MemoBrain：作为推理智能体大脑的执行记忆

Hongjin Qian[1],**Zhao Cao**[2],**Zheng Liu**[1*1] 北京智源人工智能研究院[2] 中国人民大学高瓴人工智能学院
{chienqhj,zhengliu1026}@gmail.com

arXiv:2601.08079v1 [cs.AI] 12 Jan 2026

摘要

工具增强型智能体框架中的复杂推理本质上具有长时程特征，推理轨迹和短暂的工具产物会不断累积，给大语言模型有限的工作上下文带来压力。若没有显式的记忆机制，这种累积会破坏逻辑连贯性并削弱任务对齐。因此，记忆并非仅是辅助性的效率问题，而是支撑长时程连贯、目标导向推理的核心组成部分。

我们提出**MemoBrain**，一种面向工具增强型智能体的执行记忆模型，可围绕推理步骤构建依赖感知记忆，捕捉关键中间状态及其逻辑关系。MemoBrain作为副驾驶与推理智能体协同工作，在不阻碍执行的情况下整理推理进展，并主动管理工作上下文。具体而言，它会剪除无效步骤、折叠已完成的子轨迹，并在固定的上下文预算下保留紧凑且显著性高的推理主干。通过这些机制，系统能够对推理轨迹实施显式认知控制，而非被动累积上下文。

我们在多个具有挑战性的长时程基准测试上评估MemoBrain，包括 GAIA、Web-Walker 和 BrowseComp-Plus，结果表明，相较于强基线，它始终带来提升。我们的代码位于此代码库。

1 引言

近年来，大语言模型（LLM）迅速发展，使推理、决策和信息寻求领域的应用日益复杂（Zhao 等人，2024；Wang 等人，2024）。工具增强型智能体框架通过交替进行推理并调用搜索等外部工具，进一步拓展了这些能力引擎和代码执行器（Zhang 等人，2025a；Qian 等人，2025a）。然而，长时程的工具集成交互不可避免地会累积上下文信息，包括中间推理轨迹和工具生成的产物（Ye 等人，2025年）。随着上下文不断增长，大语言模型有限的工作上下文会不堪重负，使短暂信息占据主导地位，并分散对任务相关证据的注意力，进而导致认知过载和任务对齐度下降（Hu 等人，2025年）。

因此，引入显式记忆机制已被广泛视为将基于智能体的框架扩展到复杂、长时程任务中的关键因素（Zhang 等人，2025b）。现有的智能体记忆主要有两种形式。跨任务记忆侧重于跨任务持久保存信息，积累经验、偏好和用户特定知识，以支持个性化、适应以及从过往交互中学习（Gao 等人，2025年）。长期记忆则旨在保留并组织长时间交互中的信息，例如长期对话或大型文档集合，通常通过摘要、参数化存储或基于图的表示来实现（Xu 等人，2025年；Yan 等人，2025a；Yu 等人，2025年；Kang 等人，2025年）。尽管这两种形式在保留和调用既有信息方面成效显著，但它们都无法原生适应复杂推理场景；在此类场景中，记忆必须主动追踪推理进展、捕捉中间结论之间的依赖关系，并管理信息流，以维持连贯性和任务对齐。

在这项工作中，我们聚焦于一种独特的记执行记忆。**执行记忆是一种任务专用的进程内记忆机制，专为工具增强型智能体系统中的复杂推理而设计。它为每项任务从头初始化，并随推理轨迹在线演化**。与跨任务记忆不同，它不会在任务完成后继续保留

*通讯作者。

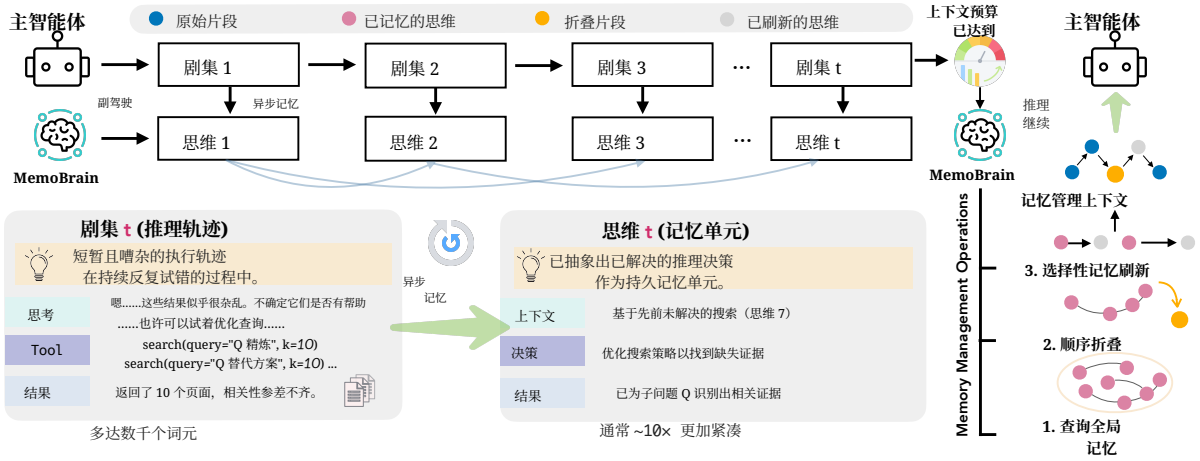


图 1：MemoBrain 将瞬时推理会话异步抽象为紧凑且具备依赖感知能力的记忆单元，并维护全局推理记忆。达到记忆预算上限时，它会通过折叠或刷新过往信息来构建记忆管理上下文，从而实现连贯的长时程推理。

，也不会编码持久先验。与长期记忆不同，它不仅仅是逐步维护的条件上下文，而是一种主动管理的表示，能够适应推理过程不断演变的结构。从工程角度看，执行记忆充当显式的上下文控制机制，在有界上下文预算下，以即时方式向推理智能体提供与任务相关的信息。从认知角度看，它发挥执行性角色，对推理步骤之间的逻辑依赖关系进行建模，维护对任务进展的全局层次化视图，并指导后续行动，以保持连贯、目标导向的推理。

越来越多的近期研究强调了在复杂推理任务中管理规模庞大且不断演变的上下文的重要性。代表性方法通过抽象或压缩中间推理步骤来降低上下文开销（Wu 等人，2025c；Ye 等人，2025年），另一些方法则将任务分解为辅助子任务，并对其结果进行总结后重新整合（Li 等人，2025d，c）。这些方法在长程推理基准测试中展现出显著的实证收益，凸显了从工程角度看记忆的实际价值。然而，大多数现有方法将记忆视为一种以提升效率为导向的上下文缩减机制，因此记忆在全局任务对齐、轨迹级控制以及推理过程的认知建模方面基本处于被动状态（Hu 等人，2025年）。

这些观察表明，在有界上下文预算下维持与任务对齐的推理，需要对信息流进行明确控制。因此，我们将记忆视为一种框架级组件，与推理智能体协同工作，而非单个模型的内部副产物。基于这一视角，我们提出**MemoBrain，这是一种执行记忆模型，以类似副驾驶的方式与推理智能体协同运作**。MemoBrain 会在完整的推理轨迹上维护全局记忆，并对工作上下文实施执行控制，及时提供与任务相关的信息，同时抑制不必要的认知负荷。通过控制每一步所呈现的信息，MemoBrain 使得在有界上下文预算下进行长时程的目标导向推理成为可能。

如图1所示，MemoBrain 围绕两个互补的过程来组织记忆：记忆构建和记忆管理。在记忆构建过程中，已完成的推理剧集会从临时的执行会话中逐步抽象出来，形成紧凑的记忆单元，称为思维单元。每个思维单元都编码了所处理的子问题、使用的信息或工具，以及由此产生的结果，并通过明确的依赖关系与先前的思维单元相连。这个过程生成了结构化且具备依赖关系感知能力的轨迹记忆，在保留长时程推理逻辑骨干的同时，舍弃短暂的执行产物。

基于这一全局轨迹记忆，Mem-oBrain 为推理智能体执行主动上下文管理。在固定的上下文预算下，它会选择性地折叠已完成的子轨迹，刷新低效用的记忆元素，并优先保留高显著性的推理骨干。这些操作共同确保工作上下文保持紧凑，同时在逻辑上足够充分，使智能体能够

随着推理范围扩大，持续进行连贯且与任务一致的推理。

为优化记忆构建与记忆管理以满足其不同的功能需求，MemoBrain 采用任务特定的优化策略，使学习目标与下游推理行为保持一致，从而使记忆能够作为长时程推理中的活动且自适应组件发挥作用。我们将 MemoBrain 集成到具备强大长时程推理能力的工具增强型智能体中，在 4B 至 14B 的模型规模范围内对其进行评估。在包括 BrowseComp-Plus（Chenet al.，2025年）、GAIA（Mialon et al.，2023）和 WebWalker（Wu etal.，2025b）在内的各项具有挑战性的基准测试中，MemoBrain 在有界上下文预算下的复杂信息检索与推理任务上始终能够提升性能。

我们的贡献有三点：（1）我们提出了执行记忆这一全新的记忆范式，专用于工具增强型智能体框架中的复杂推理，将记忆形式化为有界上下文预算下用于上下文管理和认知建模的显式控制机制。（2）我们提出了*MemoBrain*，这是一种独立执行记忆模型，与推理智能体协同运行，构建依赖感知的任务内记忆，并通过显式记忆操作主动管理工作上下文。（3）我们通过实验证明，将执行记忆集成到不同的工具增强型智能体框架中，能够在多个模型规模和具有挑战性的基准测试上始终提升性能，凸显其在不同智能体生态系统中的通用性与有效性。

2 相关工作

记忆已成为扩展大语言模型（Packer 等人，2023；Zhang 等人，2025b；Chhikara 等人，2025年）的一项基础机制，尤其适用于涉及长上下文和复杂信息访问的场景（Hu 等人，2025年）。大量研究聚焦于**智能体长期记忆**，即为模型配备机制，使其能够在智能体框架中跨越长时间的交互保留并组织信息（Zhang 等人，2025a）。在这些场景中，记忆通常基于长文档、检索段落集合或多轮对话历史构建，并通过结构化抽象形式实现，例如基于图的组织、键值压缩，或分层存储（Zhong 等人，2024；Wang 等人，2023；Chhikara 等人，2025年；Edge 等人，2024；Qian 等人，2025b；Kang 等人，2025年；Yan 等人，2025a）。这类记忆机制主要旨在支持知识的持续积累与复用，为对话智能体、长上下文问答和交互式助手等任务提供长期先验（Li 等人，2025e；Yan 等人，2025b；Xu 等人，2025年）。因此，这类机制通常通过 LoCoMo（Maharana 等人，2024）、LongBench（Bai 等人，2025年）和 RULER（Hsieh 等人，2024）等基准进行评估。在这些基准中，记忆基于提供的交互历史或长上下文构建，并在推理过程中被访问。

在工具增强的复杂推理中，已有研究强调了**上下文管理**(Mei 等人, 2025年）的重要性。AgentFold 和 ContextFolding 等方法通过折叠或抽象冗余推理步骤来减少上下文开销（Ye 等人, 2025年; Sun 等人, 2025年），而 DeepAgent 等方法则将子任务委派给辅助智能体，并仅重新整合摘要后的结果（Li 等人, 2025d,c）。尽管这些方法效果显著，但它们侧重于记忆的工程层面，并未将其建模为用于全局任务感知或推理引导的认知控制机制。相比之下，我们将工具增强推理中的记忆形式化为**执行记忆**，这是一种在线机制，用于构建推理轨迹的结构化表示，并明确控制信息流。通过将上下文管理与认知建模相结合，执行记忆能够实现连贯的长时程推理，超越纯粹工程层面的优化。

3 方法

3.1 MemoBrain 概述

考虑一个复杂推理任务 $\mathcal{X}$，它需要一系列推理片段 $\{x_1, x_2, \ldots, x_T\}$来收集充分的证据并给出最终答案 $\mathcal{Y}$。如图1所示，每个推理片段 x_t 都是一个自成一体的进展单元，通常包括任务推理、与外部工具（如网络搜索或代码执行）的交互，以及工具反馈，这些共同产生多样且不断演变的执行轨迹。

记忆构建。 MemoBrain 将推理过程序列逐步转化为结构化的轨迹记忆。每个已完成的过程 x_t 都会被抽象为一个精简的记忆单

元，称为 思维 v_t，用于捕捉其对任务进展的语义贡献。随着推理的进行，新生成的思维单元会被整合进一个全局的依赖感知记忆结构中，

$$\mathcal{G}_t = \mathcal{G}_{t-1} \cup \{v_t\}, \tag{1}$$

从而形成不断演化的记忆图 $\mathcal{G}$，它表示超越瞬时执行细节的高层推理轨迹。

记忆管理。 推理代理在有界上下文预算下运行。

当上下文积累接近预算上限时，MemoBrain 会启动执行性记忆管理流程，以调节已积累的推理结构如何得到维护和呈现。从高层次来看，MemoBrain 以当前轨迹记忆 $\mathcal{G}_t$ 为条件，并调用记忆模型 $\phi(\cdot)$来生成一组执行操作：

$$\mathcal{O}_t = \phi(\mathcal{G}_t), \tag{2}$$

这些操作规定了应如何重新组织轨迹记忆的各个部分。应用这些操作后，将得到更新后的记忆状态：

$$\mathcal{G}_{t+1} = \mathrm{Apply}(\mathcal{G}_t, \mathcal{O}_t). \tag{3}$$

基于 $\mathcal{G}_{t+1}$，MemoBrain 会重新组织工作上下文，以保留一个紧凑但在语义上足够充分的表示，供后续推理使用。

3.2 记忆构建

思维形成。 推理过程完成后，MemoBrain 会将其抽象为一个思维单元。每个推理过程 x_t 表示为：

$$x_t = (\tau_t, \omega_t), \tag{4}$$

其中 τ_t 表示暂时性的执行层面信息（例如工具调用和原始输出），而 ω_t表示推动任务进展的已解析语义结果。MemoBrain 不保留执行细节，而是专注于捕获 ω_t的任务相关贡献。在当前记忆状态 $\mathcal{G}_{t-1}$ 以及专门的记忆指令（见表3）的条件下，MemoBrain 应用：

$$v_t = \phi(x_t, \mathcal{G}_{t-1}), \tag{5}$$

其中， $\phi(\cdot)$会将已完成的剧集映射为执行层抽象，从而捕捉使用了哪些信息来解决哪个子问题，以及取得了什么结果。每个思维都编码了该剧集在整体推理轨迹中的功能角色，在保留其对任务进展贡献的同时，舍弃短暂的执行过程产物。新形成的思维 v_t随后通过将全局状态从 $\mathcal{G}_{t-1}$ 更新为 $\mathcal{G}_t$而整合进记忆中。该过程与推理智能体并行异步运行，使 MemoBrain 能够充当副驾驶，持续记录并组织推理进展，而不会中断推理执行。

每个新创建的思维单元都会被初始化为活动状态，表示其参与当前推理轨迹。思维单元可以处于两种激活状态之一：活动或非活动，并可在记忆管理期间进行更新。这样，无需移除已完成、已被取代或无益的推理路径在记忆图中的结构性角色，即可将其停用。

依赖建模。 为保留推理的逻辑结构，MemoBrain 对新思维单元如何建立在现有思维单元之上进行建模：

$$\mathrm{Dep}(v_t) \subseteq \{v_1, \ldots, v_{t-1}\}, \tag{6}$$

表示当前思维单元所依赖的先前结论。这些依赖关系编码了超越单纯时间顺序的因果与语义结构。

随着推理的推进，累积的思维单元及其依赖关系形成不断演化的有向记忆图 $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$，其中 $\mathcal{V}_t = \mathcal{V}_{t-1} \cup \{v_t\}$和 $\mathcal{E}_t = \mathcal{E}_{t-1} \cup \{(v_i, v_t) \mid v_i \in \mathrm{Dep}(v_t)\}$。该图提供了推理过程的全局轨迹级表示，超越了智能体的局部上下文窗口。

3.3 记忆管理

记忆管理规定了如何将执行操作应用于轨迹记忆，以便在上下文受限的情况下调控推理结构的持久性与可见性。

在每个管理步骤中，MemoBrain 都会生成一组执行操作 $\mathcal{O}_t$，作用于当前的轨迹记忆 $\mathcal{G}_t$，并由管理提示词指引（见表4）。每项 $\mathcal{O}_t$ 操作都会作用于选定的节点或子图，并据此重写记忆结构。本文考虑两类操作：

$$\mathcal{O}_t \subseteq \{\textsc{Fold}(\cdot), \textsc{Flush}(\cdot)\}. \tag{7}$$

序列轨迹折叠。FOLD 操作针对的是其关联子任务已得到明确解决的子轨迹。形式上，MemoBrain 会识别一个连通子图$\mathcal{T}_{i:j} \subset \mathcal{G}_t$，该子图由一系列思维单元$\{v_i, \ldots, v_j\}$组成，这些思维单元共同处理同一个子问题。如果终结思维单元v_j编码了明确的结果（例如，已核实目标信息，或确认始终不存在证据），那么$\mathcal{T}_{i:j}$中的中间推理过程便不再带来额外的信息增益。在这种情况下，MemoBrain 会执行折叠操作：

$$\mathcal{T}_{i:j} \Rightarrow \bar{v}, \tag{8}$$

将整个子轨迹折叠为一个摘要思维节点$\bar{v}$，记录已解决的结论。$\mathcal{T}_{i:j}$中的所有思维单元都会从当前推理上下文中移除。由于子任务已得到明确解决，因此可以安全地删除相应的推理过程，而不会损害任务相关语义。

选择性记忆刷新。 FLUSH 操作针对的是相对于当前推理状态而言信息效用已失效的单个思维单元。具体而言，MemoBrain 会识别出符合以下任一类别的思维单元v_k：（i）未能产生有意义反馈的无效推理步骤；（ii）已被后续改进取代的探索性尝试；或（iii）其结论已与当前轨迹无关的步骤。

与折叠不同，清除操作并不假设子任务已解决。MemoBrain 不会将此类节点彻底移除，而是采用以下方式：

$$v_k \Rightarrow \hat{v}_k, \tag{9}$$

将v_k替换为一个精简思维节点$\hat{v}_k$，在舍弃执行级细节的同时保留高层结构信息。这些精简思维节点会记录先前尝试的存在及其结果，从而避免冗余探索，同时避免增加不必要的上下文负载。

上下文重组。 执行记忆转换后，MemoBrain 会通过将更新后的轨迹记忆$\mathcal{G}_{t+1}$投射到新的上下文C_{t+1}中，重组工作上下文：

$$C_{t+1} = \psi(\mathcal{G}_{t+1}), \tag{10}$$

其中，$\psi(\cdot)$将$\mathcal{G}_{t+1}$中的活动思维单元映射到相应的上下文表示。对于对于折叠后的轨迹，C_t中所有相关的推理步骤都会被其摘要思维节点$\bar{v}$替代。对于已刷新的节点，冗长的执行轨迹会被精简思维节点$\hat{v}$取代。由此得到的上下文比C_t小得多，同时保留了连贯且目标导向推理所需的逻辑结构。

3.4 优化

MemoBrain 采用两阶段训练策略进行优化，以反映记忆构建和记忆管理各自不同的学习特性。两项功能均由单一记忆模型ϕ实现，并通过提示条件化，使其能够以不同角色运行。

阶段一：记忆构建。 记忆构建被视为一种高频语义抽象问题。由于这一过程主要涉及从执行轨迹中提取显著结果，且决策歧义较低，我们使用由强大的教师模型（DeepSeek V3.2）生成的高质量思维标注，对ϕ进行监督微调：

$$\mathcal{L}_{\text{const}} = -\mathbb{E}_{x_t} \log \phi(v_t \mid x_t). \tag{11}$$

这一阶段使 MemoBrain 具备稳定且一致的抽象能力，为下游记忆控制奠定基础。

第二阶段：记忆管理。 记忆管理是在有界上下文预算下，对轨迹记忆$\mathcal{G}_t$进行的一种稀疏、由决策驱动的过程。在每个触发点，MemoBrain 会根据当前记忆状态预测一组记忆操作$\mathcal{O}_t$（例如 FOLD 或 FLUSH）。由于这些决策涉及压缩与语义充分性之间的全局权衡，直接监督并不充分。

因此，我们采用直接偏好优化（Rafailov 等人，2024）来优化记忆管理。对于每个记忆状态$\mathcal{G}_t$，我们会采样多个候选操作集，评估其下游推理表现，并构建偏好对$(\mathcal{O}^+, \mathcal{O}^-)$：

$$\mathcal{L}_{\text{mgmt}} = -\mathbb{E} \log \sigma\left(\log \frac{\phi(\mathcal{O}^+ \mid \mathcal{G}_t)}{\phi(\mathcal{O}^- \mid \mathcal{G}_t)}\right). \tag{12}$$

4 实验

4.1 数据集和基线

数据集。 我们在三个基准测试上评估 MemoBrain 和基线方法：（1）**GAIA** (Mialon 等人，

方法	通用人工智能助手				WebWalkerQA			
	L1	L2	L3	Avg.	Easy	Med.	Hard	Avg.
直接推理（不使用检索）								
QwQ-32B	25.6	9.6	16.7	16.5	7.5	2.1	3.8	4.0
GPT-4o	23.1	15.4	8.3	17.5	6.7	6.0	4.2	5.5
DeepSeek-R1-671B	43.6	26.9	8.3	31.1	5.0	11.8	11.3	10.0
检索增强生成								
基础检索增强生成 (QwQ-32B)	33.3	36.5	8.3	32.0	36.9	26.1	33.5	31.2
查询规划 (QwQ-32B)	48.7	25.0	8.3	32.0	28.8	35.7	30.8	32.5
迭代检索增强生成 (QwQ-32B)	51.3	28.8	8.3	35.0	29.4	32.9	31.3	31.5
工具集成推理								
ReAct（QwQ-32B）	48.7	34.6	16.7	37.8	35.6	29.1	13.2	24.1
ReAct（GPT-4o）	51.2	34.6	8.3	34.6	34.6	42.0	23.9	33.8
ReAct（Qwen3-30B-A3B）	48.7	26.9	8.3	33.0	26.3	27.5	21.7	25.2
WebThinker-32B †	56.4	50.0	16.7	48.5	58.8	44.6	40.4	46.5
WebDancer（QwQ-32B） †	56.4	48.1	25.0	46.6	49.4	55.0	29.6	43.2
ReSum-GRPO †	–	–	–	48.5	–	–	–	–
DeepAgent-RL †	66.7	59.6	25.0	58.3	–	–	–	–
AgentFold-30B-A3B †	–	–	–	67.0	–	–	–	–
GLM-4.6	76.9	59.6	33.3	63.1	64.4	62.9	48.8	58.2
DeepResearch-30B-A3B	79.5	67.3	41.7	68.9	72.5	71.8	61.3	68.2
MemoBrain-8B								
搭配 GLM-4.6	79.5	71.2	50.0	71.8	68.8	69.6	61.3	66.5
搭配 DeepResearch-30B-A3B	**82.1**	**69.2**	**58.3**	**74.5**	**73.1**	**72.1**	**64.2**	**69.6**

表 1：主要实验结果。最佳分数以**粗体**显示，次佳分数以下划线标出。我们报告 Pass@1。标有 † 的结果引自原始论文。MemoBrain 的结果通过将 MemoBrain-8B 集成到相应的基础智能体（GLM-4.6 和 DeepResearch-30B-A3B）中获得。

LLM	Acc	搜索调用
Sonnet 4	37.35	9.03
Kimi-K2	35.42	11.22
oss-120b-high	44.10	18.35
GLM-4.6	48.19	25.60
DeepResearch-30B-A3B	51.93	17.14
MemoBrain		
搭配 GLM-4.6	55.06	24.62
搭配 DeepResearch-30B-A3B	**60.36**	**27.12**

表 2：BrowseComp-Plus 上采用相同检索设置时的性能比较

2023）（通用人工智能助手）包含 450 多个真实世界查询，涉及多步推理、工具使用和多模态理解。遵循先前的工作（Ye 等人，2025年；Wu 等人，2025c；Li 等人，2025c），我们在 103 道纯文本验证题上进行评估。（2）**WebWalker** (Wu 等人，2025b）包含跨领域的 680 个查询，要求智能体遍历相互链接的网页，并整合分散在多个子页面中的证据。

对于 GAIA 和 WebWalkerQA，谷歌搜索被用作在线搜索工具。

(3) **BrowseComp-Plus** (Chen 等，2025年）源自 BrowseComp 基准（Wei 等，2025年），包含 830 道问题，其答案简短且可验证。对于 BrowseComp-Plus，系统提供了一个固定的候选网页语料库，智能体将该语料库上的本地稠密检索作为搜索工具。

基线。我们将 MemoBrain 与涵盖三类的代表性基线进行了比较评估。直接推理包括不使用外部工具的强大独立大语言模型，例如 QwQ-32B、GPT-4o 和 DeepSeek-R1-671B（DeepSeek-AI，2025；Yang 等，2025；OpenAI，2024）。检索增强生成涵盖将检索到的证据注入上下文的方法，包括基础检索增强生成，以及结合查询规划或迭代优化的增强变体（Shao 等，2023；Chan 等，2024）。工具集成推理则包括将推理与检索操作交替进行的方法，例如 ReAct、Search-o1 和 WebThinker（Yao 等，2023；Li 等，2025b、2025d），以及近期的 WebDancer、ReSum、AgentFold 和 DeepAgent（Wu 等，2025a、2025c；Ye 等，2025；Li 等，2025c）。值得注意的是，ReSum、AgentFold 和 DeepAgent 引入了明确形式的上下文管理，例如摘要、轨迹压缩或学习式控制，这些代表了工程实现

现有智能体框架中的智能体记忆 s.

我们进一步纳入 GLM-4.6 和 DeepResearch- 30B-A3B，作为强大的工具增强型智能体基线（Li 等人，2025a；GLM 等人，2024）。Memo- Brain 被集成到这些智能体中，作为外部记忆组件。实现细节见附录A.1。

4.2 主要结果

我们在表1和表2中总结了主要实验结果，由此可以得出若干一致的观察结论。

(1) **MemoBrain 与 GLM-4.6 和 DeepResearch-30B-A3B 集成时，均能持续提升其基础智能体的性能**。在所有评测基准测试中，MemoBrain 相较于原始智能体框架均取得了显著提升，表明引入明确的执行记忆组件普遍有利于工具增强推理。尤其是与 DeepResearch-30B-A3B 结合时，MemoBrain 在所有基准测试中均取得最佳整体表现，优于所有基线。这些结果表明，MemoBrain 能适配不同的推理智能体，并且无需针对特定架构进行定制，即可作为插件式记忆模块有效集成。

(2) MemoBrain 在**更具挑战性的推理场景中带来更大幅度的提升**，尤其是在 BrowseComp-Plus 和 GAIA 的 Level-3 子集中；在这些场景中，长时程推理和大量工具交互会给推理智能体的原始上下文窗口带来压力。通过显式建模推理结构，并及时提供任务相关上下文，MemoBrain 缓解了认知过载，使智能体能够进行更深入的探索。表2也进一步印证了这一点：经 MemoBrain 增强的智能体能够更有效地使用工具，从而取得明显提升；这表明在上下文限制下，它们的工具使用更有效且更持久。

（3）与 DeepAgent 和 AgentFold 等以往的上下文管理方法相比，**MemoBrain 通过将记忆与推理智能体解耦，表现得更加有效**。现有方法将折叠或摘要直接嵌入单个智能体中，使记忆控制与推理执行相互纠缠。相比之下，MemoBrain 采用副驾驶式设计，引入显式的执行记忆，从外部管理推理轨迹，从而取得更强的实证性能，并在各种工具增强型智能体框架中提供更灵活、可复用的记忆模块。

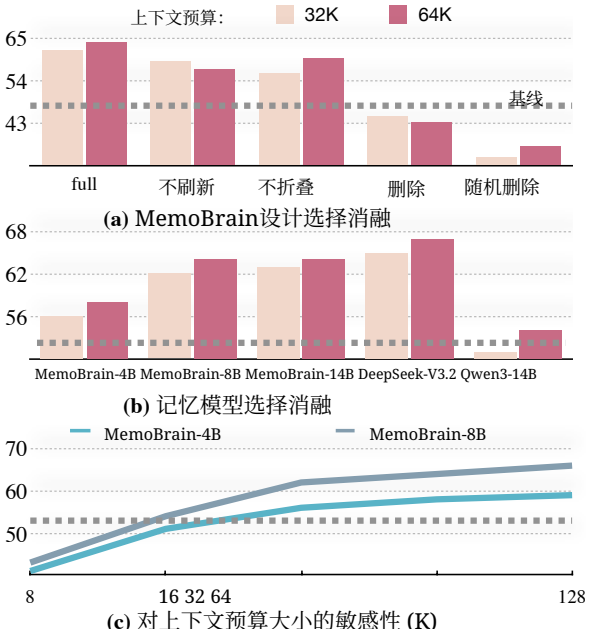


图2：MemoBrain 的消融研究。所有实验均在从 BrowseComp-Plus 中随机抽取的 100 个实例上进行，并使用 DeepResearch-30B-A3B 作为基础推理智能体。

4.3 消融研究

为论证 MemoBrain 的技术有效性，我们从三个维度开展了全面的消融研究：（**a**）执行记忆设计选择、（**b**）记忆模型选择，以及（**c**）对记忆预算大小的敏感性。结果见图2。

执行记忆设计的影响。我们首先对 MemoBrain 的关键设计组件进行消融，以评估它们各自的贡献。移除任何主要组件都会持续导致性能下降，这表明 MemoBrain 的有效性源于各组件的协同设计。具体而言，在保留记忆图结构且仅启用一种操作——轨迹折叠或选择性清除——时，性能均优于不含执行记忆的基线。相比之下，直接删除已清除会话或随机移除推理步骤等策略会显著削弱性能。这些结果表明，MemoBrain 的收益并非仅仅来自缩短上下文长度，而是源于能够保留推理轨迹逻辑骨干的结构化且具语义感知的记忆操作。

记忆模型选择的影响。 我们评估了采用不同规模 Qwen3 骨干模型（4B、8B 和 14B）的 MemoBrain。在所有情况下，与单独使用推理

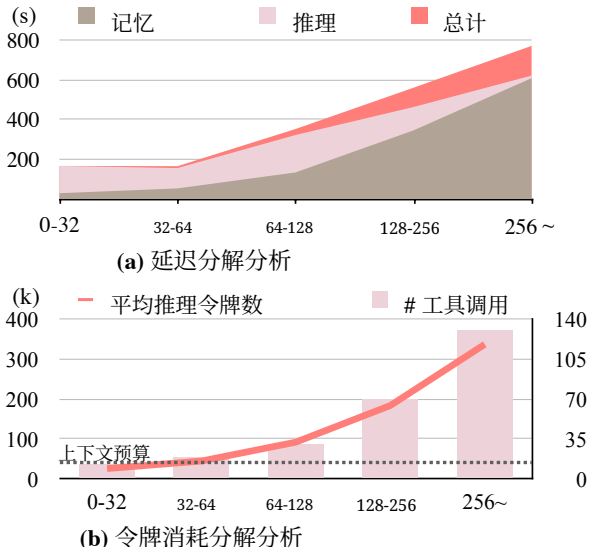


图 3：MemoBrain 的效率分析。在完整 BrowseComp-Plus 基准测试中，采用 32K 记忆预算时的端到端运行时间统计；推理模型使用 4×H100 GPU，记忆模型使用 2×H100 GPU，批量大小为 8。

智能体相比，集成 MemoBrain 都能显著提升性能，这表明执行记忆带来的益处并不局限于特定的模型规模。尽管采用 DeepSeek-V3.2 这类更强的记忆模型能取得最佳整体结果，但使用 8B 或 14B 骨干模型的 MemoBrain 表现也不相上下。综合考虑效率与效果之间的权衡，我们在大多数实验中采用 MemoBrain-8B。

此外，我们观察到，缺乏针对任务的优化的模型（例如 Qwen3-14B）表现出较弱的指令遵循能力，常常生成无效的记忆更新或不恰当的管理操作，从而干扰推理轨迹。这表明，有效的执行记忆更多依赖于有针对性的优化，而不仅仅是模型规模。

对记忆预算大小的敏感性。最后，我们分析了 MemoBrain 对上下文预算的敏感性。在所有设置下，较大的预算通常会带来更好的性能。当预算低于 16K 时，性能可能低于基线，因为保留的上下文不足以支持有意义的推理扩展。然而，超过 64K 后，性能提升开始趋于饱和，这表明记忆预算足以应对大多数任务，而且触发记忆管理的频率可能更低。这一趋势表明，MemoBrain 在合理范围内对预算选择具有稳健性，无需过大的记忆预算也能发挥作用。

4.4 效率分析。

我们评估了 MemoBrain 在有效推理上下文长度方面的效率（图3）。图3 (a) 将延迟分解为推理时间（推理智能体运行时间）、记忆构建时间（异步记忆构建）和总时间；若触发记忆管理，总时间还包括其耗时。由于记忆构建以异步方式进行，只要其耗时低于推理时间，就不会增加端到端延迟。对于最长达 128K 词元的轨迹，与推理时间相比，记忆构建和记忆管理的开销都微不足道。虽然随着记忆图增长，较长轨迹的记忆相关延迟也会增加，但即使超过 256K 词元，记忆构建仍始终不会阻塞推理，这证实了 MemoBrain 在副驾驶式异步执行模式下能够高效扩展。

图3 (b) 展示了词元消耗和工具使用情况。尽管基础智能体受限于 128K 的上下文窗口，集成 MemoBrain 后能够在数十万词元上进行有效推理，并支持显著更多的工具调用，同时仍在更小的固定上下文预算下运行。这一结果表明，MemoBrain 通过提供紧凑的即时上下文支持，在保留关键推理结构的同时，扩展了工具增强型智能体的实际推理范围，而不是依赖无限制地扩展原始上下文。

5 结论

本文研究工具增强型智能体框架中的复杂推理。在这些框架中，长程工具交互会导致推理轨迹和瞬态产物不断累积，给有限的大语言模型上下文带来压力，并损害任务对齐和推理连贯性。为应对这一挑战，我们提出执行记忆，将记忆重新构想为一种针对推理结构的显式控制机制，而非被动的上下文存储。基于这一视角，我们提出*Memo-Brain*，这是一种副驾驶式记忆系统，能够维护全局推理图，并采用预算感知的折叠与刷新机制，以提供紧凑的即时上下文。在多个具有挑战性的基准测试中，实验结果表明该系统持续取得显著的性能提升；消融实验和效率研究进一步验证了所提设计的有效性与可扩展性。更多案例研究见附录A.3。

局限性

尽管 Memo- Brain 在多个基准测试中展现出显著成效，但仍存在若干值得指出的局限性。

（1）MemoBrain 假设底层推理智能体能够持续进行长程工具使用和迭代推理。然而，许多当前的大语言模型往往在进行有限次数的工具调用后便过早终止推理，常常在收集到充分证据之前就得出结论。在这种情况下，累积的上下文尚未达到触发记忆管理的阈值，MemoBrain 也就鲜有机会介入。我们在使用 Qwen3-30B-A3B-2507-Instruct 和 Qwen3-30B-A3B-2507-Thinking 进行的实验中都观察到了这一局限：过早终止限制了执行记忆在实践中的益处。

（2）尽管 MemoBrain 维护的记忆图天然支持更丰富、受人类记忆启发的认知操作，例如重新激活先前已刷新的推理片段，或对记忆图进行分区以开展并行探索，但本研究仅关注两种核心操作：折叠和选择性清除。这些设计旨在单独考察执行记忆控制的基本影响。探索更高级的记忆操作及其协调方式，仍是未来研究的重要方向。

（3）我们的实验对比未涵盖所有可能相关的基线。有些记忆增强方法主要针对长上下文场景设计，无法直接适用于动态推理轨迹。另一些方法缺少公开实现，或所需计算资源超出我们的限制。此外，我们复现的 DeepResearch-30B-A3B 在 GAIA 和 WebWalker 上的结果略低于原论文报告的结果，部分原因是工具实现存在差异（例如搜索质量和代码执行）。尽管如此，所有对比实验均在相同设置下进行，因此 MemoBrain 的相对提升仍然有效且具有意义。

参考文献

Yushi Bai、Shangqing Tu、Jiajie Zhang、Hao Peng、Xiaozhi Wang、Xin Lv、Shulin Cao、Jiazheng Xu、Lei Hou、Yuxiao Dong、Jie Tang 和 Juanzi Li。2025年。Longbench v2：迈向更深入的理解以及在真实长上下文多任务上的推理。预印本，arXiv:2412.15204。

Chi-Min Chan、Chunpu Xu、Ruibin Yuan、Hongyin Luo、Wei Xue、Yike Guo 和 Jie Fu。2024。RQ-RAG：学习优化查询，以实现检索增强生成。载于第一届语言建模会议。

Zijian Chen、Xueguang Ma、Shengyao Zhuang、Ping Nie、Kai Zou、Andrew Liu、Joshua Green、Kshama Patel、Ruoxi Meng、Mingyi Su、Sahel Shari- fymoghaddam、Yanxi Li、Haoran Hong、Xinyu Shi、Xuye Liu、Nandan Thakur、Crystina Zhang、Luyu Gao、Wenhu Chen 和 Jimmy Lin。2025年。Browsecomp-plus：一种更公平、透明的深度研究智能体评测基准。*CoRR*，abs/2508.06600。

Prateek Chhikara、Dev Khant、Saket Aryan、Taranjeet Singh 和 Deshraj Yadav。2025年。Mem0：构建具备可扩展长期记忆、可用于生产环境的人工智能代理。*CoRR*，abs/2504.19413。

DeepSeek-AI。2025年。Deepseek-r1：通过强化学习激发大语言模型的推理能力。*CoRR*，abs/2501.12948。

Darren Edge、Ha Trinh、Newman Cheng、Joshua Bradley、Alex Chao、Apurva Mody、Steven Truitt 和 Jonathan Larson。2024。从局部到全局：一种面向查询的摘要图检索增强生成方法。预印本，arXiv:2404.16130。

Huan-ang Gao、Jiayi Geng、Wenyue Hua、Mengkang Hu、Xinzhe Juan、Hongzhang Liu、Shilong Liu、Ji- ahaa Qiu、Xuan Qi、Yiran Wu、Hongru Wang、Han Xiao、Yuhang Zhou、Shaokun Zhang、Jiayi Zhang、Jinyu Xiang、Yixiong Fang、Qiwen Zhao、Dongrui Liu、Qihan Ren、Cheng Qian、Zhenhailong Wang、Minda Hu、Huazheng Wang、Qingyun Wu、Heng Ji 和 Mengdi Wang。2025年。自进化代理综述：迈向人工超级智能。*CoRR*，abs/2507.21046。

GLM团队、Aohan Zeng、Bin Xu、Bowen Wang、Chen- hui Zhang、Da Yin、Diego Rojas、Guanyu Feng、Han- lin Zhao、Hanyu Lai、Hao Yu、Hongning Wang、Ji- adai Sun、Jiajie Zhang、Jiale Cheng、Jiayi Gui、Jie Tang、Jing Zhang、Juanzi Li、Lei Zhao、Lindong Wu、Lucen Zhong、Mingdao Liu、Minlie Huang、Peng Zhang、Qinkai Zheng、Rui Lu、Shuaiqi Duan、Shu- dan Zhang、Shulin Cao、Shuxun Yang、Weng Lam Tam、Wenyi Zhao、Xiao Liu、Xiao Xia、Xiaohan Zhang、Xiaotao Gu、Xin Lv、Xinghan Liu、Xinyi Liu、Xinyue Yang、Xixuan Song、Xunkai Zhang、Yifan An、Yifan Xu、Yilin Niu、YuantaoYang、Yueyan Li、Yushi Bai、Yuxiao Dong、Zehan Qi、Zhaoyu Wang、Zhen Yang、Zhengxiao Du、Zhenyu Hou 和 Zihan Wang。2024。ChatGLM：从 GLM-130B 到 GLM-4 All Tools 的大语言模型家族。预印本，arXiv:2406.12793。

Cheng-Ping Hsieh、Simeng Sun、Samuel Kriman、Shan- tanu Acharya、Dima Rekesh、Fei Jia 和 Boris Gins- burg。2024 年。Ruler：长上下文语言模型的真实上下文长度是多少？

预印本，arXiv:2404.06654。

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, BoyangLiu,FangyiZhu,JiahangLin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxi- ang Sun, Yutao Zhu, Hao Sun, Boci Peng, Zhen- rong Cheng, Xuanbo Fan, Jiaxin Guo, Xinlei Yu, Zhenhong Zhou, Zewen Hu, Jiahao Huo, Junhao Wang, Yuwei Niu, Yu Wang, Zhenfei Yin, Xiaobin Hu, Yue Liao, Qiankun Li, Kun Wang, Wangchun- shu Zhou, Yixin Liu, Dawei Cheng, Qi Zhang, Tao Gui, Shirui Pan, Yan Zhang, Philip Torr, Zhicheng Dou, Ji-Rong Wen, Xuanjing Huang, Yu-Gang Jiang, and Shuicheng Yan. 2025年人工智能智能体时代的记忆。预印本，arXiv:2512.13564。

JiazhengKang、Mingming Ji、Zhe Zhao 和Ting Bai。2025年。人工智能智能体记忆操作系统。预印本，arXiv:2506.06326。

Baixuan Li、Bo Zhang、Dingchu Zhang、Fei Huang、GuangyuLi、Guoxin Chen、Huifeng Yin、Jialong Wu、Jingren Zhou、Kuan Li、Liangcai Su、Litu Ou、Li- wen Zhang、Pengjun Xie、Rui Ye、Wenbiao Yin、Xin- miao Yu、Xinyu Wang、Xixi Wu、Xuanzhong Chen、Yida Zhao、Zhen Zhang、Zhengwei Tao、Zhongwang Zhang、Zile Qiao、Chenxi Wang、Donglei Yu、Gang Fu、Haiyang Shen、Jiayin Yang、Jun Lin、Junkai Zhang、Kui Zeng、Li Yang、Hailong Yin、Maojia Song、Ming Yan、Peng Xia、Qian Xiao、Rui Min、Ruixue Ding、Runnan Fang、Shaowei Chen、Shen Huang、Shihang Wang、Shihao Cai、Weizhou Shen、Xiaobin Wang、Xin Guan、Xinyu Geng、Yingcheng Shi、Yuning Wu、Zhuo Chen、Zijian Li 和 Yong Jiang。2025a。通义深度研究技术报告。*CoRR*，abs/2510.24701。

Xiaoxi Li、Guanting Dong、Jiajie Jin、Yuyao Zhang、Yujia Zhou、Yutao Zhu、Peitian Zhang 和 Zhicheng Dou。2025b。Search-o1：智能体搜索增强型大型推理模型。*CoRR*，abs/2501.05366。

Xiaoxi Li、Wenxiang Jiao、Jiarui Jin、Guanting Dong、Ji ajie Jin、Yinuo Wang、Hao Wang、Yutao Zhu、Ji- Rong Wen、Yuan Lu 和 Zhicheng Dou。2025c。Deepagent：具备可扩展工具集的通用推理智能体。*CoRR*，abs/2510.21618。

Xiaoxi Li、Jiajie Jin、Guanting Dong、Hongjin Qian、Yu- tao Zhu、Yongkang Wu、Ji-Rong Wen 和 Zhicheng Dou。2025d。Webthinker：赋能大型推理模型，使其具备深度研究能力。*arXiv* 预印本 *arXiv:2504.21776*。

Zhiyu Li, Shichao Song, HanyuWang, Simin Niu,Ding Chen, JiaweiYang, Chenyang Xi, Huayi Lai,Jihao Zhao, Yezhaohui Wang, JunpengRen, Ze-haoLin, Jiahao Huo, TianyiChen, Kai Chen, Ke-hangLi, Zhiqiang Yin, QingchenYu, Bo Tang,HongkangYang, Zhi-Qin John Xu,and FeiyuXiong.2025e. Memos：面向大语言模型中记忆增强生成（MAG）的操作系统. 预印本，arXiv:2505.22101。

AdyashaMaharana, Dong-Ho Lee, SergeyTulyakov,Mohit Bansal,Francesco Barbieri, andYuwei Fang.2024. 评估大语言模型智能体的超长期对话记忆. 预印本，arXiv:2402.17753。Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Bao- long Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, et al. 2025. 大语言模型上下文工程综述。 *arXiv* 预印本 *arXiv:2507.13334*。Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. GAIA：通用人工智能助手基准测试。载于第十二届国际学习表征会议。

OpenAI。2024年。 Gpt-4 技术报告。 预印本，arXiv:2303.08774。

Charles Packer、Vivian Fang、Shishir G Patil、Kevin Lin、Sarah Wooders 和 Joseph E Gonzalez。2023年。Memgpt：朝向作为操作系统的大语言模型。*arXiv* 预印本 *arXiv:2310.08560*。

Cheng Qian、Emre Can Acikgoz、Qi He、Hongru Wang、Xiusi Chen、Dilek Hakkani-Tür、Gokhan Tur 和 Heng Ji。2025a。 Toolrl：奖励是工具学习所需的一切。 预印本，arXiv:2504.13958。

Hongjin Qian、Zheng Liu、Peitian Zhang、Kelong Mao、Defu Lian、Zhicheng Dou 和 Tiejun Huang。2025b。Memorag：通过全局记忆增强检索增强提升长上下文处理。在*ACM Web Conference 2025* 论文集，WWW '25，第 2366–2377 页，美国纽约州纽约市。计算机协会。

Rafael Rafailov、Archit Sharma、Eric Mitchell、Christo- pher D Manning、Stefano Ermon 和 Chelsea Finn。2024 年。直接偏好优化：你的语言模型其实是一个奖励模型。神经信息处理系统进展，36。

Zhihong Shao、Yeyun Gong、Yelong Shen、Minlie Huang、Nan Duan 和 Weizhu Chen。2023 年。通过迭代检索生成协同增强检索增强型大语言模型。载于计算语言学协会论文集：*EMNLP 2023*，第 9248–9274 页。

Weiwei Sun、Miao Lu、Zhan Ling、Kang Liu、Xuesong Yao、Yiming Yang 和 Jiecao Chen。2025年。 通过上下文折叠扩展长时程大语言模型智能体。 预印本，arXiv:2510.11967。

Lei Wang、Chen Ma、Xueyang Feng、Zeyu Zhang、Hao Yang、Jingsen Zhang、Zhiyuan Chen、Jiakai Tang、Xu Chen、Yankai Lin、Wayne Xin Zhao、ZheweiWei 和 Jirong Wen。2024 年。 基于大语言模型的自主智能体综述。 计算机科学前沿，18(6):186345。

Weizhi Wang、Li Dong、Hao Cheng、Xiaodong Liu、Xifeng Yan、Jianfeng Gao 和 Furu Wei。2023 年。通过长期记忆增强语言模型。

载于《神经信息处理系统进展》第 *36* 卷：*2023*年神经信息处理系统年会，*NeurIPS 2023*，美国路易斯安那州新奥尔良，*2023* 年 *12* 月 *10* 日至 *16* 日。

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus 和 Amelia Glaese。2025年。Browsecomp：一项简单却颇具挑战性的浏览智能体基准测试。*arXiv* 预印本 *arXiv:2504.12516*。

Jialong Wu, Baixuan Li, RunnanFang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Yong Jiang, Pengjun Xie, Fei Huang 和 Jin- gren Zhou。2025a。Webdancer：迈向自主信息寻求主体。*CoRR*，abs/2505.22648。

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie 等。2025b。Webwalker：网页遍历中的大语言模型基准测试。*arXiv* 预印本 *arXiv:2501.07572*。

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Minhao Cheng, Shuai Wang, Hong Cheng, 和 Jingren Zhou。2025c。Re-sum：通过上下文摘要解锁长时程搜索智能。*CoRR*，abs/2509.13313。

Ziyi Xia, KunLuo, Hongjin Qian, 和 Zheng Liu。2025年。面向深度研究的开放数据合成。预印本，arXiv:2509.00375。

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, 和 Yongfeng Zhang。2025年。A-mem：面向大语言模型智能体的智能体记忆。预印本，arXiv:2502.12110。

B. Y. Yan、Chaofan Li、Hongjin Qian、Shuqi Lu 和 Zheng Liu。2025a。通过深度研究实现通用智能体记忆。预印本，arXiv:2511.18423。

Sikuan Yan、Xiufeng Yang、Zuchao Huang、Ercong Nie、Zifeng Ding、Zonggen Li、Xiaowen Ma、Kristian Kersting、Jeff Z. Pan、Hinrich Schütze、Volker Tresp 和 Yunpu Ma。2025b。Memory-r1：通过强化学习增强大语言模型智能体对记忆的管理与利用。预印本，arXiv:2508.19828。

An Yang、Anfeng Li、Baosong Yang、Beichen Zhang、Binyuan Hui、Bo Zheng、Bowen Yu、Chang Gao、Chengen Huang、Chenxu Lv、Chujie Zheng、Day- iheng Liu、Fan Zhou、Fei Huang、Feng Hu、Hao Ge、Haoran Wei、Huan Lin、Jialong Tang、Jian Yang、Jianhong Tu、Jianwei Zhang、Jian Yang、Ji- axi Yang、Jingren Zhou、Junyang Lin、Kai Dang、Keqin Bao、Kexin Yang、Le Yu、Lianghao Deng、Mei Li、Mingfeng Xue、Mingze Li、Pei Zhang、Peng Wang、Qin Zhu、Rui Men、Ruize Gao、Shixuan Liu、Shuang Luo、Tianhao Li、TianyiTang、Wenbiao Yin、Xingzhang Ren、Xinyu Wang、Xinyu Zhang、Xu- ancheng Ren、Yang Fan、Yang Su、Yichang Zhang、Yinger Zhang、Yu Wan、Yuqiong Liu、Zekun Wang、Zeyu Cui、Zhenru Zhang、Zhipeng Zhou 和 Zi- han Qiu。2025年。Qwen3技术报告。*CoRR*，abs/2505.09388。

Shunyu Yao、Jeffrey Zhao、Dian Yu、Nan Du、Izhak Shafran、Karthik Narasimhan 和 Yuan Cao。2023。ReAct：在语言模型中协同推理与行动。发表于国际学习表征会议（*ICLR*）。

Rui Ye、Zhongwang Zhang、Kuan Li、Huifeng Yin、Zhengwei Tao、Yida Zhao、Liangcai Su、Liwen Zhang、Zile Qiao、Xinyu Wang、Pengjun Xie、Fei Huang、Siheng Chen、Jingren Zhou 和 Yong Jiang。2025年。AgentFold：采用主动上下文管理的长时程网络代理。*CoRR*，abs/2510.24699。

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang,Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. 2025年 Memagent：通过多轮对话强化学习记忆代理重塑长上下文大语言模型。预印本，arXiv:2507.02259。

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Yue Liao, Hongru Wang, Mengyue Yang, Heng Ji, Michael Littman, Jun Wang, Shuicheng Yan, Philip Torr, and Lei Bai. 2025a 大语言模型智能体强化学习的发展现状：综述。*CoRR*，abs/2509.02547。

Zeyu Zhang、Quanyu Dai、Xiaohe Bo、Chen Ma、RuiLi、Xu Chen、Jieming Zhu、Zhenhua Dong 和 Ji-Rong Wen。2025b。基于大语言模型的智能体的记忆机制综述。*ACM Trans.Inf.Syst.*，43(6):155:1–155:47。

Wayne Xin Zhao、Kun Zhou、Junyi Li、Tianyi Tang、Xiaolei Wang、Yupeng Hou、Yingqian Min、Beichen Zhang、Junjie Zhang、Zican Dong、Yifan Du、Chen Yang、Yushuo Chen、Zhipeng Chen、Jinhao Jiang、Ruiyang Ren、Yifan Li、Xinyu Tang、Zikang Liu、Peiyu Liu、Jian-Yun Nie 和 Ji-Rong Wen。2024。大型语言模型综述。预印本，arXiv:2303.18223。

Wanjun Zhong、Lianghong Guo、Qiqi Gao、He Ye 和 Yanlin Wang。2024 年。Memorybank：通过长期记忆增强大语言模型。载于第三十八届 *AAAI* 人工智能会议（*AAAI 2024*）、第三十六届人工智能创新应用会议（*IAAI 2024*）、第十四届人工智能教育进展研讨会（*EAAI 2014*），*2024* 年 *2* 月 *20* 日至 *27* 日，加拿大温哥华，第 19724–19731 页。AAAI Press。

你是一名记忆代理，正在协作完成一项复杂的多步骤研究任务。你的工作是根据需要创建新的节点和边，将最新的推理步骤追加到当前推理图中。

推理图由以下部分组成：
- 节点：
  - “任务”：用户提出的原始问题或主要任务。- “子任务”：推理过程中分解出的子问题。- “证据”：推理过程中为解决某个（子）任务而得出的事实性答案或结论。- “摘要”：综合若干先前节点的摘要节点。（注意：此步骤中不会创建新的摘要节点。）

新的摘要节点。）
- 边：
  - 将“任务”节点连接到新创建的“子任务”节点（分解任务）。- 将“任务”或“子任务”节点连接到更具体或重新表述的“子任务”。- 将“证据”节点连接到其支持或回答的任务或子任务。

输入（你将获得）：1. 当前推理图的状态（以人类可读形式或 JSON 形式呈现），包括所有现有节点及其类型和边。2. 一个新的推理步骤（子任务、中间结果或证据），必须追加到图中。你的目标：1. 判断应根据当前推理步骤创建哪些新节点（子任务或证据）。- 明确标出节点的"kind"："subtask"或"evidence"。- 记录要点，以列表形式概括推理步骤的核心信息，例如["role": "", "content": "", ...],- 此步骤中不要创建"summary"类型的节点。

2. 判断需要哪些新边来连接新节点与现有节点或其他新节点：- 对于每条边，指定"src"和"dst"（src 是节点 ID，dst 是临时_ID），并可选择为这条边提供简短的"rationale"。

3. 将更新内容作为单个 JSON 对象输出，包含以下字段：- "add_edges"：新边对象列表，每个对象至少包含 "src"、"dst"，并可选包含 "rationale"。规则：- 输出必须是严格有效的 JSON；不得包含任何评论或解释。- 可酌情使用现有节点 ID 作为边的 src。- 避免在图中创建循环。- 只为推理过程中的有意义新步骤创建新节点。

表3：MemoBrain 中用于执行记忆构建的系统提示。

A 附录

A.1 实现细节

MemoBrain 基于 Qwen3 指令微调模型系列初始化（Yang 等人，2025年），包括 Qwen3-4B-Instruct-2507、Qwen3-8B和 Qwen3-14B。训练数据由 InfoSeek 基准（Xia 等人，2025年）合成，该基准包含超过 50K 个复杂推理问答对；合成过程中，我们使用 DeepSeek-V3.2 作为记忆预言机，生成执行层面的监督信号，最终得到 37,719 个记忆构建样本和 3,016 个记忆管理样本。数据合成产生了 389 美元的 API 成本。所有模型均使用LlamaFactory 进行训练，学习率为 2×10^{-5}，梯度累积步数为 8，每设备批量大小为 1，最大梯度范数为 1.0，使用 AdamW 优化器，训练 3 轮，最大序列长度为 32K 词元。记忆构建通过监督微调进行优化，而记忆管理则通过基于偏好的学习进行优化。4B、8B 和 14B 模型的总训练时间分别为 17、21 和 27 小时。

对于 GAIA 和 WebWalker，我们使用 Google 自定义搜索 API 检索排名前 10 的结果，并使用Jina.AI的网页阅读器 API 获取网页内容，再使用Qwen3-30B-A3B-Instruct-2507对其进行提炼。代码执行通过 Python 内置的`exec()` 函数完成。对于 BrowseComp-Plus，则使用 Qwen3-Embed-8B 进行稠密检索。

在主要实验中，MemoBrain 针对 GAIA 和 Web-Walker 的记忆预算设为 32K 词元，针对 BrowseComp-Plus 的记忆预算设为 64K 词元；当累积上下文达到预算时，便触发记忆管理，而基线方法则保留其原有的上下文限制（例如 DeepResearch-30B-A3B 的 128K）。我们使用 Pass@1 作为评估指标，并使用 GPT-4o-2024-08-06 作为基于大语言模型的评审器。所有实验均在 8 张 NVIDIA H100 80G GPU 上进行。

A.2 提示

在表 3 和表4中，我们分别展示 MemoBrain 使用的记忆化提示词和记忆管理提示词。本研究使用的其他提示词，包括用于复杂推理的系统提示词和评估提示词，均已提供在我们的代码中，可访问此代码库。

A.3 案例研究

表 5展示了一个由六个推理过程构成的代表性记忆图，其根节点对应原始输入任务。这个案例展示了 MemoBrain 如何将一段漫长且错综复杂的推理轨迹重新组织为一组语义连贯的片段，同时在记忆图中明确编码它们之间的依赖关系。通过执行记忆构建与管理，MemoBrain 对不断演进的问题求解过程进行抽象，包括子任务分解、证据验证和假设排除，而非保留原始执行轨迹。由此生成的记忆图为推理过程提供了一种认知上可解释的表征，只保留高显著性结论和关键逻辑依赖。值得注意的是，原本长达数万个词元的轨迹，在记忆图中仅用数百个词元便可紧凑地表示，凸显了 MemoBrain 在有界上下文预算下维持推理连贯性并与任务保持一致的能力。

在表6中，我们进一步展示了 MemoBrain 中一项具体的记忆构建操作示例。每个由内部推演、工具调用和工具响应组成的推理过程，在整合到全局记忆图之前，都会被抽象为一个带有临时标识符的思维单元。在此过程中，MemoBrain 会提取该过程的语义结果，分配恰当的思维类型和激活状态，并与前驱思维建立明确的依赖关系。这一构建流程将瞬态轨迹转化为稳定且易于解释的记忆单元，为后续的记忆管理和上下文调节奠定结构基础。如案例所示，记忆构建使 MemoBrain 能够逐步构建推理轨迹的依赖感知表示，同时舍弃对后续推理不必要的底层细节。

在表7中，我们展示了一个将记忆管理 操作应用于已构建记忆图的具体示例。随着推理轨迹不断扩展，工作上下文接近预设预算时，MemoBrain 会评估全局记忆结构，并执行管理操作以调节信息暴露。具体而言，当已完成子轨迹所关联的子任务得出确定结果后，这些子轨迹会被折叠为精简的摘要思维；而效用较低或已被取代的思维则会被选择性刷新，以减少上下文冗余。重要的是，这些操作会保留剩余思维之间的依赖结构，确保全局推理过程连贯，并与任务目标保持一致。通过折叠和刷新执行层轨迹，MemoBrain 将数百个词元浓缩为一小组高显著性记忆单元，从而在有界上下文预算下持续进行长程推理，同时不牺牲关键的中间结论。

以下是一个复杂任务的记忆图，详细列出了关键步骤及其依赖关系。该图以缩进结构呈现，例如：节点 1 [节点类型] [状态] [节点信息]-[->] 节点 2（一条边） [边的理由（解释通往节点 2 的推理过程）]每个节点都包含其当前状态（活动或非活动）。你的任务：1. 根据当前推理状态，找出哪些节点是冗余的，可以刷新。冗余节点包括：无效推理步骤（例如，没有产生有意义的反馈的步骤），或已被后续步骤取代并完善的节点。此操作称为“刷新”。2. 找出图中哪些路径可以折叠。如果某个子任务经过多个推理步骤后得出了确定性推理结果（例如，找到目标信息，或证据节点表明明确无法获得证据），则可以合并此前相关的推理步骤。此操作称为“折叠”。重要约束：- 不要在刷新_操作和折叠_操作中同时包含同一个节点。每个节点只能作为一种操作的目标，因为刷新和折叠操作都会使节点变为非活动状态。示例指南：- 仅对状态为“活动”的节点执行刷新和折叠操作；对于其他状态的节点，忽略这些操作。- 不要将同一个节点分配给多个刷新_操作或折叠_操作。确保每个活动节点最多出现在一种操作中。- 对于“刷新_操作”，请简要说明每个节点在当前推理上下文中冗余或已被取代的原因。- 对于“折叠_操作”，请将可以折叠的连续相关节点 ID 分组，并提供摘要备注（以包含 role 和 content 的字典列表形式）。- 不要在 JSON 之外添加任何解释或评论。- 仅输出 JSON 对象，不要添加其他格式或文本。

表4：MemoBrain 中用于执行记忆管理的系统提示。

- 节点 1: [任务] [活动] 开始解决任务：你能提供符合以下条件的个人姓名吗？- 截至2023年12月，该人士是一个成立于2009年的研究小组的协调员。- 共同编辑了一本由劳特利奇于2018年出版的书籍。- 与其共同编辑该书的人曾于2019年在一场会议上担任主旨演讲嘉宾。- 2020年之前曾担任专题小组的召集人。- 于2012年发表了一篇文章。- 博士学位论文研究的是一位英国作家的作品。 [输入任务]-[->] 节点 2 [理由：将主要任务拆解为一个子任务，根据初始条件搜索相关信息。] 节点 2: [子任务] [活动] ['角色': '助手', '内容': "搜索2018年共同编辑劳特利奇书籍的个人，重点关注博士学位论文研究英国作家作品且符合其他条件的人士。", '角色': '用户', '内容': '返回关于2018年劳特利奇共同编辑书籍、成立于2009年的研究小组协调员（2023年）以及2019年专题小组召集人和主旨演讲嘉宾的查询搜索结果。']-[->] 节点 3 [理由：搜索结果中的证据为该子任务提供支持，带来可供进一步分析的潜在线索或数据。] 节点 3: [证据] [活动] ['角色': '助手', '内容': '分析搜索结果，找出可能符合条件的候选人。', '角色': '用户', '内容': '发现了多个结果，包括 Mathew Abbott（《迈克尔·弗里德与哲学：现代主义、意向与戏剧性》的编辑之一，该书由劳特利奇于2018年出版），但仍需核实其他信息，例如研究小组协调员身份、专题小组召集人身份、2012年发表的文章以及研究英国作家的博士学位论文。']-[->] 节点 4 [理由：将子任务进一步拆解为更精确的搜索查询，以找到符合所有条件的个人。] 节点 4: [子任务] [活动] ['角色': '助手', '内容': '优化搜索查询，聚焦于具体条件：2018年共同编辑劳特利奇出版的书籍、成立于2009年的研究小组协调员身份，以及其他信息，如2020年前担任专题小组召集人、2012年发表文章、博士学位论文研究英国作家。', '角色': '用户', '内容': "搜索 'co-edited Routledge book 2018'、'research group founded 2009 coordinator'、'convenor panel 2019 keynote speaker'、'article published 2012'、'PhD thesis on English writer'。"]-[->] 节点 5 [理由：搜索结果中的证据为该子任务提供支持，提供可供分析、以找出潜在候选人的数据。] 节点 5: [证据] [活动] ['角色': '助手', '内容': '优化后的查询结果显示了潜在匹配者，例如 Mona Baker（成立于2009年的知识谱系研究网络的共同协调员）和 Pooyan Ta mimi Arab（2023年共同编辑了《劳特利奇物质宗教手册》，而非2018年出版）。仍需核实其他条件，例如2020年前担任专题小组召集人、2012年发表文章以及研究英国作家的博士学位论文。', '角色': '用户', '内容': '结果中包含各种教师简介，但没有明确符合所有条件的人士；还需要进一步分析。']-[->] 节点 6 [理由：将子任务拆解为更具体的搜索查询，以找到2018年共同编辑的劳特利奇书籍，且其中一位主旨演讲嘉宾于2019年发表演讲。] 节点 6: [子任务] [活动] ['角色': '助手', '内容': "搜索 'Routledge 2018 co-edited Keynote 2019'，查找劳特利奇于2018年出版的共同编辑书籍，且其中一位共同编辑曾于2019年在会议上发表主题演讲。", '角色': '用户', '内容': '返回关于2018年劳特利奇共同编辑书籍及2019年主旨演讲嘉宾的查询搜索结果。']-[->] 节点 7 [理由：搜索结果中的证据为该子任务提供支持，提供可供分析、以找出潜在候选人的数据。] 节点 7: [证据] [活动] ['角色': '助手', '内容': '分析该查询的搜索结果，找出可能符合条件的人士，例如2018年共同编辑劳特利奇书籍且于2019年担任主旨演讲嘉宾的人。', '角色': '用户', '内容': '搜索结果显示了各种教师简介和网页，但没有直接匹配项；需要进一步优化搜索，或查询其他来源。']I记忆图：482个词元；原始推理轨迹：17,268个词元。]

表 5：记忆图示例。节点对应 MemoBrain 生成的记忆思维单元，每个节点均标注其思维类型、激活状态和压缩表示。有向边 (-[->]) 表示依赖关系，刻画推理步骤如何层层递进。

```
{
  "add_节点": [
    {
      "tmp_id": "tmp6",
      "类型": "子任务",
      "思维": [
        {
          "角色": "助手",
          "内容": "搜索'劳特利奇 2018年 共同编辑 主题演讲 2019年'，查找出版于
    2018年、由劳特利奇出版且由多人共同编辑的书籍，其中一位共同编辑曾在2019年的一次会议上发表主题演讲
    。"
        },
        {
          "角色": "用户",
          "内容": "返回有关劳特利奇 2018年共同编辑的书籍以及
    2019年主旨演讲嘉宾的查询搜索结果。"
        }
      ]
    },
    {
      "tmp_id": "tmp7",
      "kind": "evidence",
      "thought": [
        {
          "role": "assistant",
          "content": "分析查询的搜索结果，以识别与以下内容可能匹配的结果
    条件，例如2018年的共同编辑劳特利奇书籍，以及2019年的主旨演讲嘉宾。"
        },
        {
          "role": "user",
          "content": "搜索结果显示了各种教师简介和页面，但没有直接匹配项;
    需要进一步筛选或寻找替代来源。"
        }
      ]
    }
  ],
  "添加_边": [
    {
      "src": 4,
      "dst": "tmp6",
      "理由": "将原始任务分解为一项聚焦搜索，查找由劳特利奇共同编辑的书籍
    出版于2018年，且在2019年有一位主旨演讲嘉宾。"
    },
    {
      "src": "tmp6",
      "dst": "tmp7",
      "理由": "检索到的搜索结果提供了证据，可用于评估子任务
    标准是否得到满足。"
    }
  ]
}
```

表6：记忆构建示意图。新生成的思维单元首先会被分配临时ID，之后在插入记忆图时重新编排索引。有向边表示思维单元之间的依赖关系。

```
{
  "刷新_操作": [
    {
      "id": 2,
      "理由": "此节点已被节点 20 取代，节点 20 提供了确凿证据，证明 Laura
     Lojo-Rodriguez 是话语与身份（D&I）研究组的协调员，这与
    第一个线索相符。节点 2 的子任务已成功解决。"
    },
    {
      "id": 3,
      "理由": "此节点已被节点 30 及后续节点（31-52）取代，后者提供了
    关于合编书籍线索的全面摘要和详细证据。节点 3 的子任务
    已通过更具体的搜索进一步细化并得到解决。"
    },
    {
      "id": 47,
      "理由": "这个节点是多余的，因为其用于核实 2012 年文章发表的子任务
     已由证据节点（48-50）处理，这些节点一致表明没有关于一篇
    2012 年文章的直接证据，因此无需进一步探索。"
    }
  ],
  "fold_ops": [
    {
      "ids": [48, 49, 50],
      "理由": "这些节点为 2012 年文章发表提供了确凿证据
    子任务。 所有搜索结果均一致表明，2012年没有已确认的发表记录。",
      "notes": [
        {
          "角色": "助手",
          "内容": "研究目标是确定 Laura Lojo-Rodriguez 是否发表过一篇
    学术论文。研究涉及有针对性的姓名-年份搜索查询，以及对其公开出版记录进行
    系统性审查。"
        },
        {
          "角色": "用户",
          "内容": "未发现 2012年出版物的证据。虽然有其他年份的出版物记录，
     已记录的年份，根据一致的否定证据，可以明确排除符合 2012年这一条件的可能性
     所有来源中的发现。"
        }
      ]
    }
  ]
}
```


表 7：记忆管理示例。MemoBrain 通过刷新低效用思维单元，并将已完成的子轨迹折叠为精简的摘要节点，对记忆图执行操作。