Memory OS of AI Agent

EMNLP 2025

北京邮电大学 & 腾讯 AI LAB

马浩然

2025.9.20

章节介绍

- 作者介绍
- 研究背景与动机
- ARIA 系统设计
- 实验评估
- 总结与讨论

章节介绍

- 作者团队介绍
- 研究背景与动机
- MemoryOS 系统设计
- 实验评估
- 总结与讨论

作者团队介绍

Jiazheng Kang

来自北京邮电大学百家 AI 课题组，导师为白婷

百家 AI 课题组主要研究方向为大语言模型，记忆机制和智能体推荐系统等领域

论文发表：

- [EMNLP'25] Memory OS of AI Agent
- [arXiv'24] BaiJia: A Large-Scale Role-Playing Agent Corpus of Chinese Historical Characters
- [IPM'24] GPR-OPT: A Practical Gaussian optimization criterion for implicit recommender systems.
- [arXiv'24] GMoE: Empowering LLMs Fine-Tuning via MoE Graph Collaboration

章节介绍

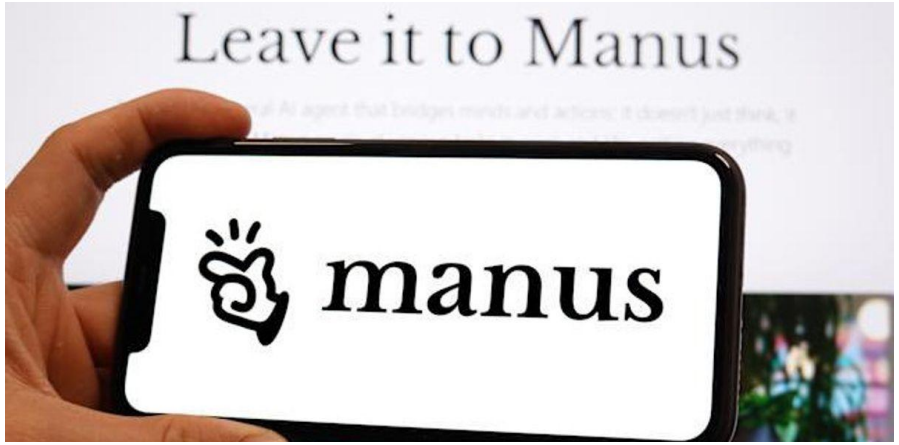
- 作者介绍
- **研究背景与动机**
- MemoryOS 系统设计
- 实验评估
- 总结与讨论

研究背景与动机

从 LLM 到 AI Agnet

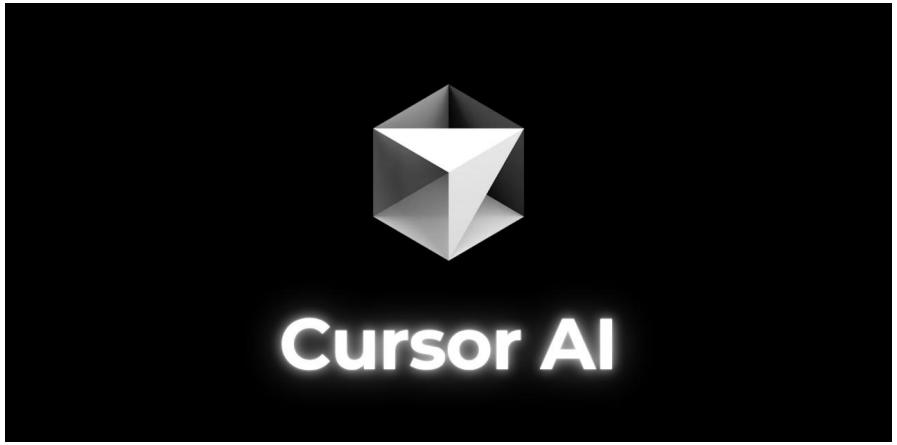
- **LLM 的革命性进展**

近年来，以GPT系列为代表的大语言模型（LLMs）取得了飞速发展，在文本理解、生成和推理方面展现出惊人的能力



- **AI Agent 新范式的广泛应用**

基于LLM强大的“大脑”，AI Agent 应运而生。它们不仅仅是聊天机器人，更是能够自主理解、规划并执行复杂任务的智能实体，如帮助用户规划日程的手机个人助手，完成编程任务的 Cursor

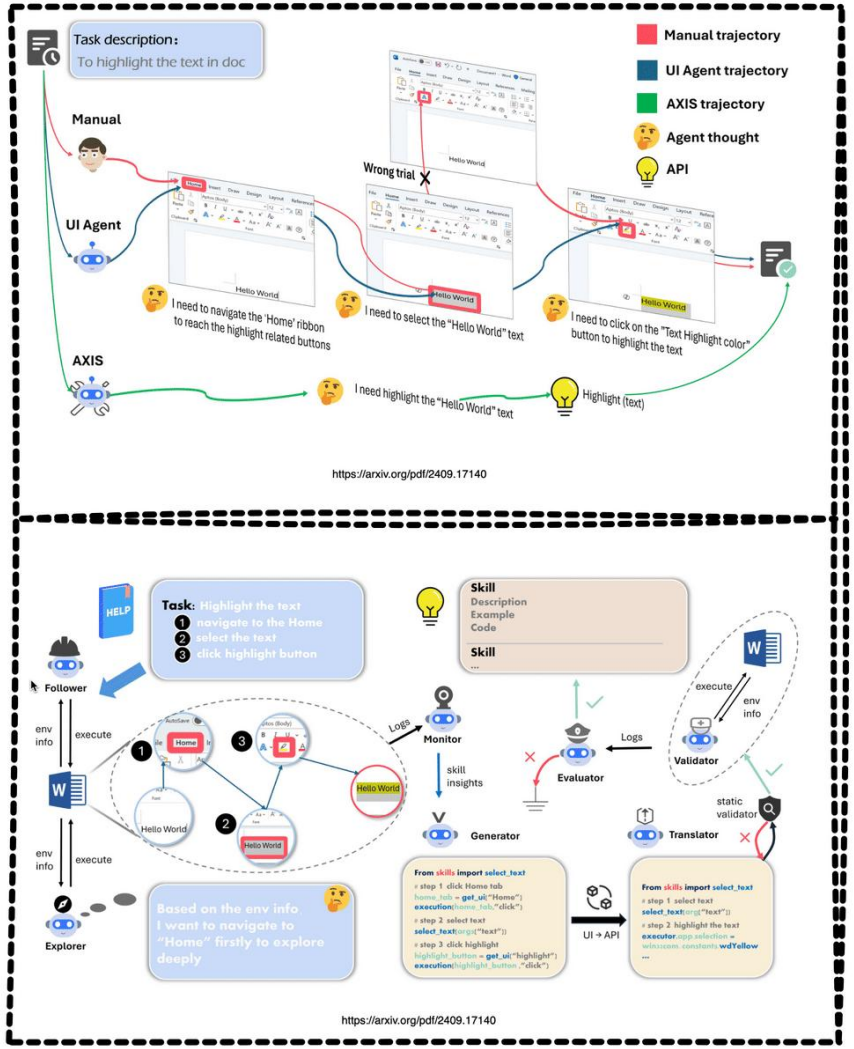


研究背景与动机

新范式伴随新局限

- 当前现状：AI Agent 执行复杂任务时，往往需要与环境（如外部 API、数据库）或用户进行多轮、长期交互。例如，跨天处理项目规划、持续健康监测或多轮客户支持对话

- 当前 LLM 的局限：LLM上下文窗口长度有限（gpt-4o-128k），无法高效存储和检索长期历史信息，导致 Agent 在多轮交互中容易“遗忘”上下文、重复错误或无法保持一致性



研究背景与动机

现有LLM架构难以支持长时程、有状态的 Agent 交互

局限一：LLMs 上下文长度不足

- 信息丢失：在长对话或执行长周期任务时，早期的关键指令、用户偏好或中间结论会被移出上下文窗口，导致 Agent 丧失任务的连续性

- 对话连贯性断裂：模型无法关联时间跨度较大的对话内容，导致前后矛盾或重复提问，严重影响交互质量

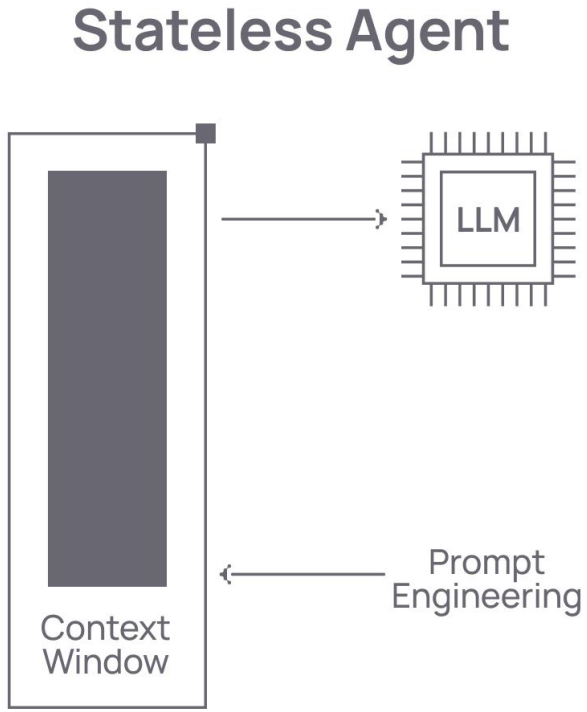
模型	上下文长度
DeepSeek-R1	128K
GPT-4o	128K
Claude 4 Sonnet	200K

研究背景与动机

现有 LLM 架构难以支持长时程、有状态的 Agent 交互

局限二：无状态交互模型，LLMs本质上是无状态的模型本身不具备跨会话（cross-session）保留和积累环境特定知识的能力

- 知识无法沉淀：在多次交互中产生的信息与知识无法被有效存储和在未来任务中复用，降低了 Agent 任务成功率和回答准确性
- 缺乏个性化演进： Agent无法构建和动态更新用户的长期画像，导致其服务始终停留在“一次性”的浅层交互，难以实现个性化



研究背景与动机

现有记忆增强方法的核心思路与局限

核心思想：**存储+检索**

- **A-Mem**：将记忆动态地组织成结构化的“笔记”
- **MemGPT**：模仿操作系统的内存管理
- **Think-in-Memory**：不存储原始对话，而是存储模型推理和思考后产生的思维链
- **MemoryBank - 检索与遗忘机制**：基于“艾宾浩斯遗忘曲线”动态调整记忆的权重

上述工作仅针对独立技术点进行优化，缺乏统一的全局管理

章节介绍

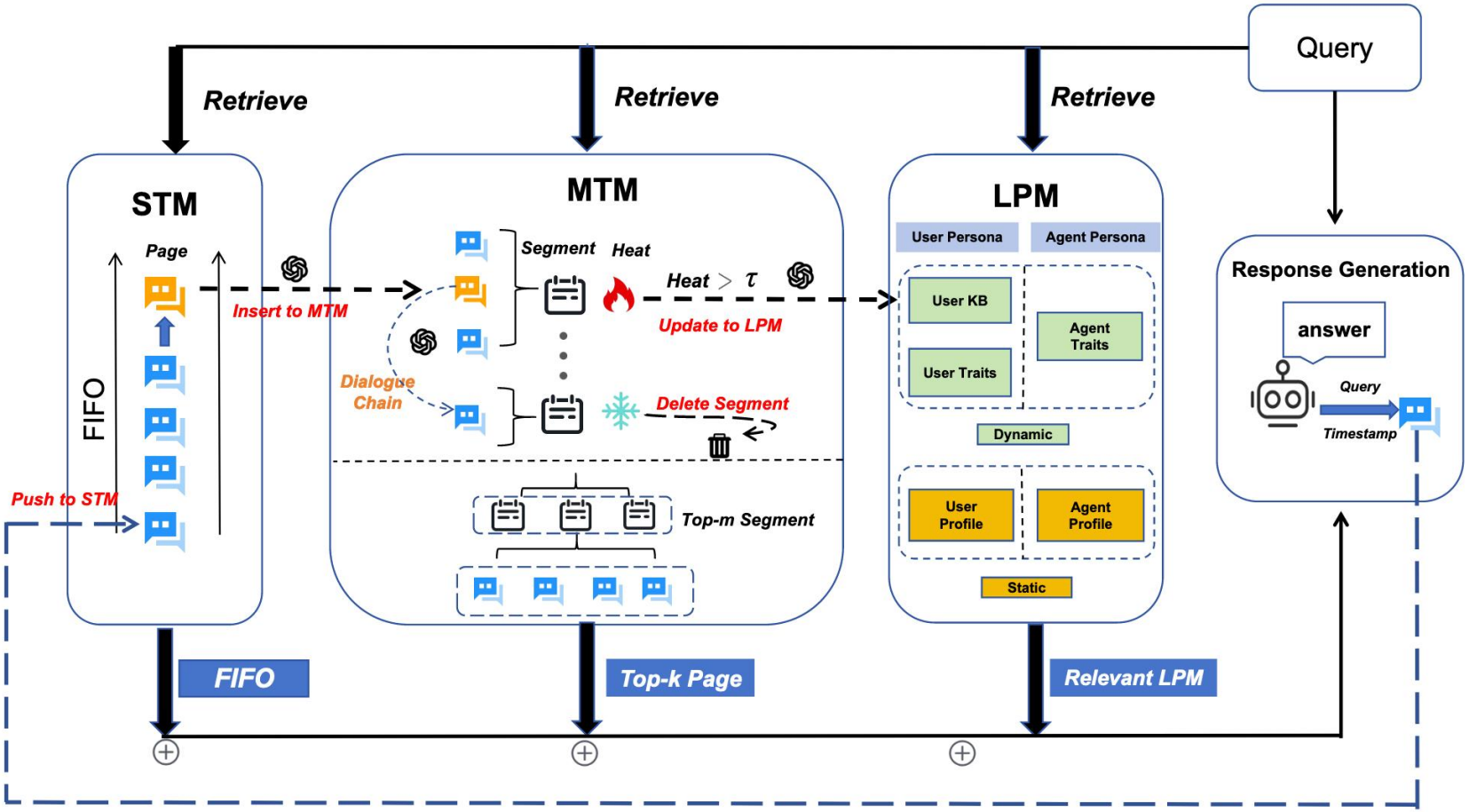
- 作者介绍
- 研究背景与动机
- **MemoryOS 系统设计**
- 实验评估
- 总结与讨论

MemoryOS 系统设计

像操作系统管理计算机资源一样管理 Agent 记忆

- 借鉴计算机操作系统中的内存管理机制（分层、分段、分页、动态更新）来系统性地解决 Agent 的长期记忆问题

- 构建存储、更新、检索和生成 4 个协同模块进行记忆全生命周期管理

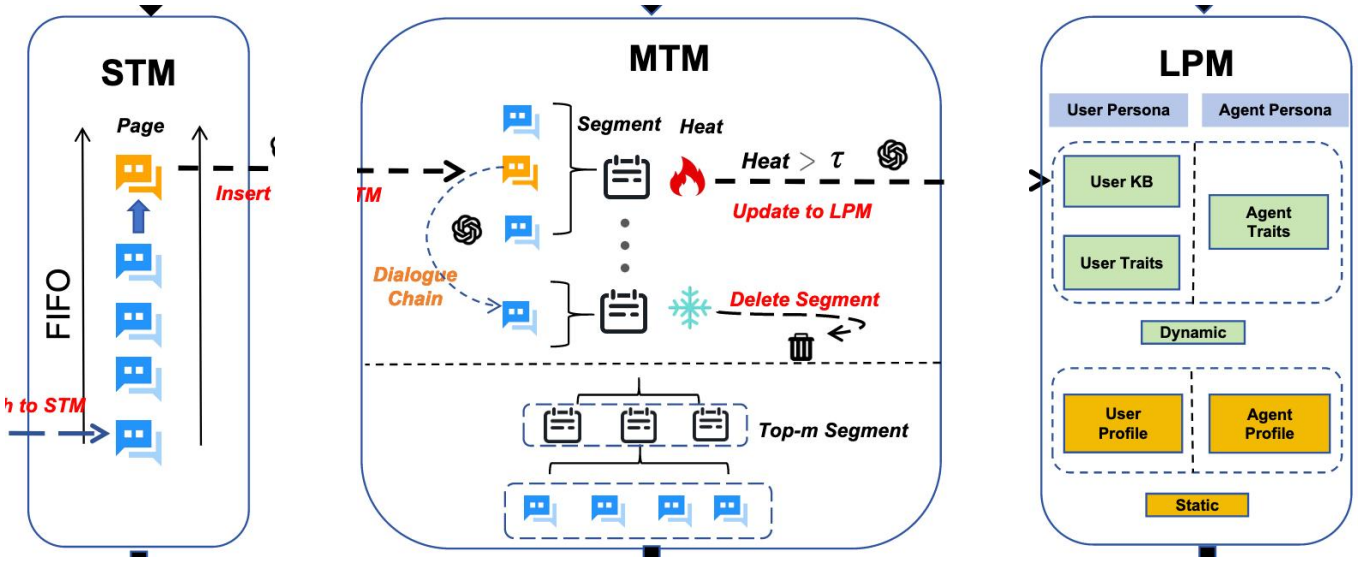


MemoryOS 系统设计

记忆存储模块

为了实现高效的记忆管理，MemoryOS 设计了一个分层式的存储架构，它由三种不同类型的存储单元构成

- 短期记忆 (Short-Term Memory, STM)
- 中期记忆 (Mid-Term Memory, MTM)
- 长期个人化记忆 (Long-term Personal Memory, LPM)



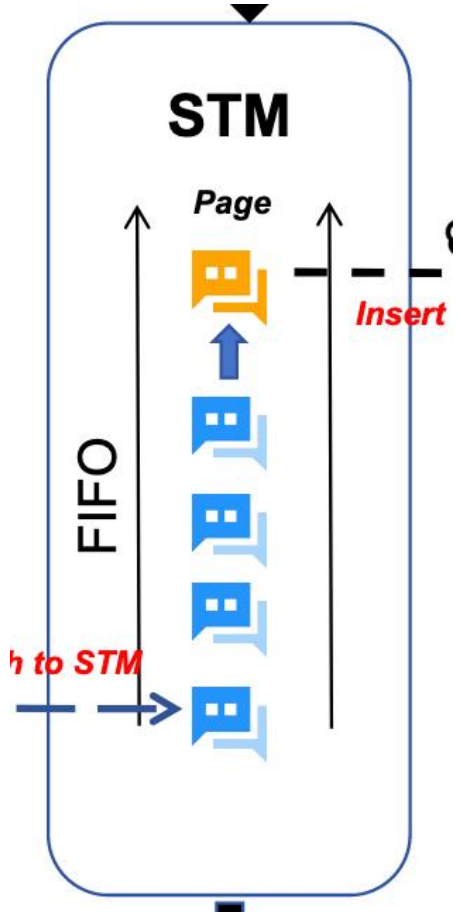
MemoryOS 系统设计

记忆存储模块

短期记忆 (STM)

- **功能:** 负责存储实时的、正在进行的对话数据 。 AI 的“工作内存”或“缓存”
- **存储结构:** 信息被组织成一个个独立的 “对话页”
- **页面结构:** 每个对话页包含三个核心元素：用户的查询、模型的响应和时间戳

- **链接机制**：为了保证对话的上下文连贯性，STM利用 LLM 自动评估新页面与前序页面的语义相关性，每个新页面都会尝试链接到之前的页面构建成一条“对话链”，并为每一个链生成元信息



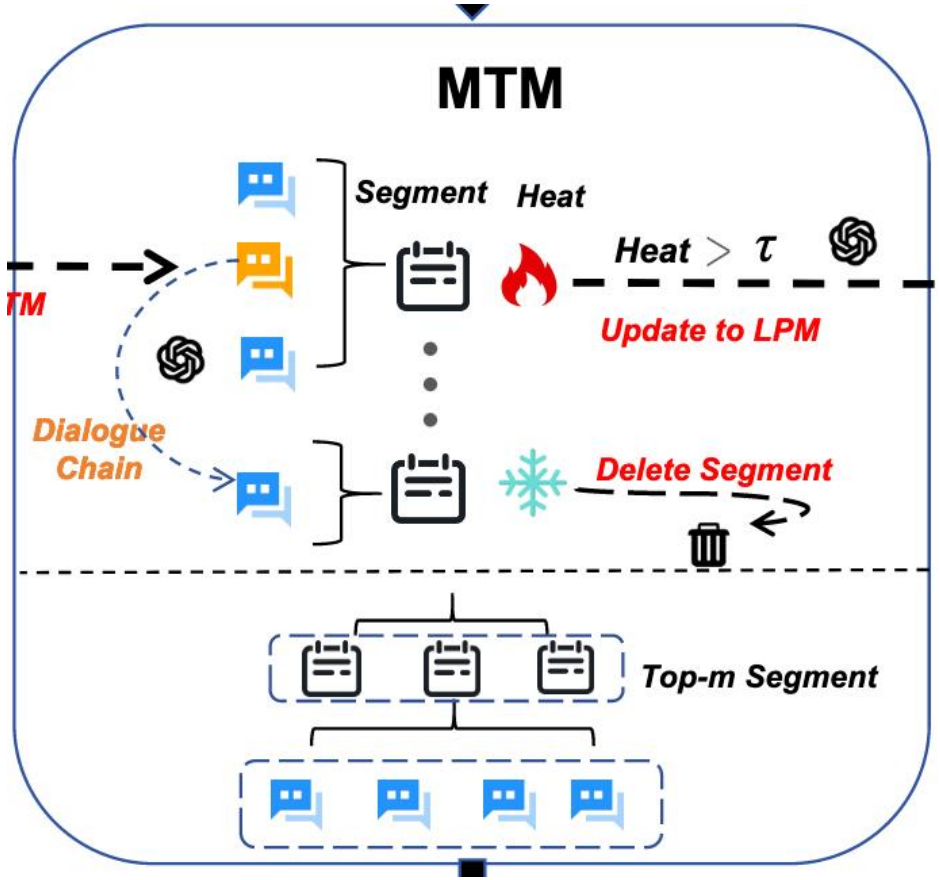
MemoryOS 系统设计

记忆存储模块

中期记忆 (MTM)

- 功能: 将相关对话按 “话题 (Topic)” 进行智能归纳和整理
- 存储结构: 具有相同主题的“对话页 (Pages)”会被自动聚合，形成一个 “段 (Segment)” 每个“段”代表一个独特的话题，并包含多页相关的对话内容

- 聚合机制：一个页面是否属于某个段，取决于它们的相似度分数是否超过阈值，相似度分数综合考虑了语义相似度和关键词相似度



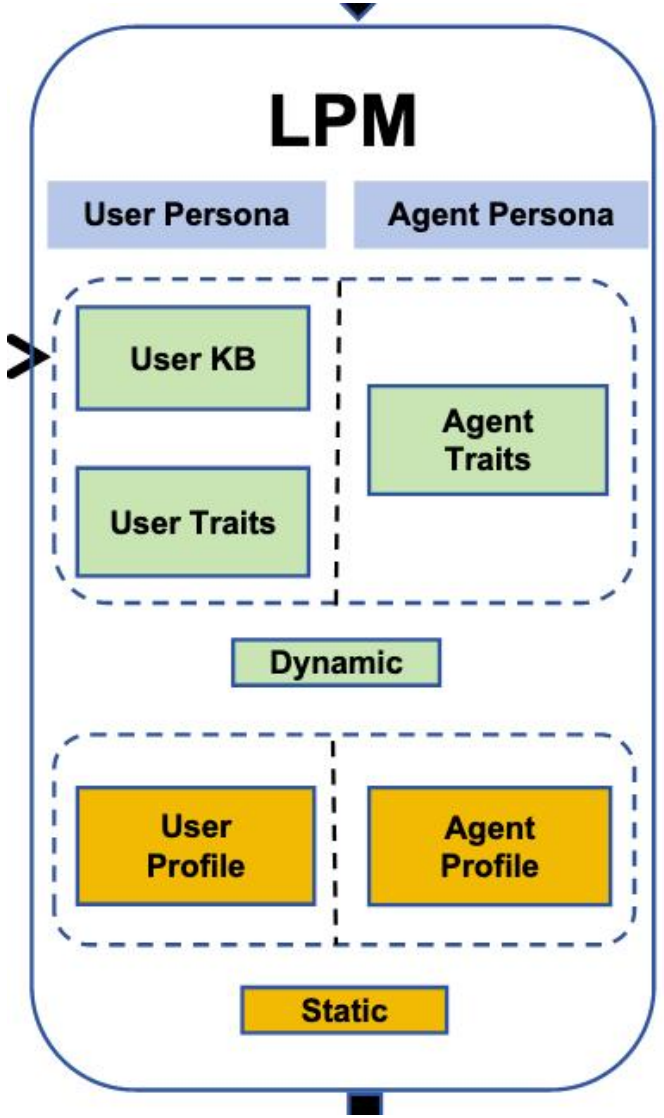
$$F_{score} = cos(e_s, e_p) + F_{Jacard}(K_s, K_p)$$

MemoryOS 系统设计

记忆存储模块

长期个人化记忆 (LPM)

- **功能:** PM 负责确保持久化存储用户和智能体的核心信息，这是实现长期交互一致性与深度个性化的关键
- **用户画像**
  - 用户档案：静态，如用户姓名、年龄等
  - 用户知识库：动态，关于用户的知识性事实
  - 用户特征：动态，对用户兴趣、习惯和偏好的总结
- **Agent 画像**
  - Agent 档案：静态，Agent 自身固定的角色设定
  - Agent 特征：动态，交互过程中的动态记忆



MemoryOS 系统设计

记忆更新模块

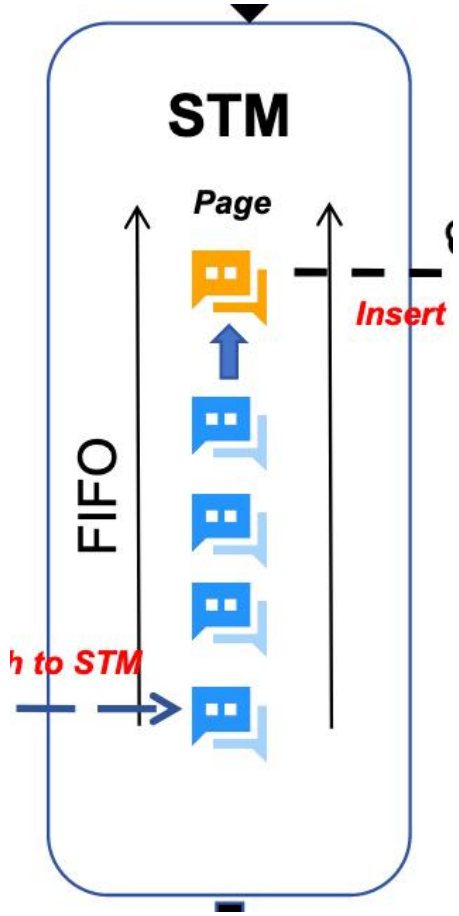
STM 到 MTM - “先进先出”的信息流

1. 短期记忆 (STM) 是一个长度固定的队列，用于存放最近的对话页

2. 每当有新的对话页产生时，它会被添加到队列的末尾

3. 如果此时队列已满，最旧的（即最早进入的）那个对话页就会被从STM中移出

4. 这个被移出的页面并不会被丢弃，而是被传输到中期记忆 (MTM) 中，等待后续的话题归类和整理



MemoryOS 系统设计

记忆更新模块

MTM 到 LPM - 基于“热度”的智能晋升与淘汰

引入“热度” 评分机制来管理 MTM 中的话题“段 ”，热度公式由三个核心指标加权决定

$$Heat = \alpha \cdot N_{visit} + \beta \cdot L_{interaction} + \gamma \cdot R_{recency}$$

指标	含义	解释
N	访问频率	该话题段被检索和访问的次数，频繁被访问意味着它很重要
L	交互深度	该话题段内包含的对话页总数，页数越多意味着围绕该话题的讨论越深入
R	近期性	距离上一次访问该话题段的时间有多近，这是一个时间衰减系数，最近访问的更重要

MemoryOS 系统设计

记忆更新模块

MTM 到 LPM - 基于“热度”的智能晋升与淘汰

基于热度的两大操作

- 晋升至 LPM：当一个话题段的热度超过预设阈值（例如，热度 > 5）时，系统会认为这是一个非常重要且稳定的用户兴趣点，该段内的信息会被提炼、总结，并更新到长期个人化记忆 (LPM) 的用户知识库 (User KB) 和用户特征 (User Traits) 中，更新后，该段的交互深度会被重置为零，使其热度自然下降，避免信息冗余

- 淘汰：MTM的容量是有限的，当容量达到上限时，系统会自动淘汰（删除）热度最低的话题段

MemoryOS 系统设计

记忆检索与生成模块

精准唤醒与生成：在需要时找到对的记忆

当用户发起一个新的查询 (Query) 时，MemoryOS 会启动一个协同工作流程，从三层记忆中精准地“唤醒”相关信息，并生成最终响应

并行检索三层记忆

- STM：全部检索，获取所有对话页
- MTM：两阶段检索，首先根据语义检索 top-m 个段，再进一步检索出 top-k 个页
- LTM：从动态的用户知识库和 Agent 特征中根据语义检索 top-k 个条目，全部检索静态的用户档案和 Agent 档案

- 最后会所有的记忆连同用户原始查询拼接并送入 LLM 进行响应生成

章节介绍

- 作者介绍
- 研究背景与动机
- MemoryOS 系统设计
- **实验评估**
- 总结与讨论

实验评估

评估实验设置

数据集

- GVD: 一个模拟15个虚拟用户与助手进行10天交互的多轮对话数据集，覆盖多个话题
- LoCoMo: 专为评估超长对话记忆能力而设计的基准，平均对话轮数高达300轮，包含约9000个Token

评估指标

- GVD：记忆检索准确率，回答正确性，上下文连贯性； （DeepSeek R1模型打分）
- LoCoMo：F1 Score 和 BLEU-1

对比基线

- Think in Memory, MemoryBank, MemGPT, A-MEM

实验评估

GVD 数据集表现

在 GVD 数据集上，所有方法的表现都相对较好，但 MemoryOS 仍然在所有指标上都取得了最优成绩

相较于表现次优的 A-Mem 模型，在使用 GPT-4o-mini 时，MemoryOS 方法在准确率(Acc.)上提升了 3.2%，在正确性(Corr.)上提升了 5.4%

Model	Method	Acc. ↑	Corr. ↑	Cohe.↑
GPT-4o-mini	TiM	84.5	78.8	90.8
	MemoryBank	78.4	73.3	91.2
	MemGPT	87.9	83.2	89.6
	A-Mem	90.4	86.5	91.4
	Ours	**93.3**	**91.2**	**92.3**
Improvement (%)		3.2% ↑	5.4% ↑	1.0% ↑
Qwen2.5-7B	TiM	82.2	73.2	85.5
	MemoryBank	76.3	70.3	82.7
	MemGPT	85.1	80.2	86.9
	A-Mem	87.2	79.5	87.8
	Ours	**91.8**	**82.3**	**90.5**
Improvement (%)		5.3% ↑	3.5% ↑	3.1% ↑

实验评估

LoCoMo 数据集表现

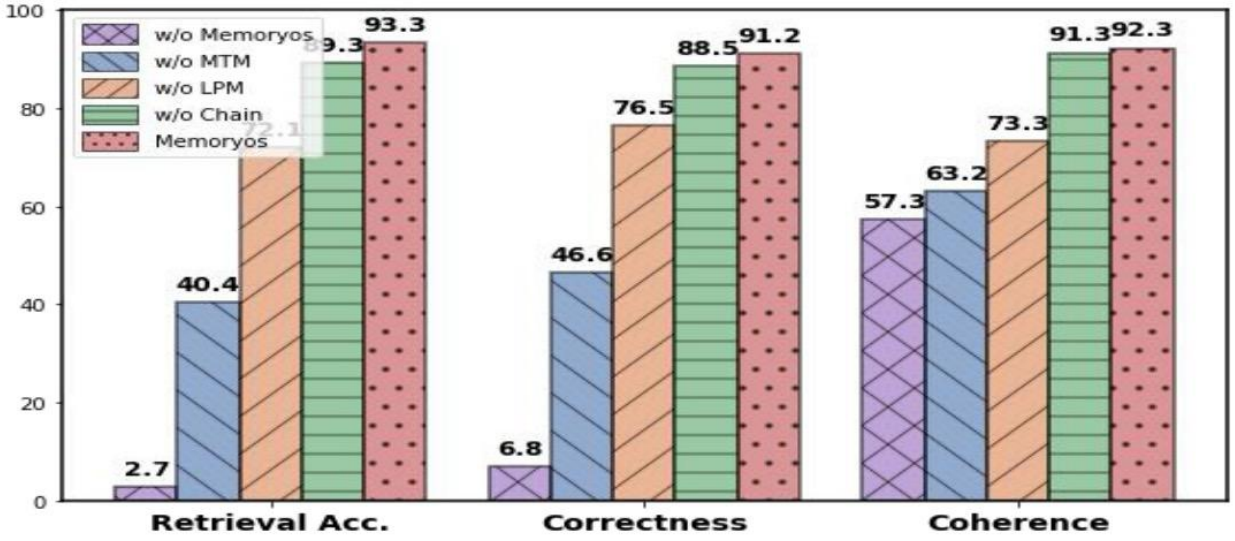
- 在更能体现长期记忆能力的 LoCoMo 基准上，MemoryOS 的优势被进一步放大
- 使用 GPT-4o-mini 时，MemoryOS 相较于基线模型，平均 F1 分数提升了 49.11%，平均 BLEU-1 分数提升了 46.18%

Model	Method	Single Hop		Multi Hop		Temporal		Open Domain		Avg. Rank ↓	Avg. Rank ↓
		F1 ↑	BLEU-1 ↑	F1 ↑	BLEU-1 ↑	F1 ↑	BLEU-1 ↑	F1 ↑	BLEU-1 ↑	(F1)	(BLEU-1)
GPT-4o-mini	TiM	16.25	13.12	18.43	17.35	8.35	7.32	23.74	22.05	3.8	4.0
	MemoryBank	5.00	4.77	9.68	6.99	5.56	5.94	6.61	5.16	5.0	5.0
	MemGPT	26.65	17.72	25.52	19.44	9.15	7.44	41.04	34.34	2.2	2.5
	A-Mem	27.02	20.09	45.85	36.67	12.14	12.00	44.65	37.06	–	–
	A-Mem*	22.61	15.25	33.23	29.11	8.04	7.81	34.13	27.73	3.0	2.5
	Ours	**35.27**	**25.22**	**41.15**	**30.76**	**20.02**	**16.52**	**48.62**	**42.99**	**1.0**	**1.0**
Improvement (%)		32.35%↑	42.33%↑	23.83%↑	5.67%↑	118.80%↑	111.52%↑	18.47%↑	25.19%↑	–	–
Qwen2.5-3B	TiM	4.37	5.01	2.54	3.21	6.20	5.37	6.35	7.34	4.3	3.5
	MemoryBank	3.60	3.39	1.72	1.97	6.63	6.58	4.11	3.32	4.8	4.8
	MemGPT	5.07	4.31	2.94	2.95	7.04	7.10	7.26	5.52	2.8	3.8
	A-Mem	12.57	9.01	27.59	25.07	7.12	7.28	17.23	13.12	–	–
	A-Mem*	10.31	8.76	16.31	11.07	6.94	7.31	12.34	10.62	2.3	2.0
	Ours	**23.26**	**15.39**	**21.44**	**14.95**	**10.18**	**8.18**	**26.23**	**22.39**	**1.0**	**1.0**
Improvement (%)		125.61% ↑	75.68% ↑	31.45% ↑	35.05% ↑	46.69% ↑	11.90% ↑	112.56% ↑	110.83% ↑	–	–

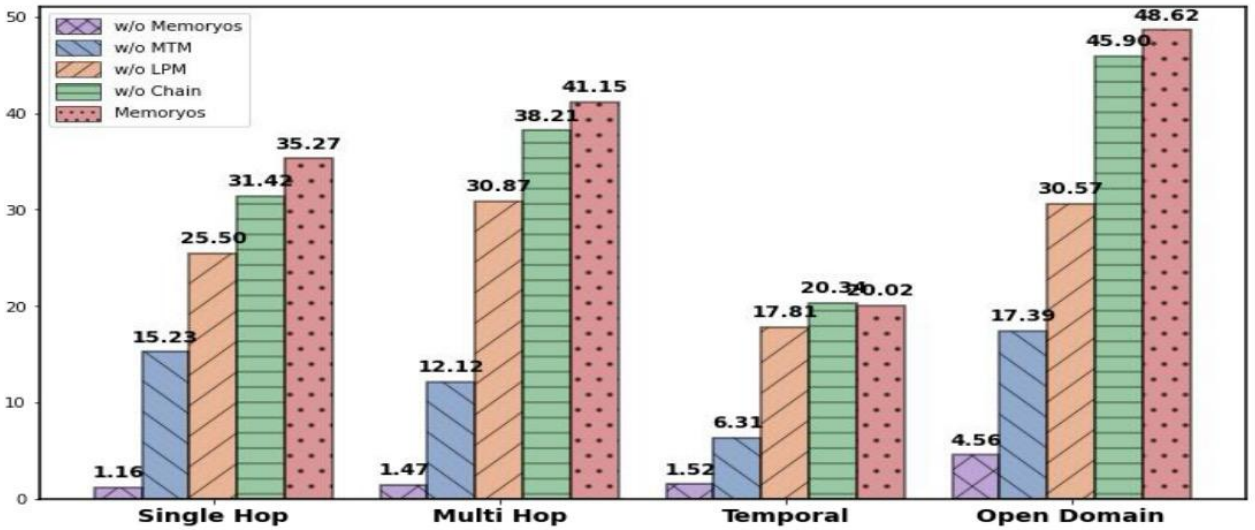
实验评估

消融实验验证

- 中期记忆 (MTM) 的影响最为显著
- 长期个人化记忆 (LPM) 的贡献位居第二
- 对话链 (Chain) 机制贡献影响相对最小



(a)Ablation results on the GVD dataset with GPT-4o-mini



(b)Ablation results on the LoCoMo dataset with GPT-4o-mini using the F1 score.

章节介绍

- 作者介绍
- 研究背景与动机
- MemoryOS 系统设计
- 实验评估
- **总结与讨论**

总结与讨论

MemoryOS 总结

动机: 当前的大语言模型（LLMs）因其固定的上下文窗口和无状态特性，在 Agent 场景下处理长对话时面临严峻挑战，导致缺乏长期记忆和个性化能力

方案: 通过一个分层式存储架构（STM, MTM, LPM），并结合了分段分页的存储策略与热度驱动的动态更新、检索机制，实现了对记忆的系统化、自动化管理

自己工作的思考

当前有关 Agent Memory 仅考虑从文本的形式上去存储管理记忆，忽略了关联的 LLMs 内部计算缓存的形式，如 KV Cache；如何从计算缓存的角度考虑 Memory 的管理与复用是我们关注的问题