| **TriMem** | **92.46** | **83.23** | **85.72** | **55.79** | **92.17** | **82.28** | **78.17** | **45.01** |

Table 4: **Performance comparison on the LongMemEval benchmark.** We report the LLM-judged correctness score of different memory systems with GPT-4.1-mini. TriMem achieves the best overall performance while exhibiting consistently competitive results across different sub-tasks.

| Method | Temporal | Multi-Session | Knowledge-Update | Single-Session-User | Single-Session-Assistant | Single-Session-Preference | Average |
|---|---|---|---|---|---|---|---|
| Full-context | 27.06% | 30.08% | 41.03% | 47.14% | 32.14% | 60.00% | 39.57% |
| Mem0 | 40.60% | 50.37% | 69.23% | 87.14% | 48.21% | 63.33% | 59.81% |
| LightMem | **85.71%** | 47.37% | **92.30%** | 88.57% | 21.43% | 76.67% | 68.67% |
| SimpleMem | 83.46% | 60.92% | 79.48% | 85.71% | 75.00% | 76.67% | 76.87% |
| **TriMem** | 79.70% | **72.18%** | 91.03% | **94.29%** | **91.07%** | **80.00%** | **82.80%** |

- Our method remains effective on different long-term agent memory benchmarks, demonstrating **the strong generalization ability** of TriMem.

# Rationality of Our Design

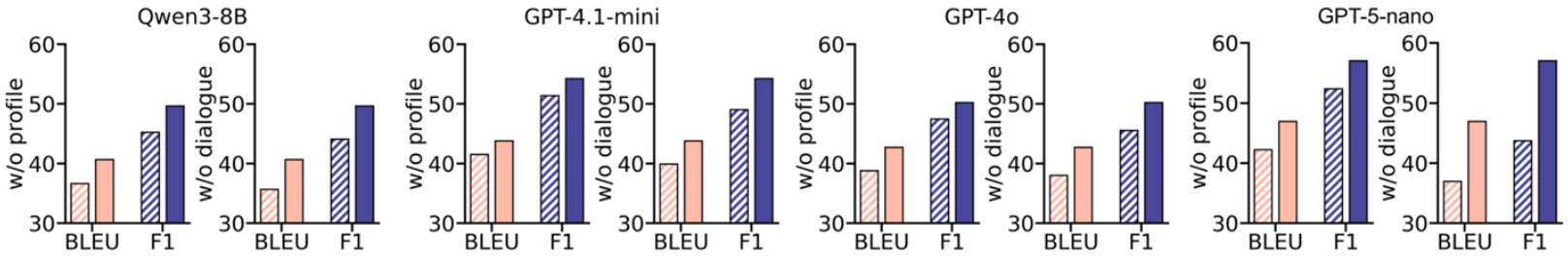



Figure 4: **Ablation of profile and raw dialogue module.** Incorporating entity profile and raw dialogue shows the best results, which demonstrates the rationality of our design.

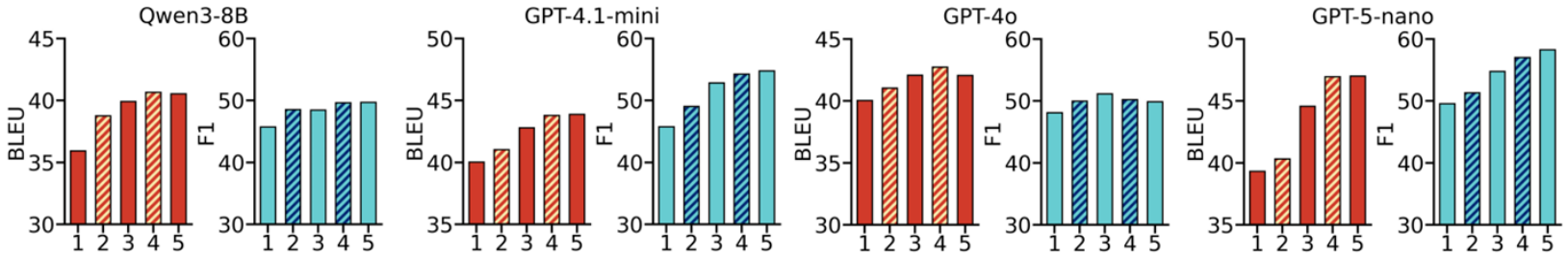



Figure 5: **Impact of varying evolution step.** Appropriate update steps lead to overall improvements, while further updates result in excessive refinement, which negatively affects the performance.

- The agent suffers a noticeable **performance drop** when either component is excluded, demonstrating that incorporating profiles and raw dialogues is a reasonable design
- The performance **gradually improves** with the increase of evolution steps, and reaches the optimal level when the number of steps is set to 4.

# Analysis of Prompt Evolution

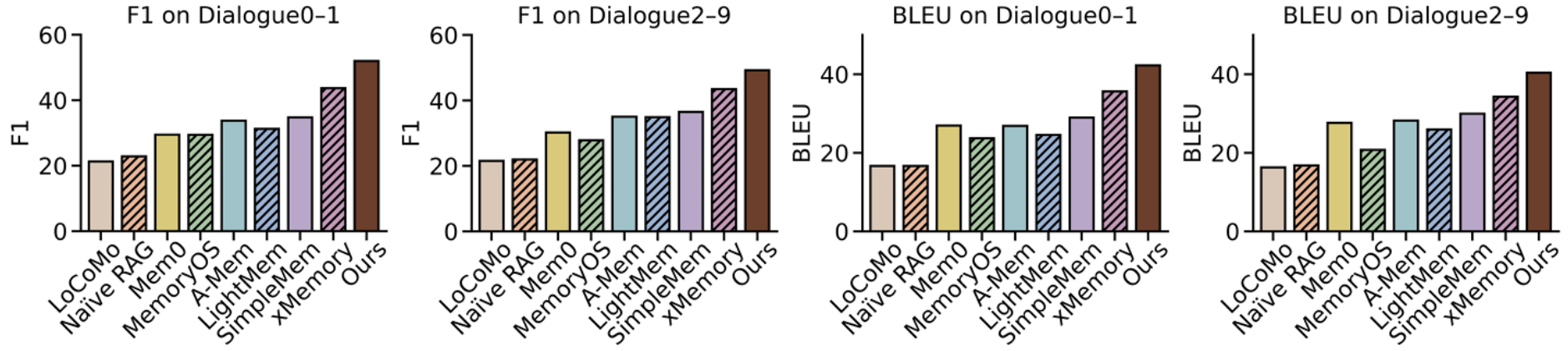
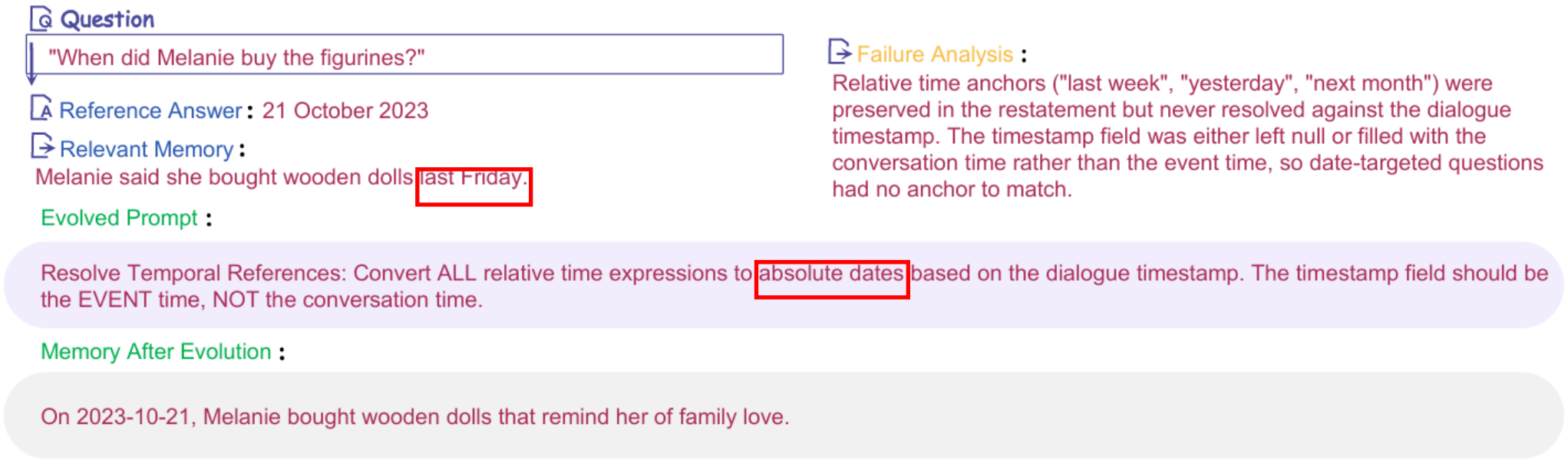



Figure 9: **Details on two subsets.** The memory system can consistently achieve better performance on both subsets, indicating the stability and generalization ability of our prompt evolution method.

- Prompt evolution was conducted exclusively on **dialogue 0** and **dialogue 1** of LoCoMo using Qwen3-8B. To evaluate the generalizability of the selected prompts, we report TriMem's performance with Qwen3-8B on two LoCoMo subsets in Figure 9. TriMem consistently outperforms the baselines on both subsets, indicating that prompt optimization does not overfit to the two development dialogues.

# Analysis of Prompt Evolution




- The failure case in early rounds drove the addition of a tip to the extraction prompt, making the prompts become increasingly detailed

# More Prompt Evolution Examples

**Question**

"What instruments does Melanie play?"

Reference Answer: clarinet and violin

Relevant Memory:
Melanie plays the clarinet, an instrument she started young as a form of self-expression.

Failure Analysis:
Multi-fact dialogues were summarized into the most salient fact, with the remaining facts merged into vague context that lost the queryable nouns. Recall on aggregation questions (instruments, hobbies, languages, pets, locations visited) collapsed even when the dialogue contained every fact.

Evolved Prompt:

Atomic Facts: Extract individual facts as SEPARATE entries. If one dialogue mentions 3 activities, create 3 entries. Do not merge unrelated facts into one summary. "Melanie plays the clarinet, which she started as a child." "Melanie also plays the violin, which she picked up in college."

Memory After Evolution:

Melanie plays the clarinet, which she started as a child.
Melanie also plays the violin, which she picked up in college.

Avoid mixing different pieces of information into the same entry.

**Question**

"What personality traits might Melanie say Caroline has"

Reference Answer: Thoughtful, authentic, driven

Relevant Memory:
Caroline is well-regarded by Melanie as a kind and motivated person.

Failure Analysis:
The profile builder was paraphrasing other-speaker descriptors into a generic register, losing the exact adjectives the other speaker actually used. For character-trait and inter-personal-perception questions in LoCoMo, the gold answers are typically the verbatim adjectives that appear in the dialogue — once paraphrased away, they were unrecoverable from either the profile or the underlying memory entries.

Evolved Prompt:

For [How Others Describe Them]: quote the exact adjectives/phrases other speakers use about this person (e.g. "thoughtful", "driven").

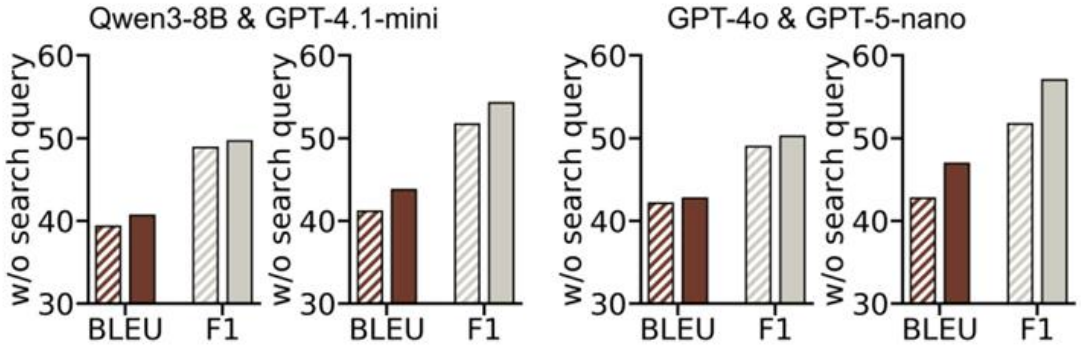
Memory After Evolution:

Melanie has called Caroline 'thoughtful', 'authentic', and 'driven' across multiple sessions.

Record others' evaluations

https://tmlr-trimem.github.io/ & https://github.com/tmlr-group

# Necessity and Visualization of Search Query




**Question**

"What does Caroline's necklace symbolize?"

Target Query:

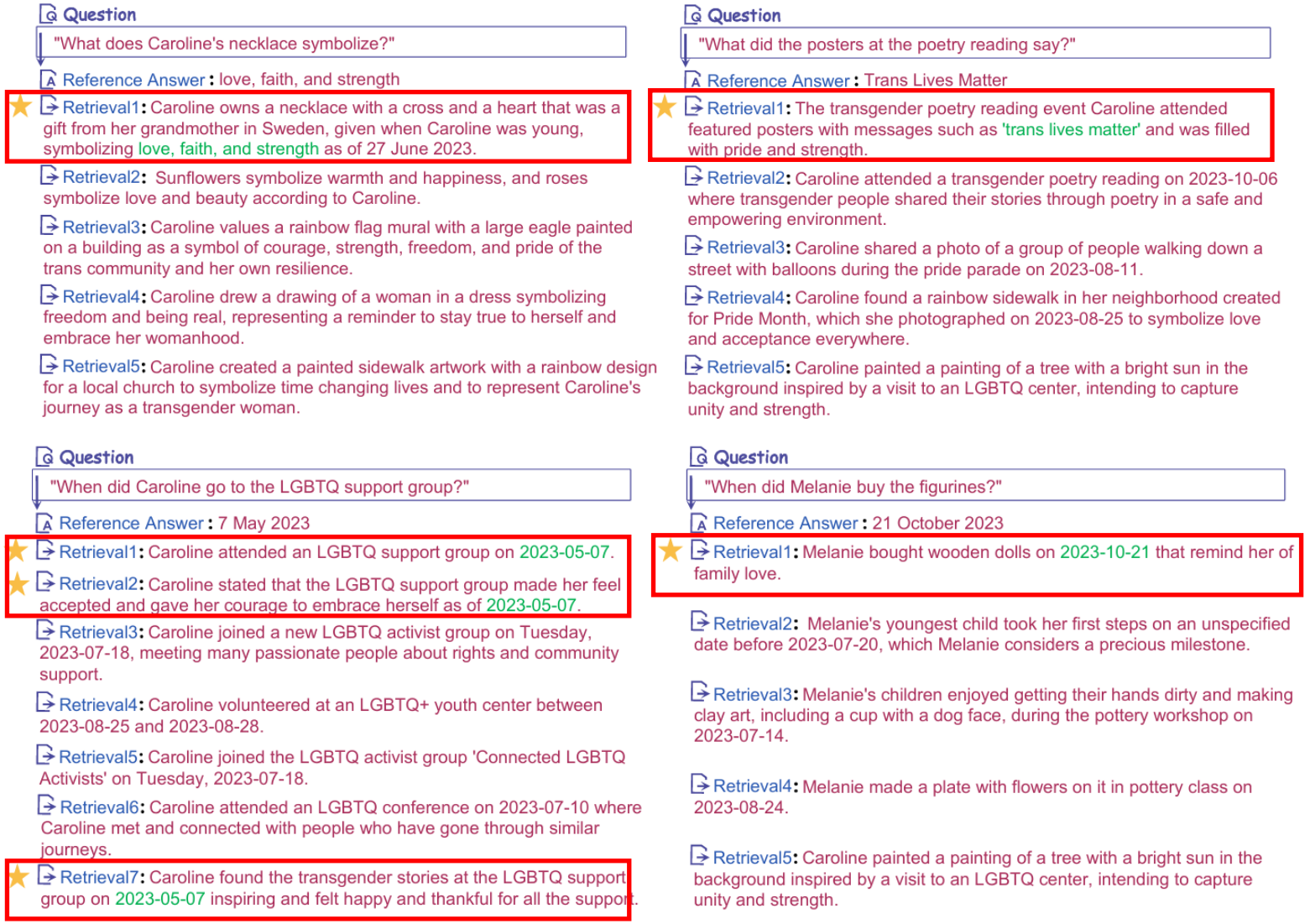
What does Caroline's necklace symbolize?
What is the symbolism of Caroline's necklace?
In what context does Caroline's necklace appear and what is its background?

Key Information:

Caroline, necklace, symbolize, symbolism

- The performance is significantly improved after adding the search query, indicating that the search query helps the model achieve more accurate retrieval, thus confirming the necessity of this module.
- Through the required information analysis, TriMem can break down the problem, enabling a more accurate search.

# Necessity and Visualization of Search Query




- We visualize the content of the retrieval entries, the most relevant entries have been marked with star. It can be seen that our system can precisely retrieve relevant memory to support agent reasoning

https://tmlr-trimem.github.io/ & https://github.com/tmlr-group

# Outline

- Background and Motivation
- System Design of Our Method
- Experimental Results
- Conclusion

# Conclusion

- ✓ Conceptually, we systematically exploring the limitations of existing paradigms spanning storage fidelity, indexing efficiency and reasoning quality.
- ✓ Technically, we propose TriMem, which innovatively introduces entity profile module to avoid shallow reasoning, preserves raw dialogue identifier to avoid detail loss and iteratively optimize system prompts to avoid performance fluctuation.
- ✓ Experimentally, we conduct extensive explorations to verify the effectiveness of TriMem under different scenarios.

arXiv: **arxiv.org/pdf/2605.19952**

Project Page**:** **tmlr-trimem.github.io**

Code: **github.com/tmlr-group/TriMem**

## Resources

arXiv

Project Page

Code

https://tmlr-trimem.github.io/ & https://github.com/tmlr-group