

大语言模型时代的内存：统一框架下的模块化架构与策略 [实验、分析 & 基准]

Yanchen

Wu*

香港中文大学
(深圳)

Tenghui

Lin*

CUHK

Yingli

Zhou

CUHK-Shenzhen

Fangyuan

Zhang

HITSZ

Qintian

Guo

BIT

Xun Zhou

HITSZ

Sibo Wang

CUHK

Xilin Liu

Huawei Cloud

Yuchi Ma

Huawei Cloud

Yixiang

Fang

CUHK-Shenzhen

摘要

记忆在基于大语言模型（LLM）的智能体执行长周期复杂任务（如多轮对话、游戏对战、科学发现）中成为核心模块，其中记忆能够实现知识积累、迭代推理和自我演化。文献中已提出多种记忆方法，但这些方法尚未在相同实验情景下进行系统且全面的对比。本文首先从高层次视角总结了一个统一框架，涵盖所有现有的智能体记忆方法。随后，我们在两个知名基准上广泛比较了代表性智能体记忆方法，并深入分析了各类方法的有效性。作为实验分析的副产品，我们还通过整合现有方法中的模块设计了一种新型记忆方法，其性能优于当前最先进的方法。最后，基于这些发现，我们提出了具有前景的未来研究方向。我们相信，对现有方法行为的更深入理解，将为未来研究提供宝贵的全新洞见。

PVLDB 参考文献格式：

Yanchen Wu, Tenghui Lin, Yingli Zhou, Fangyuan Zhang, Qintian Guo, Xun Zhou, Sibow Wang, Xilin Liu, Yuchi Ma, and Yixiang Fang. 大语言模型时代的内存：统一框架下的模块化架构与策略 [实验、分析 & 基准]. PVLDB, 19(1): XXX-XXX, 2026. doi:XX.XX/XXX.XX

PVLDB Artifact Availability:

The source code, data, and/or other artifacts have been made available at <https://github.com/Yanchen398/Memory-in-the-LLM-Era>.

*前两位作者对本研究的贡献相同。

本作品采用知识共享署名-非商业性使用-禁止演绎 4.0 国际许可协议。访问 <https://creativecommons.org/licenses/by-nc-nd/4.0/> 以查看该许可证的副本。对于超出此许可证范围的任何使用，请通过电子邮件 info@vldb.org 获取授权。版权由版权所有人/作者持有。出版权已授予 VLDB 基金会。VLDB 期刊论文集，第 19 卷，第 1 期，ISSN 2150-8097。doi:XX.XX/XXX.XX

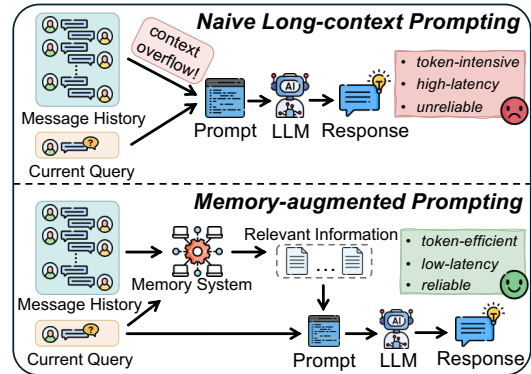


图 1: 朴素长上下文提示与记忆增强型提示概述。

1 引言

大语言模型（LLMs）如 GPT-5 [79]、Qwen3 [96] 以及 Claude Sonnet 4.6 [3] 的发展，正在人工智能领域引发一场革命 [19, 29, 43, 51, 63, 86, 88, 102]。在这一成功的基础上，基于大模型的智能体迅速兴起，并被广泛部署于从工业自动化到个人辅助的众多领域。例如，用于软件工程任务的 SWE-agent 系统 [98] 以及像 Open-Claw [64] 这样的个人助理智能体，展示了基于大模型的智能体如何自主地规划、推理并执行复杂的多步骤工作流。这些智能体正日益被期望能够自主运行，适应多样化的环境，并支持针对用户需求量身定制的个性化交互。支撑这种智能行为的一个关键能力是记忆机制 [67, 101]。如图 1 所示，记忆机制使智能体能够超越简单的长上下文提示，通过保留并利用过往交互中的相关信息实现持续学习。通过为智能体配备记忆机制，它们可以随时间积累经验，保持上下文知识，并做出更为明智的决策——这类似于人类依靠记忆从过往经历中学习并指导未来行动的方式。

近年来，越来越多的记忆方法被提出，以增强基于大语言模型（LLM）的智能体在交互过程中保留、组

表 1: 代表性智能体记忆方法的分类。

Method	Information Extraction	Management Operations	Storage Structure	Retrieval Mechanism
A-MEM [93]	Direct archive, Summarization-based extract	Connect, Update	Flat, Vector	Vector-Based
MemoryBank [103]	Direct archive	Integrate, Update, Filter	Flat, Vector	Vector-Based
MemGPT [66]	Direct archive	Integrate, Transform, Update	Hierarchical, Vector	Lexical-Based, Vector-Based
Mem0 [11]	Direct archive, Summarization-based extract	Integrate, Update, Filter	Flat, Vector	Vector-Based
Mem0 ^g [11]	Graph-based extract	Connect, Update, Filter	Flat, Graph	Vector-Based, Structure-Based
MemoChat [56]	Direct archive	Integrate	Flat	LLM-Assisted
Zep [72]	Direct archive, Graph-based extract	Connect, Transform, Update	Hierarchical, Graph	Lexical-Based, Vector-Based, Structure-Based
MemTree [73]	Direct archive	Connect, Integrate, Update	Flat, Tree	Vector-Based
MemoryOS [32]	Direct archive	Connect, Integrate, Transform, Update, Filter	Hierarchical, Vector	Lexical-Based, Vector-Based
MemOS [44]	Direct archive, Summarization-based extract	Connect, Integrate, Update	Hierarchical, Tree	Lexical-Based, Vector-Based

组织和利用历史信息的能力。这些方法旨在使智能体摆脱无状态推理的局限，进而支持长期规划、个性化以及自适应决策。

受到无状态 LLM 智能体的局限性以及对持续上下文推理需求日益增长的推动，来自多个领域的研究人员——包括数据库、数据挖掘、机器学习和自然语言处理——开始致力于开发高效且可扩展的记忆机制，以服务于智能体 [66, 67, 73, 87, 90]。

在表 1 中，我们总结了十种具有代表性的智能体记忆方法。我们根据四个关键维度对其进行分类：潜在存储结构、信息提取机制、记忆管理策略以及检索方法。

经过仔细的文献调研，我们得出以下观察结果。首先，缺乏一个统一的框架来抽象化并系统分析智能体记忆方法。其次，大多数先前的研究仅报告整体性能结果，却很少深入探讨这些方法中各个组件的作用与影响。第三，针对不同方法之间的全面且系统的比较——尤其是关于准确率和效率方面的对比——仍然不足。

我们的工作。我们通过提出一个统一的、模块化的框架，并对代表性智能体记忆方法进行深入的实验研究，解决了这些不足。该框架将记忆机制分解为四个阶段，包括①信息提取，②记忆管理，③内存存储，和④信息检索。在此框架下，我们在两个典型的长期对话基准——LOCOMO [57] 和 LONGMEMEVAL [91] 上对比了代表性方法。除了整体性能外，我们还分析了实际鲁棒性维度，包括上下文可扩展性和位置敏感性。基于这些分析，我们进一步设计了一种新的智能体记忆方法，该方法在性能和成本效率方面均取得了最佳表现。

综上所述，我们的主要贡献列于下文：

- 我们提出一个统一框架，将典型的智能体记忆方法分解为四个模块化组件，从而实现对其差异的系统性比较。
- 我们对 LOCOMO 和 LONGMEMEVAL 进行了全面的实验研究，并对其 token 成本、上下文可扩展性、证据位置敏感性以及大模型主干依赖性进行了分析。

- 基于上述分析，我们提出了一种新的智能体记忆方法，该方法达到了当前最先进的性能。我们进一步得出了若干关键洞察，并指出了未来研究的有前景方向。

路线图。第 2 节介绍预备知识。第 3 节提出统一框架。第 4–7 节刻画该框架内的代表性设计选择。第 8 节报告实验结果与分析。第 9 节总结经验与机遇。第 10 节回顾相关工作，第 11 节结论。

2 初步研究

在本节中，我们将介绍大语言模型时代现有记忆方法的一些重要概念和典型工作流。同时，还将探讨 RAG 与记忆之间的关系。

2.1 大语言模型相关概念

我们将在下面介绍两个与大语言模型相关的基本概念。

大型语言模型提示。提示 [6, 12, 54] 通过构建一个包含任务指令、当前输入以及可选的示范或辅助证据的输入上下文（提示），来指定大语言模型的任务。模型的输出随后基于此上下文生成。

提示在基于大语言模型的系统中尤为重要，因为它提供了一种轻量级、无需训练的接口以实现任务适配 [54]。通过修改输入上下文即可引导模型行为，而无需更改模型参数。这一特性使得在实际应用中可以直接用自然语言表达任务规范和约束 [1, 65]。

除了最终响应生成之外，提示还常用于驱动以大语言模型为核心的流水线中的中间子任务 [4, 33, 76, 99]。典型例子包括提取关键信息、整合中间结果等。

基于大模型的智能体。基于大语言模型的智能体将大语言模型作为核心决策模型，用于序列动作选择 [40, 78, 99]。与单轮提示不同，智能体在闭环中运行：它接收观测，进行推理或规划，执行动作（可能通过工具），从环境中获取反馈，然后进入下一步 [97]。交互历史和可用状态被合并为文本上下文，智能体基于此预测下一步动作。

基于大语言模型的智能体的动作空间通常包括自然语言回复和结构化工具调用 [44, 66]（例如信息搜索、API 调用以及内存读写操作），这些能力使得智能体能够完成多步骤任务，而不仅仅是单次生成。因此，基于大语言模型的智能体必须在对话轮次和会话之间保留并重用信息，这推动了高效记忆机制的发展。

2.2 智能体记忆

记忆被引入基于大语言模型的智能体，以弥补上下文窗口的限制 [26, 52, 57]。由于模型仅依赖有限数量的 token，早期对话中产生的、超出当前提示范围的信息很容易丢失，这会降低长时对话和多会话任务中的性能 [77, 91, 94]。显式的记忆模块 [5, 21, 66, 67, 83] 使系统能够持久保存交互生成的信息——如用户偏好、关键事件、中间决策以及任务约束——并在相关时重新引入这些信息，从而提升一致性，并支持依赖长期上下文的推理 [21, 67]。

记忆增强型系统的工作流通常始于从持续的交互中选择性地提取重要信息——例如事实、用户偏好或重要事件——并将它们作为记忆条目存储。这些条目随后经过一系列管理操作：[14, 70, 103]（整合），将相似的记忆进行融合，以提高一致性并减少冗余；[27, 53, 66]（更新），修改存储内容以保持准确率并反映最新知识；[41, 67, 103]（过滤），移除过时、冗余或低效用的记忆，以维持系统的效率和相关性；以及 [28, 84]（增强），标记或突出显示重要记忆，以便于识别和检索。这一结构化流程确保了记忆系统保持有序、可扩展，并与持续的用户需求保持一致。当推理或生成需要额外上下文时，记忆系统会检索并提供最相关的信息给大语言模型 (LLM)，从而支持短期适应性和长期连续性，贯穿不断演化的交互过程。

记忆与检索增强生成 (RAG) [18, 23, 36, 87] 是相关但不同的机制。记忆主要针对随时间演变、依赖交互的状态信息，这些信息对于个性化和跨会话连续性至关重要 [66, 103]。相比之下，RAG 主要针对外部知识的锚定，从文档集合或知识库中检索证据，以补充领域知识并减少幻觉 [18, 36]。在实际应用中，它们是互补的：记忆提供用户和会话特定的上下文，而 RAG 从外部语料库中提供与任务相关的事实证据。

3 统一框架

在本节中，我们根据统一框架将现有的智能体记忆系统分解为模块化组件，如图 2 所示。该框架包含四个关键组件：① 信息提取，② 记忆管理，③ 记忆存储，以及④ 信息检索。这些组件共同描述了现有智能体记忆系统在实际中的运作方式。

给定当前用户消息 \mathcal{M} 以及现有的记忆 \mathcal{H} ，智能体系统通过四个关键组件运行：① 信息提取。此组件

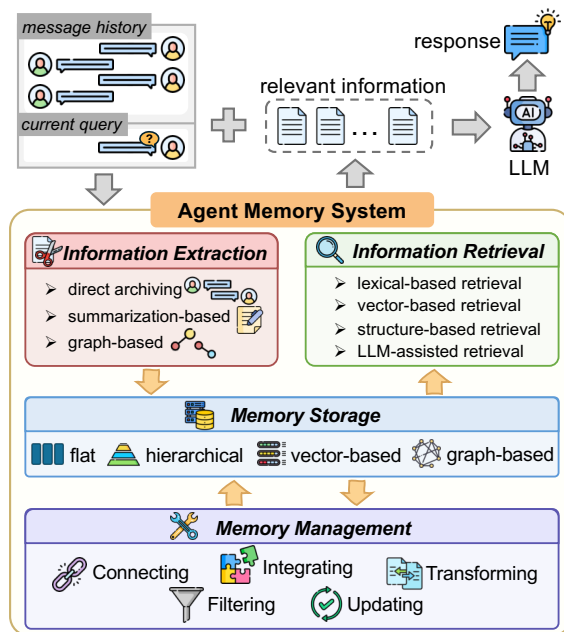


图 2: 智能体记忆系统统一框架概述。

描述了智能体系统如何从 \mathcal{M} 中识别并提取对更新记忆有用的关键信息。它会过滤掉冗余细节，并将相关内容转换为适合下游处理的不同类型知识（例如，从文本中提取的三元组、信息摘要）。② 记忆管理。此组件展示了智能体系统如何通过合并、更新、过滤和增强等操作，将新提取的信息与现有记忆 \mathcal{H} 进行整合。目标是保持记忆状态的一致性、连贯性和时效性，准确反映累积的知识。③ 记忆存储。此组件指定了智能体系统如何组织和持久化处理后的记忆。根据系统设计，可能采用向量-based、基于图形、或混合存储格式，以满足多样化的信息检索需求。④ 信息检索。当接收到新查询时，此组件决定了智能体系统如何从 \mathcal{H} 中检索最相关的信息，以支持推理或响应生成。上述四个组件捕捉了不同智能体记忆方法之间的关键功能差异。

4 信息提取

该组件旨在识别并提取来自 \mathcal{M} 中对下游记忆处理既有用又必要的信息。如图 4 所示，现有的智能体系统采用了不同的信息提取方法，大致可分为以下几类：

① 直接归档。该方法代表了信息提取最直接的形式，智能体系统仅对原始消息和时间戳进行归档，不进行任何处理。

② 基于摘要的提取。该方法利用大模型从一个或多个对话轮次中生成简洁的信息摘要。记忆方法如 A-MEM 和 MemO 从 \mathcal{M} 提取关键词和上下文标签，或提示大模型生成原始文本的抽象摘要。图 3 展示了用于此提取方法的一个代表性提示。

Prompt for Summarization-based Extraction.

提取关键信息，识别核心主题与上下文要素，并分配相关类别标签。

Format the response as a JSON object:

```
{
  "summary": "<one-sentence gist>",
  " 关键词": [
    <several keywords capturing key
      concepts>
  ],
  " 标签": [
    <several broad themes for
      classification>
  ]
}
```

分析信息：

消息

图 3: 一个基于摘要提取的样本提示。

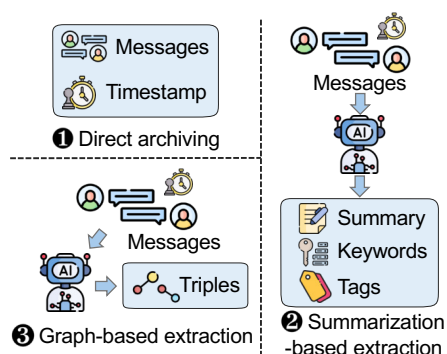


图 4: 信息提取方法。

③ 基于图形的提取。该方法利用大模型从 \mathcal{M} 中提取细粒度的实体和关系，形成用于知识图谱构建的主语-谓语-宾语三元组（例如，MemO^g，Zep）。同时，记录时间元数据，如创建或无效化的时间戳，以支持基于图形的记忆中的动态更新和时间推理。附录 A 中提供了用于基于图形提取的具体提示示例。

5 内存管理

记忆管理过程决定了智能体系统如何随时间维护、优化和演化其记忆。如图 5 所示，该过程模拟了人类记忆的生命周期，包含五个核心操作：连接相关经验、整合碎片化信息、将短期记忆转化为长期记忆、更新过时内容以及过滤过时知识。表 2 总结了代表性智能体记忆方法如何通过不同的实现范式来实例化这些操作。

通过这一过程，系统能够维持一个连贯、高效且具有适应性的记忆状态，以支持持续学习与推理。

① 连接相关经验。人类会自然地在时间和情境之间关联相关事件；智能体系统通过连接来模拟这一过程。该机制在具有语义相似度、时间接近性或上下文相关性的记忆条目之间建立显式连接，通过图中的**结构边**或离散记录间的**关联链接**实现。例如，A-MEM 和 MemoryOS 等记忆方法利用基于语义相似度或连续性的关联链接，实现连接记忆之间的同步更新与对齐。分离地，基于图形的方法如 Zep 和 MemO^g 专注于连接单个回合或实体结点，以支持概念上或时间上对齐的记忆之间的推理与检索。

② 整合碎片化记忆。人类倾向于对日常经历进行总结，仅保留关键事件而忽略细节。智能体系统通过**抽象**或**摘要**实现类似的整合。例如，MemoryBank 将重复的日常记录聚合为事件摘要，并随着经验积累不断优化全局用户画像。同样地，MemoChat 将相关对话按共享主题分组，并生成主题级别的摘要。这一过程减少了冗余，提炼出核心信息，并将零散的记忆转化为适合长期存储的简洁高层表示。

③ 跨记忆层级的变换。人类记忆会逐步将重要信息从低层级存储转移到高层级存储，反复回忆以强化记忆。智能体系统采用类似的分层迁移机制。例如，MemoryOS 实现了两阶段迁移策略：短期记忆首先按照先进先出（FIFO）策略迁移到中期存储，随后中期记忆通过基于热度的得分被提升至长期存储，该得分综合考虑访问频率和最近性。此外，如 Zep 等记忆方法将语义相关的记忆组织成社区，形成结构化、相互关联的长期表示。这一阶段在保持效率的同时强化了持久性知识。

④ 更新现有记忆。人类通过整合新经验并修正不一致性，持续地修订自己的记忆。智能体系统遵循类似的原理，通过三种主要的更新范式实现：(1) **基于规则的更新**，即根据预定义规则对现有记忆进行更新。例如，MemoryBank 采用艾宾浩斯遗忘曲线理论来随时间调整记忆强度。在 MemoryOS 中，新记忆根据语义和关键词相似性被整合到现有结构中；(2) **基于语言模型的更新**，即通过提示大型语言模型来总结、合并或解决条目之间的冲突。例如，MemTree 通过语言模型执行专门的聚合操作来更新其记忆，其中提示内容和子节点数量引导语言模型在将新内容写回父结点前，适当压缩并泛化信息。再如，Zep 的更新过程要求语言模型严格遵循提示中给出的详细语义约束和程序指南，执行解析任务。(3) **基于智能体的更新**，即智能体自主决定应用何种操作（如修改、合并、修剪），如 MemGPT 和 MemOS 所示。更具体地说，智能体可访问当前上下文以及历史或归档的记忆条目，并学习利用专用系统工具高效且灵活地管理记忆。这些策略确保了记忆的准确性、一致性和与不断演进的知识相一致。

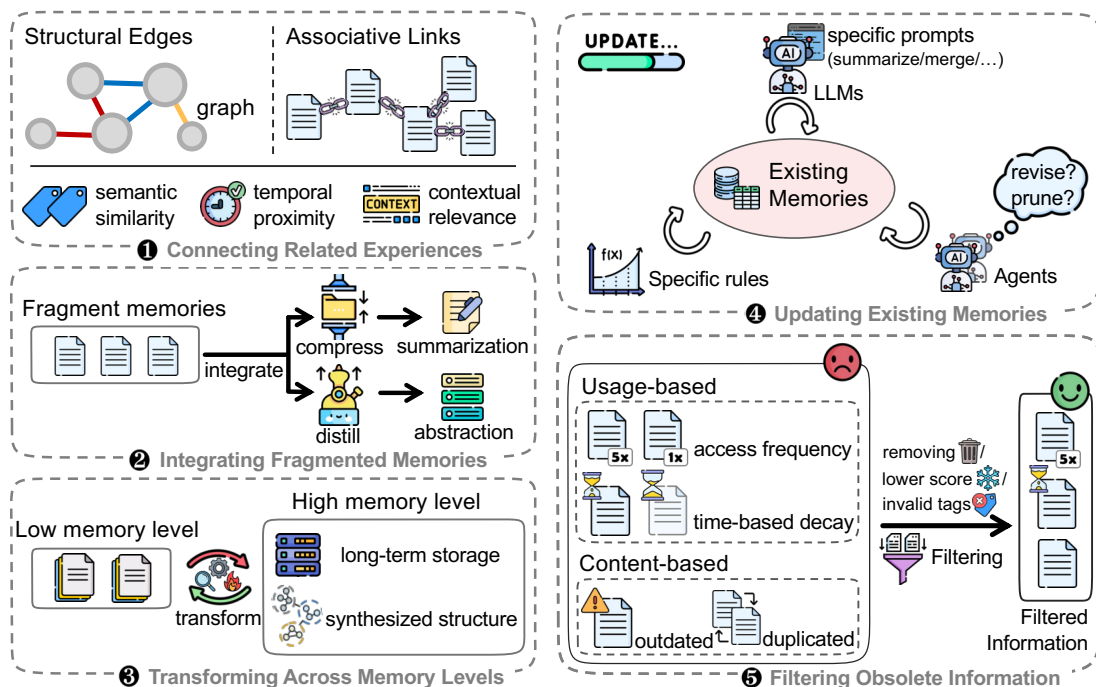


图 5: 内存管理过程的工作流。

⑤ 过滤过时信息。最后，记忆系统必须通过过滤过时或冗余的信息来保持紧凑和相关性，这可以通过直接删除记忆、降低其分配的权重得分，或应用状态标签（如“无效”）来实现。这种过滤过程类似于人类的遗忘机制，即选择性地淡化未使用或不相关的记忆。(1) **基于使用的过滤**，如 MemoryOS 和 MemoryBank 中所见，依赖于访问频率和时间衰减。创建时间久远且很少被检索的记忆会优先被过滤。(2) **基于内容的过滤**通过分析语义相似度，并利用大模型检测和过滤重复或过时的知识，如 Mem0 和 Mem0⁹ 中所述，从而减少噪声并提高检索准确率。这些机制共同维持了一个高效、轻量级的记忆系统，支持持续的适应与学习。

6 记忆存储

记忆存储组件决定了经过处理的记忆在两个主要维度上的组织与持久化方式：以组织为中心和以表示为中心。前者决定了存储系统的架构深度，包括扁平化存储和分层存储；后者则描述了所采用的技术范式，主要包括基于向量的存储和基于图形的存储。

① 扁平化存储。该方法表示一种统一的单层存储，将所有信息聚合在同质空间中，例如先进先出队列或 JSON 文件，相对于组织中心维度内的层级存储进行定义。

② 分层存储。该方法将内存划分为专用的多层架构，使各个存储组件能够履行不同的功能角色，并在不同粒度级别上运行。例如，MemoryOS 将内存组织为三层分层结构：用于及时对话的短期记忆、用于主题摘要

的中期记忆以及用于用户偏好的长期记忆。通过为各个存储组件应用不同但协同的管理与检索策略，分层存储有效地优化了计算开销与知识持久性之间的权衡。

③ 基于向量的存储。该方法将文本记忆编码为高维嵌入，随后在专用的向量库或数据库（如 FAISS [13] 和 Qdrant）中进行索引，使智能体能够执行高效的语义相似度搜索。基于向量的存储可作为独立的存储库运行，也可作为更复杂存储架构中频繁集成的基础构建块。

④ 基于图形的存储。该方法利用多种图形拓扑结构，如树、知识图谱和时序图，以保留记忆固有的丰富结构信息。例如，MemTree 将记忆组织成层次化树形结构，其中每个结点封装了聚合的文本内容，沿着树的深度提供不同层次的抽象；Zep 采用分层的时序知识图谱，同时通过将原始消息表示为结点、提取主语-谓语-宾语三元组，并将实体聚类为社区来组织记忆。这些基于图形的存储方法能够捕捉超出简单向量相似度度量范围的复杂关系和多跳关联。

7 信息检索

此组件负责控制智能体系统如何从记忆存储中识别并提取最相关的信息，以支持有根据的推理或上下文感知的响应生成。现有的信息检索策略可根据其采用的基本机制大致分为四种范式：

① 基于词法的检索。该范式依赖于表面层次的 token 或术语重叠，通常通过一些代表性技术实现，例如基

表 2: 内存管理操作的实现范式比较; “N/A” 表示该操作未被显式实现。

Method	Connecting	Integrating	Transforming	Updating	Filtering
A-MEM	Associative Links	N/A	N/A	LLM-based	N/A
MemoryBank	N/A	Summarization	N/A	Rule-based	Usage-based
MemGPT	N/A	Summarization	Stage-wise Transfer	Agent-based	N/A
MemO	N/A	Summarization	N/A	Agent-based	Content-based
MemO ^g	Structural Edges	N/A	N/A	LLM-based	Content-based
MemoChat	N/A	Abstraction	N/A	N/A	N/A
Zep	Structural Edges	N/A	Community Formation	LLM-based	N/A
MemTree	Structural Edges	Summarization	N/A	LLM-based	N/A
MemoryOS	Associative Links	Abstraction + Summarization	Stage-wise Transfer	Rule-based	Usage-based
MemOS	Structural Edges	Abstraction + Summarization	N/A	Agent-based	N/A

于集合的匹配（如使用 Jaccard 相似度系数）或评分模型（如 BM25 [74]）。基于词法的检索为确切术语匹配提供了强有力的基准，对于检索名称、特定实体或短语尤其有效，因为在这些场景中精确的措辞至关重要。

② **基于向量的检索**。该范式利用连续向量空间中的语义相似度来解决确切关键词匹配中固有的词表不匹配问题。通过使用嵌入模型将查询和记忆编码为高维向量，基于向量的检索被表述为一种基于余弦相似度等距离度量的前 k 个最近邻搜索，以找到最相关的条目。这种方法在捕捉潜在语义细微差别方面表现优异，确保相关性由语义内容决定，而非表面层次的词汇形式。为了在大规模记忆存储中保持效率，通常采用近似最近邻（ANN）搜索算法，例如 HNSW [58] 或 PQ [30]（乘积量化）。

③ **基于结构的检索**。该范式利用记忆实体之间的显式关系连接，通常在基于图形或层次化存储上操作，通过图遍历、邻域扩展或多跳推理来检索相互关联的信息簇，而非简单的查询到条目匹配。例如，MemO^g 从通过相似度搜索识别出的结点出发，探索其关系以构建一个全面的子图，从而捕捉相关且多方面的信息。类似地，Zep 采用基于广度优先搜索的图遍历算法，通过识别额外的结点和边来增强初始搜索结果。

④ **大模型辅助检索**。该范式将大模型作为主动的推理组件，用于指导或优化检索过程。除了直接决定应检索的具体信息外，大模型还可用于将模糊的用户查询转换为精确的搜索查询，或识别查询中的关键实体，以促进更精准的检索。通过利用大模型的推理能力，该范式在揭示潜在语义依赖关系方面表现出色，从而确保查询与检索到的知识之间具有更紧密的匹配。

8 实验

我们现在展示实验结果。我们在第 8.1 节中讨论实验设置，然后在第 8.2 节中报告多个实验的评估结果。

8.1 设置

► 我们评估的工作流。

我们对智能体记忆机制在三个维度上进行了系统的实验研究：(1) 我们在第 3 节所述的统一框架内收集并重新实现了 10 种代表性方法；(2) 我们使用多种互补的度量指标，在两个广泛使用的长期记忆基准上进行了全面评估；(3) 我们开展了多维度分析，以评估架构上的权衡与鲁棒性，涵盖 token 成本效率、真实位置敏感性、上下文可扩展性以及大语言模型主干依赖性。

► **基准数据集**。我们采用两个基准数据集，LOCOMO 和 LONGMEMEVAL，来评估每种记忆机制的性能。这两个数据集均旨在评估长期对话记忆能力，但代表了两种不同的交互场景：LOCOMO 基于两人之间的对话，而 LONGMEMEVAL 则基于用户与 AI 之间的交互。

- **LOCOMO**。LOCOMO 基准 [57] 包含十个用于问答评估的长期对话。每个对话平均包含 198.6 个问题，跨越 27.2 个会话，两位说话者之间约有 588.2 轮对话。问题分为四类：单跳检索、多跳检索、时间推理和开放领域知识。
- **LONGMEMEVAL**。LONGMEMEVAL 基准测试 [91] 包含 500 个高质量问题，旨在评估四项核心的长期记忆能力：信息提取、多会话推理、知识更新和时间推理。每个问题均基于专门的对话历史，这些对话历史源自长期的用户与 AI 交互，平均包含 50.2 次会话，约 115,000 个 token。

有关数据集的更多细节见附录 B.1。由于 LONGMEMEVAL 具有可配置的结构，我们构建了特定的变体来评估 **上下文可扩展性** 和 **位置敏感性**。详细的变体构造方法在附录 B.2 中有说明。

► **评估指标**。遵循两个基准和先前关于长时程对话记忆研究的评估协议，我们采用两种互补的度量方法。**F1** 通过平衡准确率和召回率来衡量 token 级别的重叠

表 3: 在 LONGMEMEVAL 上对方法的比较, 其中 紫色 表示最佳结果, 橙色 表示排除最佳结果后的最佳结果。

Method	Information Extraction						Multi-Session		Temporal		Knowledge Updates		Overall	
	user		assistant		preference									
	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
Qwen2.5-7B-Instruct														
A-MEM	46.75	41.26	43.21	36.74	9.76	0.52	16.61	14.32	19.37	15.54	25.59	21.63	25.53	21.24
MemoryBank	46.28	42.19	49.31	43.08	13.96	3.79	19.41	16.42	17.80	13.15	31.51	27.31	27.65	23.03
MemGPT	52.82	47.54	52.26	44.56	13.61	2.90	18.61	16.71	21.65	14.81	28.09	24.85	29.16	24.08
Mem0	56.51	51.23	32.35	25.99	11.44	0.69	23.02	20.72	28.86	21.19	41.28	36.37	32.46	26.95
Mem0 ^g	53.21	47.39	18.18	13.82	10.23	0.79	24.42	22.04	29.16	21.60	40.43	35.51	30.66	25.38
MemoChat	6.44	3.24	16.95	12.74	7.18	0.12	12.27	10.09	17.49	11.35	6.51	4.28	12.16	8.26
Zep	—	—	—	—	—	—	—	—	—	—	—	—	—	—
MemTree	68.48	61.85	57.79	46.73	9.50	1.00	21.96	19.88	29.97	21.77	41.49	38.56	36.92	31.05
MemoryOS	56.85	53.49	61.32	52.71	12.40	1.18	19.35	17.80	26.34	21.61	30.62	27.97	32.50	28.31
MemOS	65.48	56.02	49.40	42.44	12.17	0.77	22.81	19.72	21.34	15.65	33.99	27.75	32.48	26.38
Qwen2.5-72B-Instruct														
A-MEM	54.43	46.92	49.05	39.91	10.38	0.55	19.07	17.04	22.93	19.40	39.19	28.84	31.32	25.79
MemoryBank	52.96	48.64	57.41	51.63	12.53	4.91	28.21	24.97	25.32	21.73	36.02	29.34	35.79	30.70
MemGPT	54.81	49.39	63.23	47.84	4.78	0.03	19.29	16.04	24.48	18.47	29.21	23.16	32.63	25.82
Mem0	67.48	60.81	47.38	39.08	11.89	0.71	28.13	26.41	30.85	24.34	43.35	40.14	37.85	32.62
Mem0 ^g	62.63	54.37	40.21	35.82	10.75	0.53	31.47	27.00	33.39	28.97	47.15	43.23	38.47	33.25
MemoChat	18.98	15.29	26.38	20.21	7.80	0.09	20.46	18.39	27.33	19.24	21.33	17.70	22.09	17.18
Zep	—	—	—	—	—	—	—	—	—	—	—	—	—	—
MemTree	66.21	58.77	68.09	58.35	11.09	0.17	37.02	33.74	34.65	24.30	48.90	43.63	44.25	37.02
MemoryOS	73.00	68.23	75.10	64.12	12.43	1.20	36.56	32.67	32.31	23.63	53.46	48.96	46.04	39.42
MemOS	69.06	62.64	53.88	44.09	12.37	0.70	29.17	25.24	30.47	19.51	48.51	40.85	39.88	32.02

程度。**BLEU-1** 在引入简短惩罚项的前提下捕捉一元语法级别的修正准确率, 反映与参考答案之间的词汇保真度。这两种度量方法在两个数据集上均针对每种能力类别进行报告, 与代表性智能体记忆研究中的标准惯例保持一致。

► **实现。** 我们在所提出的统一框架下使用 Python 实现了所有方法, 确保基于原始论文和公开代码的忠实且一致的复现。所有实验均在 8 块 NVIDIA A100 (80 GB) GPU 上进行。如果某方法无法在两天内完成, 我们将在表格中将其结果标记为“—”。此外, 由于 F1 和 BLEU-1 指标对答案的冗长性非常敏感, 我们对所有生成的答案应用了统一的简化步骤。该过程的具体提示详见附录 A。

► **超参数设置。** 除非另有说明, 我们默认使用 Qwen2.5-7B-Instruct 作为 LLM 主干模型, 因其在现有的智能体记忆研究中被广泛采用。最大上下文长度设置为 20,000 token, 我们采用贪婪解码以确保输出的确定性。对于所有涉及 top- k 检索的方法, 遵循先前的工作 [32, 73, 93], 我们将 $k = 10$ 设置为符合上下文长度预算。我们采用 all-MiniLM-L6-v2 这一具有代表性和广泛应用的句子变换器模型, 作为所有方法中的统一嵌入模型。

其余所有方法特定的超参数均遵循其各自论文和代码库中报告的原始设置。

8.2 评估

► **实验 1. 整体性能。** 我们首先报告了所有智能体记忆方法在两个基准测试上, 针对两种模型规模 (Qwen2.5-7B-Instruct 和 Qwen2.5-72B-Instruct) 的性能表现。LONG-MEMEVAL 和 LOCOMO 上的结果分别如表 3 和表 4 所示。基于这些结果, 我们得出以下观察:

(1) 基于树结构的记忆方法 (例如 MemTree 和 MemOS) 通常通过以多层、多粒度的方式组织记忆, 实现优异的性能。具体而言, MemTree 在 7B 规模下于 LONG-MEMEVAL 上取得了最高的 F1 得分 36.92, 而 MemOS 在 7B 和 72B 规模下分别于 LOCOMO 上取得了最高的 F1 得分 37.05 和 42.79。树结构在上层提供高层次的概念摘要, 同时在叶结点处保留细粒度细节。通过精心设计的层次化架构, 同样可以实现这一优势, 该架构促进了不同抽象层级间高效的信息流动与变换, 正如 MemoryOS 和 Zep 所展现出的极具竞争力的结果所示。

(2) 保持信息完整性对于实现有效的记忆持久性至关重要——具体而言, 在信息提取阶段保留原始消

表 4: LOCOMO 上的方法比较。

Method	Single-Hop		Multi-Hop		Temporal		Open-Domain		Overall	
	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
Qwen2.5-7B-Instruct										
A-MEM	33.11	27.59	25.38	18.36	14.69	12.19	14.69	12.55	26.71	21.75
MemoryBank	15.34	11.92	16.63	12.41	11.57	9.49	16.09	11.16	14.84	11.46
MemGPT	21.31	16.57	18.92	13.64	17.37	13.61	11.14	8.06	19.42	15.51
Mem0	25.92	21.47	19.43	14.76	37.49	31.26	16.93	10.58	26.58	21.60
Mem0 ^g	27.90	23.45	21.14	13.98	35.70	30.54	18.67	13.44	27.71	22.57
MemoChat	7.21	5.84	9.74	6.59	5.33	4.26	14.31	10.81	7.72	5.96
Zep	40.42	34.59	30.97	20.31	22.64	17.82	21.79	17.57	33.82	27.42
MemTree	33.36	27.91	26.40	17.59	25.62	21.45	20.67	16.23	29.68	23.95
MemoryOS	33.14	28.33	28.52	19.90	17.65	14.09	22.05	16.95	28.34	23.11
MemOS	39.55	33.58	34.04	23.38	37.55	31.90	22.55	16.51	37.05	30.30
Qwen2.5-72B-Instruct										
A-MEM	42.24	38.68	27.87	21.97	28.01	22.64	23.27	18.55	30.35	25.46
MemoryBank	24.54	20.26	23.28	18.62	13.21	10.18	19.12	14.35	21.61	17.50
MemGPT	28.78	21.15	24.21	19.11	20.28	16.37	16.42	13.42	25.40	20.30
Mem0	32.83	27.89	25.35	21.14	40.76	36.03	21.04	14.09	32.38	27.51
Mem0 ^g	29.58	25.19	30.22	24.67	43.89	38.44	20.37	12.81	32.07	27.06
MemoChat	12.79	10.02	13.16	9.19	16.64	13.91	15.43	11.65	13.83	10.78
Zep	42.46	35.99	33.11	23.39	34.58	27.89	15.87	11.47	37.45	30.46
MemTree	38.73	32.42	33.01	25.07	29.81	24.48	23.02	17.46	34.85	28.49
MemoryOS	43.56	37.00	36.32	27.91	37.36	29.58	22.57	18.22	39.63	32.62
MemOS	44.30	36.85	36.67	26.95	49.59	43.13	24.88	17.87	42.79	35.16

息,并在最终响应生成阶段融入原始对话。例如,仅提取基于图形三元组的方法相较于保留原始对话片段的方法可能更容易出现信息丢失,这或许可以解释为何在许多情况下 Mem0 的表现优于 Mem0^g。

(3) 有效的记忆组织需要具备构建相关信息之间显式或隐式连接的机制,从而增强存储记忆的一致性。这一点在多跳推理任务中尤为重要。缺乏此类关联操作的方法,如 MemoryBank、MemGPT 和 MemoChat,在 LONGMEMEVAL 的多会话任务和 LOCOMO 的多跳任务上表现不佳。相比之下,尽管 Mem0 没有显式的“连接”操作符,但其在摄入过程中同时更新相似记忆的策略实现了相近的整合效果。值得注意的是,在 LONGMEMEVAL 的多会话任务中,Mem0 相较于 MemoryBank 分别取得了 18.60% 的 F1 和 26.19% 的 BLEU-1 提升。

(4) 由于时间推理的复杂性和固有难度,这些任务对主干模型的推理能力仍高度敏感。例如,当主干大语言模型从 7B 扩展到 72B 时,MemoryOS 和 MemoChat 在 LOCOMO 上的性能提升超过 2×。为减轻对大语言模型推理能力的过度依赖并提高系统的鲁棒性,设计专门用于时间信息处理的架构组件至关重要,而不应仅依赖于模型在生成回复时的上下文推理能力。

► **实验 2. token 消耗分析。**在本实验中,我们从两个角度评估了每种方法的计算开销:(1)我们分析了性能与成本之间的整体权衡。图 6 展示了每次对话的

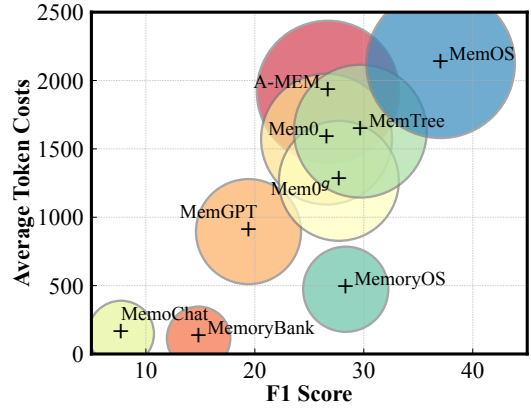


图 6: LOCOMO 上性能与 token 成本之间的整体权衡。

平均 token 消耗 (Y 轴) 与总体 F1 得分 (X 轴) 之间的关系;(2) 我们考察了每种方法在记忆摄入阶段的可扩展性。图 7 显示了随着摄入记忆量的增加,平均 token 消耗的变化情况。基于这些结果,我们得出以下观察:

(1) 通常情况下,更高的性能与更高的 token 消耗相关,反映出大规模使用大语言模型的优势。然而,内存框架的设计仍是决定成本效率的主要因素。尽管 MemTree 和 MemOS 实现了高准确率,但它们带来了显

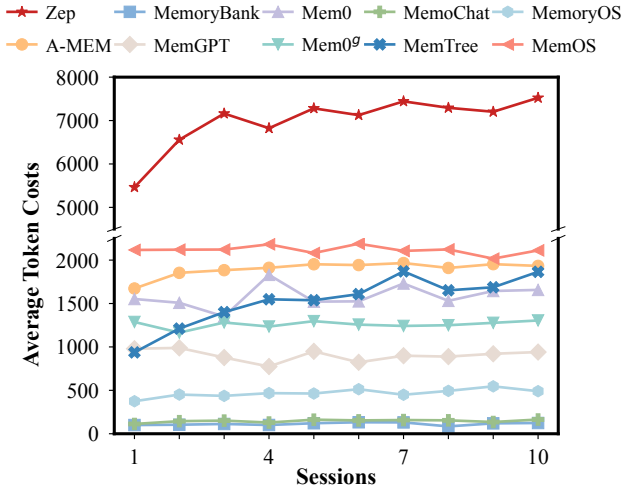


图 7: LOCOMO 上各会话的平均每次对话的 token 成本。

著的 token 开销。相比之下, MemoryOS 保持了良好的平衡, 在实现强劲性能的同时, token 成本显著降低。像 MemoChat 和 MemoryBank 这类更简单的方法虽然最大限度地减少了 token 使用量, 但未能达到足够的准确率。值得注意的是, 尽管 MemGPT 和 MemOS 都受到操作系统的启发, 并采用基于智能体的更新机制以自主决定应用哪些操作, 但 MemOS 将更多 token 分配给复杂的决策过程, 从而在性能上优于 MemGPT。

(2) 信息处理的粒度对 token 成本有显著影响。在信息提取阶段, 这一粒度取决于信息是从单个对话回合中提取, 还是从多个回合中集体提取。例如, MemoryOS 将对话划分为片段以进行中期存储, 而 MemoryBank 则以每日粒度将历史消息汇总成摘要。由于现代大模型具备强大的推理能力, 这种粒度上的粗化并不一定导致性能下降, 甚至可能提升性能, 为提高记忆效率提供了一条可行路径。

(3) 随着内存容量的增长, 某些方法在平均 token 消耗方面的可扩展性较差。对于 MemTree, 随着对话历史的累积, 树的深度增加, 导致每轮处理的成本上升, 因为每次自顶向下插入新的对话结点都需要更新路径上的所有结点。类似地, Zep 的图复杂度随对话轮次的增加而增长, 导致去重和一致性维护的成本不断上升。

► **实验 3. 上下文可扩展性分析。** 在本实验中, 我们通过将 LONGMEMEVAL 的上下文长度从 50% 扩展到 200%, 研究了各种内存架构的上下文可扩展性。在长时程交互中, 随着信息密度的增加, 内存系统面临的主要挑战从简单的检索转变为稳健的噪声抑制。我们在不同上下文大小下评估了多种架构, 以观察它们在上下文增长时保持检索准确率的能力。图 8a 展示了上下文可扩展性的总体趋势, 而详细的实验结果见附录 B.3。

表 5: LONGMEMEVAL (F1 得分) 上代表性记忆机制在信息提取子任务中的位置敏感性分析。

Method	Position	user	assistant	preference
Mem0 ^g	Early	53.95	19.62	9.93
	Middle	48.88	17.02	10.52
	Late	63.48	17.91	11.01
	Improvement	+9.53	-1.71	+1.08
MemTree	Early	52.85	49.26	9.31
	Middle	60.03	60.82	10.58
	Late	63.59	58.18	9.72
	Improvement	+10.74	+8.92	+0.41
MemOS	Early	60.65	43.00	11.20
	Middle	64.24	49.26	11.97
	Late	69.88	51.30	11.86
	Improvement	+9.23	+8.30	+0.66

(1) 当上下文规模从 50% 扩展到 200% 时, 几乎所有内存架构的 F1 得分均呈现稳定下降趋势。这种性能退化主要由无关信息密度的增加所驱动, 导致检索过程中的信噪比降低。

(2) 缩放结果揭示了源于操作复杂性的性能差异。MemOS 和 MemGPT 等方法采用“大语言模型即操作系统”范式, 要求大语言模型通过复杂的工具调用自主管理内存。当规模扩大至 200% 时, 候选空间的扩展显著增加了大语言模型在准确推理和执行这些管理指令方面的难度, 导致工具调用失败率和索引冲突率升高。相比之下, MemoryOS 通过采用显式的基于规则的分层管理机制, 降低了智能体的认知负担。通过将组织逻辑从大语言模型中卸载至确定性框架, MemoryOS 保持了高稳定性, 表明通过精心设计简化智能体内部管理开销对于实现稳健缩放至关重要。

(3) 不同的任务类别对缩放压力的敏感度存在差异。如图 9 所示, 知识更新 (KU) 尤为敏感, 表现出显著的性能下降, 因为内存容量增大导致冲突记录的密度上升。由于 KU 要求模型在相互排斥的多个版本中识别最新事实, 因此更多过时选项的存在会直接增加检索干扰。相反, 时间类任务相对稳定, 因为它们依赖于事件之间的相对顺序, 即使背景信息量增加, 这种结构上的区分依然保持清晰。与 KU 中的版本冲突不同, 事件的时间先后关系不易受到上下文增加的影响。

► **实验 4. 位置敏感性分析。** 在本实验中, 我们评估关键证据的位置对变体 LONGMEMEVAL 的检索和推理的影响, 将证据置于上下文的早期 (前 1/3)、中期 (中间 1/3) 或晚期 (后 1/3) 部分。随着证据在上下文中出现得越早, 记忆系统必须跨越更大的时间间隔, 并处理后续对话带来的更大干扰。该实验测试了不同架构是否保持对历史记录的平均访问, 还是表现出对近期输入的偏向性。如图 8b 所示, 不同位置放置

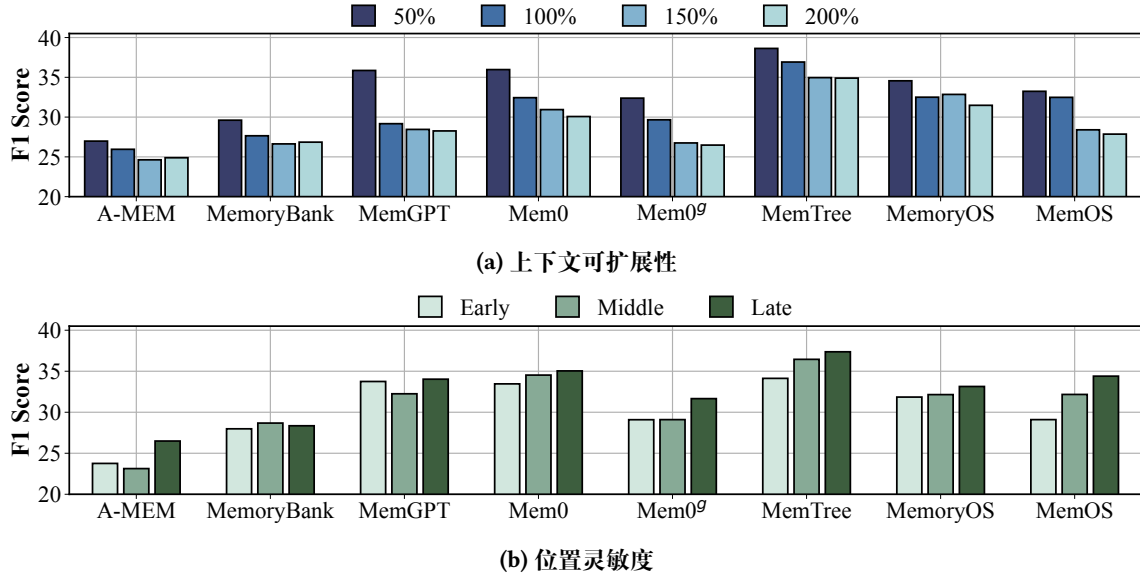


图 8: LONGMEMEVAL 上对记忆机制的鲁棒性分析。(a) 展示了输入规模从 50% 变化到 200% 时的上下文可扩展性, 而 (b) 则显示了在不同相对位置 (早期 (前 1/3)、中期 (中间 1/3)、晚期 (后 1/3)) 放置真实信息时的位置敏感性。

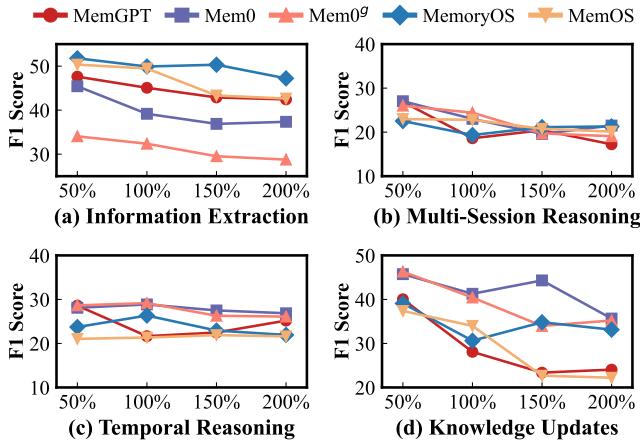


图 9: 在 LONGMEMEVAL 上, 不同任务类别下各种内存方法的上下文可扩展性。

下的性能表现存在显著差异。详细结果见附录 B.3, 主要发现总结如下:

(1) 在整体层面观察到明显的近期偏差, 随着支持性证据与查询之间的时序距离增加, 这种偏差愈发显著。如图 8b 所示, 当证据位于较晚会话而非较早会话时, 大多数方法的整体 F1 得分更高。具体而言, MemTree 的整体 F1 得分从 34.13 提升至 37.37, MemOS 的得分从 29.10 提升至 34.40, 等等。随着相关证据出现后对话内容的不断累积, 保持长距离信息的一致性与检索变得越来越具有挑战性。

(2) 内存更新策略显著影响位置敏感性。对于 A-MEM 等方法, 动态内存修订可能覆盖早期证据, 使早期会话信息容易受到后续交互的影响。层次化方法如 MemTree 和 MemOS 表现出不同的机制: 尽管原始对话在叶结点中被保留, 但高层摘要的更新会放大近期信息的影响, 从而加剧了后期-早期差距。相比之下, MemoryOS 通过跨记忆层级的阶段式传递, 更均衡地保留历史信息。早期证据在每一层级内相对独立, 而非反复与后期信息合并。因此, 后续交互不会直接重塑早期证据表示, 这解释了 MemoryOS 的后期-早期差距较小 (+1.29 F1)。

(3) 位置敏感性高度依赖于类别。我们在表 5 中考察了信息提取子任务中的三种代表性方法。短暂的、会话局部的信息表现出比持久特征更强的位置敏感性。用户和助手提取任务显示出显著的早期到晚期变化: 在用户提取任务中, MemO_g / MemTree / MemOS 的 F1 值分别提升了 +9.53 / +10.74 / +9.23; 在助手提取任务中, MemTree / MemOS 的 F1 值分别提升了 +8.92 / +8.30。相比之下, 偏好提取保持稳定: 提升幅度仅为 +1.08、+0.41 和 +0.66 F1。这表明位置敏感性与信息持久性相关: 短暂的会话局部细节更容易受到后期干扰, 而持久的偏好特征则较少受证据重定位的影响。

► **实验 5. 大语言模型主干对比。** 在本实验中, 我们评估了多种记忆方法在不同大语言模型主干上的表现。具体而言, 我们从每种设计范式中选取具有代表性和高性能的方法, 进行跨主干的对比评估。因此, 我们排除了较简单的基准方法, 以聚焦于具有非平凡记忆架构的方法。结果如表 6 所示。总体来看, 大多数方法在

表 6: 在 LOCOMO 上对不同大模型骨干网络的比较。

Model	Metric	Zep	MemTree	MemoryOS	MemOS	Ours
Qwen2.5-7B	F1	33.82	29.68	28.34	37.05	38.03
	BLEU-1	27.42	23.95	23.11	30.30	31.73
Qwen2.5-72B	F1	37.45	34.85	39.63	42.79	43.87
	BLEU-1	30.46	28.49	32.62	35.16	37.30
LLaMA3.1-8B	F1	—	23.49	32.21	33.39	35.21
	BLEU-1	—	18.19	25.56	26.24	28.40
GPT-4o-mini	F1	42.88	32.32	42.57	45.56	45.26
	BLEU-1	36.09	25.89	34.75	38.06	38.52

[†] Results for Zep on LLaMA3.1-8B are marked as “—” due to strict JSON formatting failures, as noted in its official documentation¹.

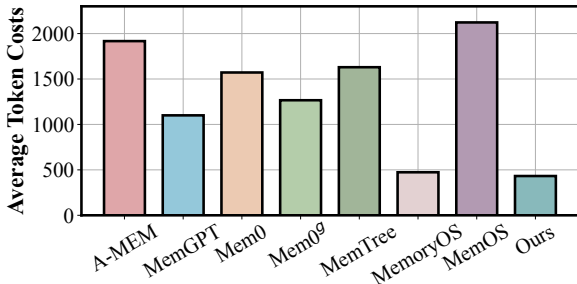


图 10: 我们新设计方法在平均 token 代价方面的比较。

闭源模型 GPT-4o-mini 主干上取得了最佳性能。在开源模型中, Qwen2.5-7B 在各类方法中普遍优于 LLaMA3.1-8B, 仅在 MemoryOS 方法中例外, 此时 LLaMA3.1-8B 表现更优。将 Qwen2.5 从 7B 缩放至 72B 为所有记忆方法带来了显著提升, 正如实验 1 所述, 这表明现有的记忆架构仍严重依赖主干模型的推理能力。

► **实验 6. 新的 SOTA 算法。**基于上述分析, 我们设计了一种新的内存框架, 在保持低 token 开销的同时实现了最先进的性能。我们的方法结合了 MemTree 和 MemOS 的树状组织结构以及 MemoryOS 的分层存储架构, 将内存划分为短期、中期和长期三个层级。图 11 展示了我们新设计方法的框架, 详细的工作流程和算法见附录 C。

表 7 和表 8 报告了性能指标, 图 10 比较了平均 token 开销。我们的方法在两个基准上均取得了最佳的整体性能, 并在所有任务类别中展现出极具竞争力的结果, 同时保持了极低的计算开销, 每轮对话的 token 数少于 450。具体而言, 在 LONGMEMEVAL 上, 我们的方法在所有评估类别中排名第一或第二。与表现最佳的现有方法相比, 其在信息提取辅助任务上的相对 F1 得分提升了 13.08%, 在整体得分上比最强基准

¹<https://help.getzep.com/graphiti/configuration/llm-configuration#ollama-local-llms>

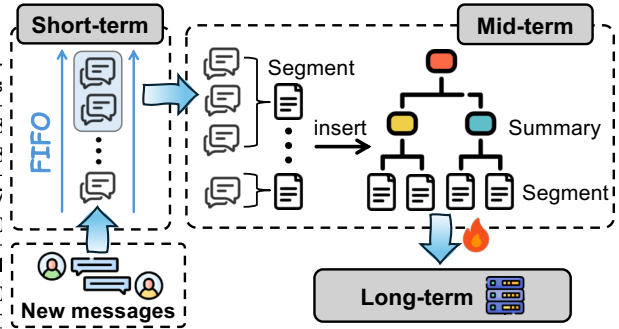


图 11: 我们新设计方法的框架。

高出 5.17%。在 LOCOMO 上, 我们的方法同样取得了最佳的整体性能, 超越了强大的基线 MemOS, 并在所有类别中持续表现出强劲结果, 尤其在基于 Qwen2.5-72B 骨干网络时提升尤为显著。尽管 MemOS 等基线通过集成多个高成本模块, 在特定复杂任务（如多跳检索和时间推理）上获得了更高得分, 但我们的方法在性能与 token 效率之间取得了最优平衡, 以更轻量化的架构实现了最佳整体性能。此外, 如表 6 所示, 我们的方法在多种 LLM 骨干网络上仍保持竞争力, 展现了强大的泛化能力。

9 课程与机遇

我们基于观察总结了对实践者的经验教训 (L), 并提出了切实可行的研究机会 (O)。

经验教训:

- **L1.** 与平面内存结构相比, 分层组织在捕捉信息之间的结构关系方面更为有效, 这可以通过采用基于树的索引或设计多级存储来实现。
- **L2.** 信息完整性是记忆机制的基础。尽管图中的三元组等结构化表示有助于提升组织性, 但在信息提取或检索阶段, 保留原始对话上下文对于防止语义损失至关重要。
- **L3.** 通过在信息提取或记忆管理阶段将多个对话轮次作为一个单元进行处理, 优化内存粒度可显著降低 token 消耗, 而适当的划分则进一步保持了检索信息的连贯性。

机遇:

- **O1.** 在真实情景中, 记忆涉及复杂且多样的信息来源, 包括文本对话、历史交互迹、以及音频、图像和视频等多模态信号。现有的记忆机制通常仅关注单一或有限的形式, 这限制了有效信息的利用。一个有前景的未来研究方向是开发统一的记忆机制, 支持异构且多模态的记忆, 在共享的存储与检索框架内实现。
- **O2.** 现有的竞争性记忆方法通常以导致存储大小迅速增长以及管理与检索开销增加的方式进行信息的管理和维护。如何在不丢失有用信息的前提下压缩记忆, 仍然是一个重大挑战, 这为探索超越显式文本和学成

表 7: 使用 Qwen2.5-7B-Instruct (F1 得分) 在 LONGMEMEVAL 上对我们新设计方法的对比。

Category	A-MEM	MemoryBank	MemGPT	Mem0	Mem0 ^g	MemTree	MemoryOS	MemOS	Ours
Information Extraction-user	46.75	46.28	52.82	56.51	53.21	68.48	56.85	65.48	67.38
Information Extraction-assistant	43.21	49.31	52.26	32.35	18.18	57.79	61.32	49.40	69.34
Information Extraction-preference	9.76	13.96	13.61	11.44	10.23	9.50	12.40	12.17	14.58
Multi-Session Reasoning	16.61	19.41	18.61	23.02	24.42	21.96	19.35	22.81	23.14
Knowledge Update	19.37	17.80	21.65	28.86	29.16	29.97	26.34	21.34	29.22
Temporal Reasoning	25.59	31.51	28.09	41.28	40.43	41.49	30.62	33.99	43.53
Overall	25.53	27.65	29.16	32.46	30.66	36.92	32.50	32.48	38.79

表 8: 基于 Qwen2.5-7B/72B-Instruct (F1 得分) 在 LOCOMO 上对新设计方法的对比。

Category	Size	A-MEM	MemoryBank	MemGPT	Mem0	Mem0 ^g	Zep	MemTree	MemoryOS	MemOS	Ours
Single-Hop Retrieval	7B	33.11	15.34	21.31	25.92	27.90	40.42	33.36	33.14	39.55	45.11
	72B	42.24	24.54	28.78	32.83	29.58	42.46	38.73	43.56	44.30	48.01
Multi-Hop Retrieval	7B	25.38	16.63	18.92	19.43	21.14	30.97	26.40	28.52	34.04	30.01
	72B	27.87	23.28	24.21	25.35	30.22	33.11	33.01	36.32	36.67	38.08
Temporal Reasoning	7B	14.69	11.57	17.37	37.49	35.70	22.64	25.62	17.65	37.55	32.23
	72B	28.01	13.21	20.28	40.76	43.89	34.58	29.81	37.36	49.59	43.97
Open-Domain Knowledge	7B	14.69	16.09	11.14	16.93	18.67	21.79	20.67	22.05	22.55	18.92
	72B	23.27	19.12	16.42	21.04	20.37	15.87	23.02	22.57	24.88	24.27
Overall	7B	26.71	14.84	19.42	26.58	27.71	33.82	29.68	28.34	37.05	38.03
	72B	30.35	21.61	25.40	32.38	32.07	37.45	34.85	39.63	42.79	43.87

压缩机制的潜在表示以实现高密度且可用的记忆提供了机会。

► **O3.** 现有的分层记忆机制主要关注将短期记忆整合到长期存储中，但不支持反向变换。一个有前景的方向是设计双向记忆变换机制，以实现跨记忆层次的高效整合与重构。

10 相关作品

在本节中，我们简要回顾了与本研究最相关的一些先验工作，包括用于数据库的大模型和检索增强生成框架。

• **RAG 框架。** RAG 已被证明在多项任务中表现出色，包括开放式问答 [31, 81]、编程上下文 [7–9]、SQL 重写 [45, 85]、自动数据库管理系统配置调试 [80, 104] 以及数据清洗 [59, 60, 71]。朴素的 RAG 技术依赖于从外部知识库中检索与查询相关的信息，以缓解大模型的“幻觉”问题。最近，大多数 RAG 方法 [15, 22, 23, 37, 69, 75, 89, 92] 已采用图结构作为外部知识，用于组织文档中的信息和关系，从而提升了整体检索性能，本文对此进行了广泛综述。在开源软件方面，LangChain [34] 和 LlamaIndex [55] 库均支持多种图数据库，同时基于图形的 RAG 应用也逐渐兴起，包括能够在 Neo4j [62] 和 NebulaGraph [61] 上创建并推理知识图谱的系统。

更多细节请参考近期关于基于图形的 RAG 方法的综述与实验研究 [24, 69, 106]。

• **大模型在数据库中的应用。** 由于大量数据库论坛讨论中蕴含了丰富的开发者经验，近期的研究 [7, 16, 20, 35, 42, 45, 82, 85, 104, 105] 已开始利用大模型来提升数据库性能。例如，GPTuner [35] 提出通过利用领域知识，借助大模型识别重要的配置参数，并对其值进行粗略初始化，以供后续优化。此外，D-Bot [104] 提出了一种基于大模型的数据库诊断系统，能够检索相关的知识片段和工具，并利用它们准确识别典型的根因。基于大模型的数据分析系统与工具也受到了研究关注 [2, 10, 17, 25, 38, 39, 46–50, 68, 95, 100]。

据我们所知，我们的工作为首个为现有所有智能体记忆方法提供统一框架的研究，并通过深入的实验结果对它们进行了全面比较。

11 结论

本文对现有的记忆方法进行了深入的实验评估与比较。我们首先提出一个统一的模块化框架，将记忆机制抽象为四个核心单元——信息提取、记忆管理、记忆存储和信息检索。在此框架下，我们在两个基准数据集上系统地评估了具有代表性的记忆方法，并进一步开展多维度实验，研究 token 成本效率、上下文可扩展

性、证据位置敏感性以及大模型主干依赖性。基于实验结果与分析，我们通过结合现有技术开发了一种新的记忆变体，在保持低开销的同时实现了优异的准确率。最后，我们总结了经验教训，并提出了有助于未来研究的实际研究机遇。

REFERENCES

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 Technical Report. *arXiv preprint arXiv:2303.08774* (2023).
- [2] Eric Anderson, Jonathan Fritz, Austin Lee, Bohou Li, Mark Lindblad, Henry Lindeman, Alex Meyer, Parth Parmar, Tanvi Ranade, Mehul A Shah, et al. 2024. The Design of an LLM-powered Unstructured Analytics System. *arXiv preprint arXiv:2409.00847* (2024).
- [3] Anthropic. 2026. Introducing Claude Sonnet 4.6: Our fastest, smartest model is now available for all. <https://www.anthropic.com/news/claude-sonnet-4-6>.
- [4] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to Retrieve, Generate, and Critique Through Self-Reflection. *arXiv preprint arXiv:2310.11511* (2023).
- [5] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 3119–3137. <https://doi.org/10.18653/v1/2024.acl-long.172>
- [6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf
- [7] Sibe Chen, Ju Fan, Bin Wu, Nan Tang, Chao Deng, Pengyi Wang, Ye Li, Jian Tan, Feifei Li, Jingren Zhou, et al. 2024. Automatic Database Configuration Debugging using Retrieval-Augmented Language Models. *arXiv preprint arXiv:2412.07548* (2024).
- [8] Sibe Chen, Yeye He, Weiwei Cui, Ju Fan, Song Ge, Haidong Zhang, Dongmei Zhang, and Surajit Chaudhuri. 2024. Auto-Formula: Recommend Formulas in Spreadsheets using Contrastive Learning for Table Representations. *Proceedings of the ACM on Management of Data* 2, 3 (2024), 1–27.
- [9] Sibe Chen, Nan Tang, Ju Fan, Xuemi Yan, Chengliang Chai, Guoliang Li, and Xiaoyong Du. 2023. Haipipe: Combining Human-Generated and Machine-Generated Pipelines for Data Preparation. *Proceedings of the ACM on Management of Data* 1, 1 (2023), 1–26.
- [10] Zui Chen, Lei Cao, Sam Madden, Tim Kraska, Zeyuan Shang, Ju Fan, Nan Tang, Zihui Gu, Chunwei Liu, and Michael Cafarella. 2023. SEED: Domain-Specific Data Curation With Large Language Models. *arXiv e-prints* (2023), arXiv–2310.
- [11] Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. <https://arxiv.org/abs/2504.19413> arXiv:2504.19413.
- [12] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A Survey on In-Context Learning. *arXiv preprint arXiv:2301.00234* (2024). <https://arxiv.org/abs/2301.00234>
- [13] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. The Faiss Library. arXiv:2401.08281 [cs.LG] <https://arxiv.org/abs/2401.08281>
- [14] Yiming Du, Wenyu Huang, Danna Zheng, Zhaowei Wang, Sebastien Montella, Mirella Lapata, Kam-Fai Wong, and Jeff Z. Pan. 2025. Rethinking Memory in LLM based Agents: Representations, Operations, and Emerging Topics. arXiv:2505.00675 [cs.CL] <https://arxiv.org/abs/2505.00675>
- [15] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv preprint arXiv:2404.16130* (2024).
- [16] Ju Fan, Zihui Gu, Songyue Zhang, Yuxin Zhang, Zui Chen, Lei Cao, Guoliang Li, Samuel Madden, Xiaoyong Du, and Nan Tang. 2024. Combining small language models and large language models for zero-shot nl2sql. *Proceedings of the VLDB Endowment* 17, 11 (2024), 2750–2763.
- [17] Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. 2024. Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. *Proceedings of the VLDB Endowment* 17, 5 (Jan. 2024), 1132–1145. <https://doi.org/10.14778/3641204.3641221>
- [18] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint arXiv:2312.10997* (2023).
- [19] Aashish Ghimire, James Prather, and John Edwards. 2024. Generative AI in Education: A Study of Educators’ Awareness, Sentiments, and Influencing Factors. *arXiv preprint arXiv:2403.15586* (2024).
- [20] Victor Giannankouris and Immanuel Trummer. 2024. $\{\lambda\}$ -Tune: Harnessing Large Language Models for Automated Database System Tuning. *arXiv preprint arXiv:2411.03500* (2024).
- [21] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis. 2016. Hybrid Computing Using a Neural Network with Dynamic External Memory. *Nature* 538, 7626 (2016), 471–476. <https://doi.org/10.1038/nature20101>
- [22] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. LightRAG: Simple and Fast Retrieval-Augmented Generation. *arXiv e-prints* (2024), arXiv–2410.
- [23] Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. <https://openreview.net/forum?id=hkujvAPVsg>
- [24] Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, et al. 2024. Retrieval-Augmented Generation with Graphs (GraphRAG). *arXiv preprint arXiv:2501.00309* (2024).
- [25] Sirui Hong, Yizhang Lin, Bang Liu, Bangbang Liu, Binhao Wu, Ceyao Zhang, Danyang Li, Jiaqi Chen, Jiayi Zhang, Jinlin Wang, Li Zhang, Lingyao Zhang, Min Yang, Mingchen Zhuge, Taicheng Guo, Tuo Zhou, Wei Tao, Robert Tang, Xiangtao Lu, Xiawu Zheng, Xinbing Liang, Yay-ing Fei, Yuheng Cheng, Yongxin Ni, Zhibin Guo, Zongze Xu, Yuyu Luo, and Chenglin Wu. 2025. Data Interpreter: An LLM Agent for Data Science. In *Findings of the Association for Computational Linguistics: ACL 2025*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 19796–19821. <https://doi.org/10.18653/v1/2025.findings-acl.1016>
- [26] Cheng-Ping Hsieh, Simeng Sun, Samuel Krizan, Shantanu Acharya, Dima Rekes, Fei Jia, and Boris Ginsburg. 2024. RULER: What’s the Real Context Size of Your Long-Context Language Models?. In *Proceedings of the First Conference on Language Modeling (COLM)*.
- [27] Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. 2023. ChatDB: Augmenting LLMs with Databases as Their Symbolic Memory. arXiv:2306.03901 [cs.AI] <https://arxiv.org/abs/2306.03901>
- [28] Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. 2025. HiAgent: Hierarchical Working Memory Management for Solving Long-Horizon Agent Tasks with Large Language Model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 32779–32798. <https://doi.org/10.18653/v1/2025.acl-long.1575>

- [29] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *arXiv preprint arXiv:2311.05232* (2023).
- [30] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. 2011. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 1 (2011), 117–128. <https://doi.org/10.1109/TPAMI.2010.57>
- [31] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv:2403.14403* (2024).
- [32] Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025. Memory OS of AI Agent. *arXiv preprint arXiv:2506.06326* (2025).
- [33] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. 2023. DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines. *arXiv preprint arXiv:2310.03714* (2023).
- [34] Langchain. 2023. Langchain. https://python.langchain.com/docs/additional_resources/arxiv_references/.
- [35] Jiale Lao, Yibo Wang, Yufei Li, Jianping Wang, Yunjia Zhang, Zhiyuan Cheng, Wanguo Chen, Mingjie Tang, and Jianguo Wang. 2024. Gptuner: A manual-reading database tuning system via gpt-guided bayesian optimization. *Proceedings of the VLDB Endowment* 17, 8 (2024), 1939–1952.
- [36] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 9459–9474.
- [37] Dawei Li, Shu Yang, Zhen Tan, Jae Young Baik, Sukwon Yun, Joseph Lee, Aaron Chacko, Bojian Hou, Duy Duong-Tran, Ying Ding, et al. 2024. DALK: Dynamic Co-Augmentation of LLMs and KG to answer Alzheimer’s Disease Questions with Scientific Literature. *arXiv preprint arXiv:2405.04819* (2024).
- [38] Hongxin Li, Jingran Su, Yuntao Chen, Qing Li, and Zhao-Xiang Zhang. 2023. SheetCopilot: Bringing Software Productivity to the Next Level Through Large Language Models. In *Advances in Neural Information Processing Systems*, Vol. 36. 4952–4984.
- [39] Lan Li, Liri Fang, Bertram Ludäscher, and Vette I Torvik. 2025. AutoDCWorkflow: LLM-based Data Cleaning Workflow Auto-Generation and Benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.). Association for Computational Linguistics, Suzhou, China, 7766–7780. <https://doi.org/10.18653/v1/2025.findings-emnlp.410>
- [40] Xinzhe Li. 2024. A Review of Prominent Paradigms for LLM-Based Agents: Tool Use (Including RAG), Planning, and Feedback Learning. *arXiv:2406.05804 [cs.AI]* <https://arxiv.org/abs/2406.05804>
- [41] Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. Compressing Context to Enhance Inference Efficiency of Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 6342–6353. <https://doi.org/10.18653/v1/2023.emnlp-main.391>
- [42] Yiyan Li, Haoyang Li, Jing Zhang, Renata Borovica-Gajic, Shuai Wang, Tieying Zhang, Jianjun Chen, Rui Shi, Cuiping Li, and Hong Chen. 2025. AgentTune: An Agent-Based Large Language Model Framework for Database Knob Tuning. *Proc. ACM Manag. Data* 3, 6, Article 293 (Dec. 2025), 29 pages. <https://doi.org/10.1145/3769758>
- [43] Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2023. Large Language Models in Finance: A Survey. In *Proceedings of the fourth ACM international conference on AI in finance*. 374–382.
- [44] Zhiyu Li, Shichao Song, Chenyang Xi, Hanyu Wang, Chen Tang, Simin Niu, Ding Chen, Jiawei Yang, Chunyu Li, Qingchen Yu, et al. 2025. Memos: A memory os for ai system. *arXiv preprint arXiv:2507.03724* (2025).
- [45] Zhaodonghui Li, Haitao Yuan, Huiming Wang, Gao Cong, and Lidong Bing. 2025. LLM-R2: A Large Language Model Enhanced Rule-based Rewrite System for Boosting Query Efficiency. *Proceedings of the VLDB Endowment* 1, 18 (2025), 53–65.
- [46] Chen Liang, Donghua Yang, Zheng Liang, Zhiyu Liang, Tianle Zhang, Boyu Xiao, Yuqing Yang, Wenqi Wang, and Hongzhi Wang. 2025. Revisiting Data Analysis with Pre-trained Foundation Models. *arXiv preprint arXiv:2501.01631* (2025).
- [47] Yiming Lin, Mawil Hasan, Rohan Kosalge, Alvin Cheung, and Aditya G Parameswaran. 2025. TWIX: Automatically Reconstructing Structured Data from Templated Documents. *arXiv preprint arXiv:2501.06659* (2025).
- [48] Yiming Lin, Madelon Hulsebos, Ruiying Ma, Shreya Shankar, Sepanta Zeigham, Aditya G Parameswaran, and Eugene Wu. 2024. Towards Accurate and Efficient Document Analytics with Large Language Models. *arXiv preprint arXiv:2405.04674* (2024).
- [49] Chunwei Liu, Matthew Russo, Michael Cafarella, Lei Cao, Peter Baille Chen, Zui Chen, Michael Franklin, Tim Kraska, Samuel Madden, and Gerardo Vitagliano. 2024. A Declarative System for Optimizing AI Workloads. *arXiv preprint arXiv:2405.14696* (2024).
- [50] Chunwei Liu, Gerardo Vitagliano, Brandon Rose, Matt Prinz, David Andrew Samson, and Michael Cafarella. 2025. PalimpChat: Declarative and Interactive AI Analytics. *arXiv preprint arXiv:2502.03368* (2025).
- [51] Lei Liu, Xiaoyan Yang, Junchi Lei, Xiaoyang Liu, Yue Shen, Zhiqiang Zhang, Peng Wei, Jinjie Gu, Zhixuan Chu, Zhan Qin, et al. 2024. A Survey on Medical Large Language Models: Technology, Application, Trustworthiness, and Future Directions. *arXiv preprint arXiv:2406.03712* (2024).
- [52] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics* 12 (2024), 157–173.
- [53] Peiqi Liu, Zhanqiu Guo, Mohit Warke, Soumith Chintala, Chris Paxton, Nur Muhammad Mahi Shafiuallah, and Lerrel Pinto. 2025. DynaMem: Online Dynamic Spatio-Semantic Memory for Open World Mobile Manipulation. In *ICRA 2025 Workshop: Human-Centered Robot Learning in the Era of Big Data and Large Models*. <https://openreview.net/forum?id=RJKUlhDJg1>
- [54] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *arXiv:2107.13586 [cs.CL]* <https://arxiv.org/abs/2107.13586>
- [55] llamaindex. 2023. llamaindex. <https://www.llamaindex.ai/>.
- [56] Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. 2023. MemoChat: Tuning LLMs to Use Memos for Consistent Long-Range Open-Domain Conversation. *arXiv preprint arXiv:2308.08239* (2023).
- [57] Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating Very Long-Term Conversational Memory of LLM Agents. <https://arxiv.org/abs/2402.17753> *arXiv:2402.17753*.
- [58] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE transactions on pattern analysis and machine intelligence* 42, 4 (2018), 824–836.
- [59] Zan Ahmad Naeem, Mohammad Shahmeer Ahmad, Mohamed Eltabakh, Mourad Ouzzani, and Nan Tang. 2024. RetClean: Retrieval-Based Data Cleaning Using LLMs and Data Lakes. *Proceedings of the VLDB Endowment* 17, 12 (2024), 4421–4424.
- [60] Avanika Narayan, Ines Chami, Laurel Orr, and Christopher Ré. 2022. Can Foundation Models Wrangle Your Data? *Proceedings of the VLDB Endowment* 16, 4 (2022), 738–746.
- [61] NebulaGraph. 2024. NebulaGraph. <https://nebula-graph.io/>.
- [62] Neo4j. 2006. Neo4j. <https://neo4j.com/>.
- [63] Yuqi Nie, Yaxuan Kong, Xiaowen Dong, John M Mulvey, H Vincent Poor, Qingsong Wen, and Stefan Zohren. 2024. A Survey of Large Language Models for Financial Applications: Progress, Prospects and Challenges. *arXiv preprint arXiv:2406.11903* (2024).
- [64] OpenClaw Team. 2026. OpenClaw: Your own personal AI assistant. <https://github.com/openclaw/openclaw>.

- [65] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems*, Vol. 35. 27730–27744.
- [66] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. 2023. MemGPT: Towards LLMs as Operating Systems. *arXiv preprint arXiv:2310.08560* (2023).
- [67] Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST ’23)*. Association for Computing Machinery. <https://doi.org/10.1145/3586183.3606763>
- [68] Liana Patel, Siddharth Jha, Carlos Guestrin, and Matei Zaharia. 2024. Lotus: Enabling semantic queries with llms over tables of unstructured and structured data. *arXiv preprint arXiv:2407.11418* (2024).
- [69] Boci Peng, Yun Zhu, Yongchao Liu, Xiaohu Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph Retrieval-Augmented Generation: A Survey. *arXiv preprint arXiv:2408.08921* (2024).
- [70] Vijay Putta, Krishna Teja Areti, Ajay Guyyala, and Prudhvi Ratna Badri Satya. 2026. Self-Reflective Memory Consolidation in Agentic Architectures. *International Journal of Computer Applications* 187, 73 (Jan 2026), 1–14. <https://doi.org/10.5120/ijca2026926236>
- [71] Yichen Qian, Yongyi He, Rong Zhu, Jintao Huang, Zhijian Ma, Haibin Wang, Yaohua Wang, Xiuyu Sun, Defu Lian, Bolin Ding, et al. 2024. UniDM: A Unified Framework for Data Manipulation with Large Language Models. *Proceedings of Machine Learning and Systems* 6 (2024), 465–482.
- [72] Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. Zep: A Temporal Knowledge Graph Architecture for Agent Memory. *arXiv preprint arXiv:2501.13956* (2025).
- [73] Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. 2024. From Isolated Conversations to Hierarchical Schemas: Dynamic Tree Memory Representation for LLMs. *arXiv preprint arXiv:2410.14052* (2024).
- [74] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (April 2009), 333–389. <https://doi.org/10.1561/15000000019>
- [75] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval. *arXiv preprint arXiv:2401.18059* (2024).
- [76] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Advances in Neural Information Processing Systems*, Vol. 36.
- [77] Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023. ZeroSCROLLS: A Zero-Shot Benchmark for Long Text Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 7977–7989. <https://doi.org/10.18653/v1/2023.findings-emnlp.536>
- [78] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS ’23)*. Curran Associates Inc., Red Hook, NY, USA, Article 377, 8634–8652 pages.
- [79] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. 2025. OpenAI GPT-5 System Card. *arXiv:2601.03267 [cs.CL]* <https://arxiv.org/abs/2601.03267>
- [80] Vikramank Singh, Kapil Eknath Vaidya, Vinayshekhar Bannihatti Kumar, Sopan Khosla, Murali Narayanaswamy, Rashmi Gangadharaiyah, and Tim Kraska. 2024. Panda: Performance Debugging for Databases Using LLM Agents. (2024).
- [81] Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering. *Transactions of the Association for Computational Linguistics* 11 (2023), 1–17.
- [82] Tarek Stolz, István Koren, Liam Tirpitz, and Sandra Geisler. 2023. GA-LOIS: A Hybrid and Platform-Agnostic Stream Processing Architecture. In *Proceedings of the International Workshop on Big Data in Emergent Distributed Environments* (Seattle, WA, USA) (BiDEDE ’23). Association for Computing Machinery, New York, NY, USA, Article 3, 6 pages. <https://doi.org/10.1145/3579142.3594287>
- [83] Theodore Sumers, Shunyu Yao, Karthik R Narasimhan, and Thomas L. Griffiths. 2024. Cognitive Architectures for Language Agents. *Transactions on Machine Learning Research* (2024). <https://openreview.net/forum?id=1i6ZCvflQJ> Survey Certification, Featured Certification.
- [84] Haoran Sun, Shaoning Zeng, and Bob Zhang. 2026. H-MEM: Hierarchical Memory for High-Efficiency Long-Term Reasoning in LLM Agents. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vera Demberg, Kentaro Inui, and Lluís Marquez (Eds.). Association for Computational Linguistics, Rabat, Morocco, 341–350. <https://doi.org/10.18653/v1/2026.eacl-long.15>
- [85] Zhaoyan Sun, Xuanhe Zhou, and Guoliang Li. 2024. R-Bot: An LLM-based Query Rewrite System. *arXiv preprint arXiv:2412.01661* (2024).
- [86] Jinqiang Wang, Huansheng Ning, Yi Peng, Qikai Wei, Daniel Tesfai, Wenwei Mao, Tao Zhu, and Runhe Huang. 2024. A Survey on Large Language Models from General Purpose to Medical Applications: Datasets, Methodologies, and Evaluations. *arXiv preprint arXiv:2406.10303* (2024).
- [87] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. 2024. A Survey on Large Language Model Based Autonomous Agents. *Frontiers of Computer Science* 18, 6 (2024), 186345. <https://doi.org/10.1007/s11704-024-40231-1>
- [88] Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S Yu, and Qingsong Wen. 2024. Large Language Models for Education: A Survey and Outlook. *arXiv preprint arXiv:2403.18105* (2024).
- [89] Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. 2024. Knowledge graph prompting for multi-document question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 19206–19214.
- [90] Lilian Weng. 2023. LLM Powered Autonomous Agents. lilianweng.github.io. <https://lilianweng.github.io/posts/2023-06-23-agent/>
- [91] 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*. <https://openreview.net/forum?id=pZiyCaVuti>
- [92] Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Meno-lascina, and Vicente Grau. 2024. Medical Graph RAG: Towards Safe Medical Large Language Model via Graph Retrieval-Augmented Generation. *arXiv preprint arXiv:2408.04187* (2024).
- [93] Xu Wujiang, Mei Kai, Gao Hang, Tan Juntao, Liang Zujie, and Zhang Yongfeng. 2025. A-MEM: Agentic Memory for LLM Agents. *arXiv preprint arXiv:2502.12110* (2025).
- [94] Jing Xu, Arthur Szlam, and Jason Weston. 2022. Beyond Goldfish Memory: Long-Term Open-Domain Conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 5180–5197. <https://doi.org/10.18653/v1/2022.acl-long.356>
- [95] Zihan Yan, Rui Xi, and Mengshu Hou. 2025. MCTuner: Spatial Decomposition-Enhanced Database Tuning via LLM-Guided Exploration. *Proc. ACM Manag. Data* 3, 6, Article 342 (Dec. 2025), 25 pages. <https://doi.org/10.1145/3769807>
- [96] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388* (2025).
- [97] Chang Yang, Chuang Zhou, Yilin Xiao, Su Dong, Luyao Zhuang, Yujing Zhang, Zhu Wang, Zijin Hong, Zheng Yuan, Zhishang Xiang, Shengyuan

- Chen, Huachi Zhou, Qinggang Zhang, Ninghao Liu, Jinsong Su, Xinrun Wang, Yi Chang, and Xiao Huang. 2026. Graph-based Agent Memory: Taxonomy, Techniques, and Applications. arXiv:2602.05665 [cs.AI] <https://arxiv.org/abs/2602.05665>
- [98] John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. arXiv:2405.15793 [cs.SE] <https://arxiv.org/abs/2405.15793>
- [99] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. arXiv:2210.03629 [cs.CL] <https://arxiv.org/abs/2210.03629>
- [100] Haochen Zhang, Yuyang Dong, Chuan Xiao, and Masafumi Oyamada. 2024. Jellyfish: Instruction-Tuning Local Large Language Models for Data Preprocessing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 8754–8782. <https://doi.org/10.18653/v1/2024.emnlp-main.497>
- [101] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing Dialogue Agents: I Have a Dog, Do You Have Pets Too? arXiv:1801.07243 [cs.AI] <https://arxiv.org/abs/1801.07243>
- [102] Yanxin Zheng, Wensheng Gan, Zefeng Chen, Zhenlian Qi, Qian Liang, and Philip S Yu. 2024. Large Language Models for Medicine: A Survey. *International Journal of Machine Learning and Cybernetics* (2024), 1–26.
- [103] Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. MemoryBank: Enhancing Large Language Models with Long-Term Memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, 19724–19731.
- [104] Xuanhe Zhou, Guoliang Li, Zhaoyan Sun, Zhiyuan Liu, Weize Chen, Jianming Wu, Jiesi Liu, Ruohang Feng, and Guoyang Zeng. 2024. D-bot: Database diagnosis system using large language models. *Proceedings of the VLDB Endowment* 17, 10 (2024), 2514–2527.
- [105] Xuanhe Zhou, Zhaoyan Sun, and Guoliang Li. 2024. Db-gpt: Large language model meets database. *Data Science and Engineering* 9, 1 (2024), 102–111.
- [106] Yingli Zhou, Yaodong Su, Youran Sun, Shu Wang, Taotao Wang, Runyuan He, Yongwei Zhang, Sicong Liang, Xilin Liu, Yuchi Ma, and Yixiang Fang. 2026. In-Depth Analysis of Graph-Based RAG in a Unified Framework. *Proceedings of the VLDB Endowment* 18, 13 (Jan. 2026), 5623–5637. <https://doi.org/10.14778/3773731.3773738>

Prompt for Graph-based Extraction.

Entity Extraction:

之前的消息:

{上文消息}

当前消息:

{当前消息}

从上述对话中提取当前消息中明确或隐含提到的实体结点。注意: 始终将发言者/演员作为第一个结点, 并提取当前消息中提到的其他重要实体、概念或演员。

INSTRUCTIONS:

1. 始终将说话者/演员作为第一个结点。说话者是每行对话中冒号之前的部分。
2. 提取当前消息中提到的其他重要实体、概念或演员。
3. 不要为关系或动作创建结点。
4. 不要为时间信息 (如日期、时间或年份) 创建结点 (这些信息稍后将添加到边中)。
5. 结点名称应尽可能明确, 使用全名。
6. 不要仅提取提到的实体。

Relation Extraction:

之前的消息:

{上文消息}

当前消息:

{当前消息}

实体:

{实体}

根据上述消息和实体, 从当前消息中提取与列出实体相关的所有事实。请注意, 仅提取所提供实体之间的事实, 且每个事实应表示两个不同结点之间的明确关系。

INSTRUCTIONS:

1. 仅提取所提供实体之间的事实。
2. 每个事实应表示两个不同结点之间的明确关系。
3. `relation_type` 应为对事实的简洁、全大写的描述 (例如: `LOVES`, `IS_FRIENDS_WITH`, `WORKS_FOR`)。
4. 提供包含所有相关信息的更详细事实。
5. 在相关情况下, 考虑关系的时间方面。

图 12: 基于图形提取的一个样本提示。

Prompt for Answer Simplification.

请提供需要简化的问题和完整句子答案, 我将为您提取最关键的部分。

遵循以下规则:

1. 提取核心信息: 找出直接回答问题的主要信息。
 2. 删除冗余表达: 去除 “根据所提供的信息…”、“答案是…”、“根据文件…” 等表述。
 3. 省略解释: 不包含原始答案中的任何理由、推理或额外背景。
- 简洁明了。

示例:

我毕业时获得的是什么学位?

根据所提供的信息, 您获得了工商管理学位。

工商管理

请提供需要简化的问题和答案内容。

- 问题: {question}

- 原始答案: {answer}

简化答案:

图 13: 请简化回答。

A 提示模板

本节详细介绍了评估中使用的几种提示模板。基于图形的信息提取示例提示如图 12 所示, 后处理答案简化提示如图 13 所示。

B 实验详情

B.1 数据集详情

在本节中, 我们对 *LOCOMO* 和 *LONGMEMEVAL* 基准的特定评估类别进行了详细分解。

LOCOMO 在多种场景下评估智能体的记忆能力。具体的任务类别定义如下:

- **单跳**问题需要基于单一会话来回答。
- **多跳**问题需要综合来自多个不同会话的信息。
- **时间推理**问题需要进行时间推理并捕捉与时间相关的信息线索。
- **开放领域知识**问题需要将提供的信息与外部知识 (如常识或世界事实) 相结合。

LONGMEMEVAL 将记忆任务分类, 以评估以下方面:

- **信息提取 (IE)**: 从广泛的交互历史中召回特定信息的能力, 包括用户 (**单会话-用户**) 或助手 (**单会话-助手**) 提到的细节, 以及模型是否能够利用用户信息生成个性化回复 (**单会话-偏好**)。

- **多会话推理 (MR)**: 能够综合多个历史会话中的信息, 回答涉及聚合和比较的复杂问题的能力。
- **知识更新 (KU)**: 识别用户个人信息变化并随时间动态更新用户知识的能力。
- **时间推理 (TR)**: 对用户信息的时间方面具有意识, 包括交互中显式的时间提及和时间戳元数据。

B.2 数据集变体构建

LONGMEMEVAL 非常适合用于受控评估: 其具备由会话池支持的可配置上下文长度, 该会话池提供主题相似的历史会话, 而不会引入冲突信息, 且每个样本仅关联一个问题, 使得对应的真值证据清晰明确。基于这些特性, 我们构建了原始数据集 (LONGMEMEVAL_S) 的一系列变体, 以评估 **上下文可扩展性** 和 **位置敏感性**。

对于 **上下文可扩展性**, 我们构建了默认上下文长度的 50%、150% 和 200% 三种变体。具体而言, 对于 50% 的变体, 我们在会话级粒度上裁剪掉一半的历史对话, 同时保留真实会话; 对于 150% 和 200% 的变体, 我们将从会话池中采样的额外会话附加到原始对话历史的末尾。

对于 **位置敏感性**, 我们通过重新定位真实会话的位置, 生成 LONGMEMEVAL_S 的位置变化版本。具体而言, 我们从原始位置提取真实会话, 并将其重新插入对话的三个等划分段落之一: 前 1/3 (早期)、中间 1/3 (中期) 和后 1/3 (晚期)。

B.3 详细结果

在本节中, 我们展示了使用 Qwen2.5-7B-Instruct 进行 Exp.3 上下文可扩展性分析和 Exp.4 位置敏感性分析的详细结果。请注意, Zep 被排除在外, 因为它在两天内无法完成 LONGMEMEVAL。结果如表 9 和表 10 所示。

C 我们新设计方法的详细信息

本节详细描述了我们的新设计方法的完整工作流, 算法见算法 1。

如图 11 所示, 新消息首先被摄入短期记忆, 并通过先进先出 (FIFO) 队列进行管理。当短期记忆达到容量时, 最旧的消息将根据语义相似度划分为若干段, 并转移到中期记忆。在此层级中, 我们维护一个记忆树, 其中每个叶结点代表一个段——捕捉其组成消息的生成摘要——而父结点则提供其子结点的聚合摘要。这种按段粒度的处理方式相较于按轮次处理显著降低了 token 开销。对于每个段的叶结点, 我们基于访问频率和近期性计算一个热值得分; 热值得分较高的段将被提升至长期记忆。

在信息检索阶段, 我们对三个存储层级分别进行独立的检索。短期记忆被完整地检索以保持上下文连续性。对于中期记忆, 我们采用双模式检索机制: 通过基于扁平向量的相似度搜索匹配高层结点语义, 而原始消息则通过束搜索从根结点遍历, 每层选择与当前查询最相似的前- k 个结点, 最终到达存储在分段叶结点的原始消息。长期记忆通过标准的基于向量的相似度检索访问。

Algorithm 1: Our Newly Designed Method

```

input : Current query  $q$ , Short-term memory  $\mathcal{M}_S$ ,
        Mid-term memory  $\mathcal{M}_M$ , Long-term memory
        tree  $\mathcal{M}_L$ , short-term capacity threshold  $\tau$ , heat
        threshold  $\theta$ , beam width  $k$ 
output: Updated memories  $(\mathcal{M}'_S, \mathcal{M}'_M, \mathcal{M}'_L)$ , Response
         $r$ 

// (1) Information Retrieval
1  $C_S \leftarrow \mathcal{M}_S$ ; // retrieve entire short-term
   memory
2  $C_{M\_flat} \leftarrow \text{VectorSearch}(\mathcal{M}_M.\text{high-level-nodes}, q)$ ;
3  $C_{M\_beam} \leftarrow \text{BeamSearch}(\mathcal{M}_M.\text{tree}, q, k)$ ;
4  $C_L \leftarrow \text{VectorSearch}(\mathcal{M}_L, q)$ ;
5  $C \leftarrow C_S \cup C_{M\_flat} \cup C_{M\_beam} \cup C_L$ ; // aggregate
   retrieved context
// (2) Response Generation
6  $r \leftarrow \text{LLM\_Generate}(q, C)$ ;
// (3) Memory Ingestion
7  $\mathcal{M}'_S \leftarrow \text{Enqueue}(\mathcal{M}_S, q, r)$ ; // ingest new
   messages via short-term FIFO queue
8 if  $|\mathcal{M}'_S| > \tau$  then
9    $\mathcal{M}_{old} \leftarrow \text{DequeueOldestHalf}(\mathcal{M}'_S)$ ;
10   $\mathcal{S}_{new} \leftarrow \text{SegmentBySemanticSimilarity}(\mathcal{M}_{old})$ ;
   // partition into segments
11   $\mathcal{M}'_M \leftarrow \text{UpdateMemoryTree}(\mathcal{M}_M, \mathcal{S}_{new})$ ;
   // leaf: segment, parents: aggregated
   summary
12 else
13    $\mathcal{M}'_M \leftarrow \mathcal{M}_M$ ;
14   $\mathcal{M}'_L \leftarrow \mathcal{M}_L$ ;
15  foreach leaf node  $n \in \mathcal{M}'_M.\text{leaves}$  do
16     $\text{score} \leftarrow \text{ComputeHeatScore}(n.\text{freq}, n.\text{recency})$ ;
17    if  $\text{score} > \theta$  then
18       $\mathcal{M}'_L \leftarrow \mathcal{M}'_L \cup \{n\}$ ; // promote high-heat
      segments
19 return  $(\mathcal{M}'_S, \mathcal{M}'_M, \mathcal{M}'_L, r)$ 

```

D 更多课程与机遇

经验教训:

► **L4**. 设计精良的内存框架将推理与模型规模解耦, 使较小的大模型能够有效处理涉及时间依赖性等复杂查

表 9: 在不同上下文规模下对 LONGMEMEVAL 方法的比较。

Method	Scale	IE		MR		TR		KU		Overall	
		F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
A-MEM	50%	41.54	35.29	17.82	15.63	20.67	16.12	24.24	20.98	26.98	22.72
	100%	39.70	33.23	16.61	14.32	19.37	15.54	25.59	21.63	25.95	21.84
	150%	37.92	31.45	15.48	13.92	20.08	17.51	21.43	18.91	24.63	21.11
	200%	36.68	30.12	16.81	14.15	20.94	16.39	21.87	18.42	24.89	20.44
MemoryBank	50%	42.04	34.88	19.06	15.65	23.62	18.08	32.93	26.53	29.61	24.11
	100%	41.19	29.39	19.41	16.42	17.80	13.15	31.51	27.31	27.65	23.08
	150%	42.37	34.52	15.62	12.93	18.57	12.56	27.64	22.06	26.63	21.03
	200%	41.56	33.74	16.69	13.35	17.43	12.83	31.02	24.52	26.85	21.32
MemGPT	50%	47.61	43.47	26.89	25.64	28.60	22.79	40.04	36.07	35.86	32.07
	100%	45.08	37.89	18.61	16.71	21.65	14.81	28.09	24.85	29.17	24.08
	150%	42.89	36.83	20.45	19.12	22.48	22.01	23.37	16.46	28.45	25.01
	200%	43.75	36.73	17.77	14.95	25.97	20.66	24.79	21.94	29.15	24.31
MemO	50%	45.43	37.99	27.03	25.14	28.10	20.93	45.74	39.95	35.97	30.56
	100%	39.17	39.10	23.02	20.72	28.86	21.19	41.28	36.37	32.44	29.58
	150%	36.88	31.09	19.59	16.91	27.50	20.55	44.32	38.84	30.94	26.37
	200%	40.60	34.21	23.37	20.70	29.20	21.92	38.76	32.78	32.69	27.20
MemO ^g	50%	34.06	32.24	26.01	23.83	28.65	29.73	46.31	48.06	32.38	31.81
	100%	32.37	30.86	24.42	22.04	29.16	21.60	40.43	35.51	29.66	26.83
	150%	29.53	24.05	19.71	16.61	26.29	18.54	33.98	30.37	26.75	21.12
	200%	32.31	26.42	21.44	19.25	29.33	20.84	39.56	34.97	29.75	24.44
MemoChat	50%	12.49	9.01	9.63	7.57	19.37	12.78	5.92	4.18	12.53	8.88
	100%	10.36	6.05	12.27	10.09	17.49	11.35	6.51	4.28	12.16	8.26
	150%	12.52	7.92	11.94	10.44	16.85	9.96	3.29	1.75	12.08	8.17
	200%	11.08	6.72	10.70	9.35	19.04	12.44	6.48	4.11	12.38	8.53
MemTree	50%	50.42	41.71	26.59	24.26	29.19	21.93	51.66	46.73	38.63	32.59
	100%	53.30	44.72	21.96	19.88	29.97	21.77	41.49	38.56	36.92	31.05
	150%	46.67	38.81	26.01	24.00	25.38	17.96	43.13	38.32	34.96	29.25
	200%	52.98	44.22	27.04	24.71	25.69	18.41	47.34	42.72	37.94	31.93
MemoryOS	50%	51.78	43.82	22.53	20.77	23.72	17.81	39.09	35.26	34.56	29.44
	100%	49.91	43.15	19.35	17.80	26.34	21.61	30.62	27.97	32.50	28.31
	150%	50.32	42.78	21.13	19.95	22.93	17.73	34.81	31.16	32.85	28.23
	200%	44.55	36.92	21.30	19.22	21.93	17.70	38.98	36.01	31.48	26.79
MemOS	50%	50.35	38.11	22.97	19.16	21.04	14.99	37.38	33.18	33.24	26.15
	100%	49.46	40.52	22.81	19.72	21.34	15.65	33.99	27.75	32.48	26.38
	150%	43.33	32.93	20.76	17.52	21.89	15.14	22.68	17.05	28.40	21.62
	200%	37.08	28.68	20.12	16.83	24.84	16.33	27.76	20.76	27.86	21.01

询。例如，在框架中采用多步推理或专用组件，可使较小的大模型更高效地应对复杂任务。

► **L5.** 与其采用破坏性更新，内存系统应采用非破坏性策略，在保留历史信息的同时标注其有效性，以实现未来的重用并防止可能有用的知识的损失。换句话说，系统不应删除旧信息，而应予以保留并标记其状态。

机遇：

► **O4.** 在复杂的查询场景中，依赖单一固定的检索策略可能较为脆弱，容易导致任务失败。不同的查询通常需要不同的检索粒度和机制。然而，现有的记忆系

统并未考虑这一问题，因此设计一种检索路由规划器，能够根据多样化的记忆-查询上下文动态选择并适应不同的检索策略，是一个有趣的研究问题。

► **O5.** 现有的记忆基准，如 LOCOMO 和 LONGMEMEVAL，是在预先收集的、静态的交互历史数据上评估记忆方法，无法反映现实世界记忆的连续性和动态演变特性。它们也未能捕捉人类记忆的关键属性，例如对近期更新的偏好以及反复提及信息的强化。此外，尽管 LOCOMO 中包含了稀疏的视觉元素，这些基准仍严格以文本为中心。开发更具挑战性、以交互驱动且全面的

表 10: LONGMEMEVAL 数据集上的位置鲁棒性评估

Method	IE						MR		TR		KU		Overall	
	user		assistant		preference									
	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
Early Evidence Sessions														
A-MEM	41.62	37.81	42.49	36.54	8.74	0.73	10.48	8.15	20.31	17.65	28.61	24.59	23.76	20.13
MemoryBank	46.01	38.36	48.08	39.74	17.09	5.71	16.95	13.32	21.60	16.34	31.25	24.93	27.98	21.94
MemGPT	54.09	48.87	50.27	43.05	10.34	1.23	24.39	21.94	29.58	22.50	35.74	31.51	33.75	28.47
Mem0	63.86	57.57	33.98	27.04	10.93	0.40	25.13	22.65	29.78	21.66	34.98	30.81	33.46	27.71
Mem0 ^g	53.95	48.70	19.62	14.92	9.93	0.75	18.31	15.90	27.88	20.12	41.37	37.03	29.09	23.89
MemoChat	6.44	3.24	16.95	12.74	7.18	0.12	12.27	10.09	17.49	11.35	6.51	4.28	12.16	8.26
MemTree	52.85	48.92	49.26	37.66	9.31	0.18	26.18	23.72	27.54	20.39	40.83	36.40	34.13	28.49
MemoryOS	52.19	49.58	61.32	52.71	12.40	1.18	19.35	17.80	26.34	21.61	30.62	27.97	31.85	27.76
MemOS	60.65	50.63	43.00	36.78	11.20	0.16	19.85	17.02	22.12	16.45	25.37	19.02	29.10	23.09
Middle Evidence Sessions														
A-MEM	40.23	39.12	43.51	37.98	6.15	0.48	9.87	7.41	18.92	14.71	29.45	25.66	23.13	19.65
MemoryBank	41.89	35.15	52.13	44.49	15.44	5.16	16.35	13.43	25.15	19.73	32.12	26.78	28.68	23.21
MemGPT	49.83	45.29	48.63	41.72	10.56	2.07	24.38	22.01	26.77	20.33	35.85	31.82	32.25	27.36
Mem0	66.46	60.15	34.52	28.05	11.29	1.01	24.82	21.91	28.07	20.26	42.37	36.83	34.53	28.59
Mem0 ^g	48.88	43.93	17.02	12.76	10.52	0.81	19.48	17.06	27.70	19.83	45.99	40.50	29.10	23.76
MemoChat	9.43	6.96	19.37	15.06	7.71	0.14	13.08	10.98	16.48	10.75	3.42	1.33	12.35	8.66
MemTree	60.03	54.29	60.82	49.21	10.58	0.69	27.39	24.75	26.25	19.35	40.60	36.58	36.45	30.59
MemoryOS	51.27	47.13	63.14	52.26	10.90	1.07	20.39	18.05	23.80	18.45	35.21	32.32	32.15	27.27
MemOS	64.24	57.41	49.26	38.28	11.97	0.50	24.15	19.73	23.58	17.95	27.22	24.14	32.17	26.14
Late Evidence Sessions														
A-MEM	50.84	47.21	41.27	32.45	9.68	0.82	12.03	9.56	24.63	18.72	28.34	27.52	26.49	22.11
MemoryBank	45.57	38.11	50.63	43.63	14.64	5.14	15.60	12.31	26.21	19.54	27.57	21.96	28.35	22.43
MemGPT	54.13	49.02	57.29	50.27	10.56	1.31	23.01	20.69	29.67	23.30	34.57	30.91	34.03	29.09
Mem0	64.06	58.32	35.17	28.08	10.93	0.40	25.71	22.07	30.37	22.89	42.04	45.81	35.04	30.44
Mem0 ^g	63.48	59.40	17.91	14.34	11.01	0.84	24.31	25.28	28.61	20.06	38.62	35.76	31.66	27.61
MemoChat	8.59	6.09	20.12	15.46	8.78	0.22	11.15	9.61	23.33	16.16	4.79	2.79	13.90	9.89
MemTree	63.59	57.77	58.18	47.98	9.72	0.93	27.45	23.73	29.59	21.18	39.71	36.07	37.37	31.09
MemoryOS	49.13	44.71	63.44	53.51	13.58	2.54	22.18	19.69	24.26	18.13	38.39	34.85	33.14	27.90
MemOS	69.88	49.34	51.30	38.25	11.86	0.18	23.46	15.87	26.25	17.80	31.62	22.41	34.40	23.65

多模态基准，更真实地反映记忆形成与使用场景，是一个有意义的研究方向。