InfiAgent：面向通用自主智能体的无限时程框架

Chenglin Yu[1],**Yuchen Wang**[2],**Songmiao Wang**[2],**Hongxia Yang**[2],和 **Ming Li**[*2]

[1]香港大学[2]香港理工大学

arXiv:2601.03204v1 [cs.AI] 6 Jan 2026

摘要

大语言模型智能体能够进行推理并使用工具，但在长时程任务中常因上下文无限增长和错误累积而陷入困境。上下文压缩或检索增强提示等常见解决方案，会在信息保真度与推理稳定性之间引入权衡。我们提出了**Infi Agent**，这是一种通用框架，通过将持久化状态外化到 *file-centric state abstraction*中，使智能体的推理上下文始终保持严格有界，不受任务时长影响。在每个步骤中，智能体都会根据工作区状态快照和固定窗口内的近期操作重建上下文。在 DeepResearch 和一项涉及 80 篇论文的文献综述任务上的实验表明，无需特定任务微调，采用 20B 开源模型的 InfiAgent 便能与更大型的专有系统相媲美，并且其长时程覆盖率显著高于以上下文为中心的基线。这些结果表明，显式状态外化可作为构建稳定长时程智能体的实用基础。

仓库：https://github.com/ChenglinPoly/infiAgent

关键词：大语言模型智能体、无限时域推理、以文件为中心的状态管理、自主研究、多智能体协作

1 引言

大语言模型（LLM）日益被部署为自主智能体，能够在各种任务中进行规划、使用工具和多步推理 [1, 2]。近期基于智能体的系统在科学研究、软件工程和信息综合等领域展现出令人鼓舞的性能，这表明大语言模型可以充当通用问题求解器，协调外部工具与资源。尽管取得了这些进展，现有大语言模型智能体在长任务时域中运行时仍然很脆弱。

一个核心挑战在于智能体状态的表示与维护方式。当前大多数智能体框架都默认将大语言模型提示视为状态的主要载体，直接在上下文窗口中累积对话历史、工具轨迹、中间计划和部分结果。随着任务时长增加，这种设计会导致上下文无限增长，迫使系统依赖截断、摘要或启发式检索来满足模型限制。这些机制会引发众所周知的故障模式，包括信息丢失、无关词元造成的干扰，以及对早期错误日益敏感，最终导致系统在长时域内表现不稳定。

检索增强生成（RAG）和长上下文模型通过支持访问外部记忆或使用更大的上下文窗口，部分缓解了上下文长度限制。然而，这些方法仍将长期任务状态与智能体即时的推理上下文交织在一起，随着执行推进，不断增加大语言模型的认知负荷。实证证据表明，随着上下文逐渐填满，智能体的性能往往会下降；在自主研究智能体中，这种现象有时被称为“状态幻觉” [3]。因此，单纯延长上下文长度并不能从根本上解决长时程稳定性问题。

近期研究（如 MAKER [4]）表明，通过极端任务分解，可以在高度结构化领域实现近乎无限执行。此类方法在具有预定义子任务边界的基于逻辑的问题中效果显著，但依赖专用的微型智能体和僵化工作流，因此难以应用于开放式

*通讯作者：ming.li@polyu.edu.hk

领域，例如科学研究；在这些领域中，任务结构、中间目标和相关信息来源都无法预先确定。

本文认为，要在大语言模型智能体中实现稳定的长时程行为，就必须明确区分持久任务状态与有界推理上下文。我们提出了**InfiAgent**，这是一种通用智能体框架，将长期状态外化为以文件为中心的表示。InfiAgent并不将历史信息隐式存储在提示中，而是将文件系统视为智能体操作、环境和中间产物的权威且持久的记录。在每个决策步骤中，智能体仅根据这一外化状态的快照以及固定大小的近期操作窗口重建其推理上下文，从而确保无论任务时长如何，上下文大小都严格保持有界。

基于这种状态抽象，InfiAgent采用分层智能体架构，强制执行结构化的任务分解和受控的工具调用。高层规划智能体处理抽象目标和状态摘要，而低层智能体执行特定领域的操作或原子动作。这种设计减少了扁平多智能体系统中常见的错误传播，并使智能体在延长的执行时程中保持一致的行为。此外，InfiAgent还引入了外部注意力机制，在主要推理上下文之外处理大型文档和大量信息，仅将与任务相关的输出注入智能体状态。

我们在 DeepResearch 基准 [3] 和一项涉及多达 80 篇学术论文的长周期文献综述任务上评估 InfiAgent。无需进行特定任务微调，搭载 20B 参数开源模型的 InfiAgent 在 DeepResearch 基准上的表现可与更大型的专有智能体相媲美，甚至更胜一筹。在涵盖数百个执行步骤的扩展评估中，所提出的以文件为中心的状态抽象能够支持可靠地完成任务，而基线智能体则常因上下文限制而性能下降。这些结果表明，显式状态外化为构建稳定、通用且能够在任意长时间跨度内运行的大语言模型智能体，提供了切实有效的基础。

2 相关工作

2.1 多智能体系统架构

传统多智能体系统主要关注点对点协作模型。尽管这类方法在简单协调任务中表现有效，但难以应对复杂的层级式任务分解。近期研究探索了层级组织方式 [5]。Moore [6] 对分层多智能体系统（HMAS）进行了分类，并指出全局效率与局部自主性之间存在权衡。综合性综述指出，设计能够无缝整合认知、规划与交互的统一智能体，仍是一项长期挑战 [7]。InfiAgent 通过严格的组合约束和以文件为中心的状态来确保稳定性，从而应对这些问题。

2.2 多智能体系统中的任务分解

任务分解一直是核心挑战。现有方法包括目标导向型 [8], 约束型 [9],和基于学习的分解 [10]。AGENTiGraph 等框架利用知识图谱（知识图谱）进行分解 [11]。然而，这些方法通常无法保证系统在长时程中的稳定性。MAKER [4] 针对逻辑任务引入了极端分解，但其在开放式研究中的适用性仍然有限。InfiAgent 采用递归式基于有向无环图的分解，既足够灵活，能够处理一般任务，又能维持严格的父子级控制。

2.3 长上下文与自主研究代理

Shojaee 等人 [3], 指出，深度研究代理中存在“状态幻觉”，并表明随着上下文逐渐填满，性能往往会下降。AgentGym [12]和 Richelieu [13] 等智能体专注于自我进化，而 AI Scientist 等系统则尝试实现端到端自动化。然而，许多方法依赖扩展的上下文窗口或手动模板。InfiAgent 通过基于文件的状态实现“零上下文压缩”，为这些上下文密集型方法提供了一种稳健的替代方案。

3 将以文件为中心的状态形式化，用于长时程智能体

我们通过区分持久任务状态 和有界推理上下文，对长时程智能体执行问题进行形式化。这一区分阐明了以上下文为中心的智能体设计的局限，并说明了 InfiAgent 所采用的以文件为中心的状态抽象为何必要。

3.1 智能体执行作为状态条件决策过程

我们考虑一个在离散时间步 $t = 1$、2、.....上运行的自主大语言模型智能体。在每个时间步，智能体根据任务状态的内部表征选择一个动作 $a_t \in \mathcal{A}$ 。在传统智能体框架中，这一状态由累积的提示上下文隐式表示:

$$c_t = \langle o_1, a_1, \ldots, o_{t-1}, a_{t-1}, o_t \rangle, \tag{1}$$

其中， o_t 表示观测，例如用户指令、工具输出或中间结果。随着 t 增加，上下文长度 $|c_t|$会无限增长，因此需要进行截断或压缩，以适应有限的上下文窗口。

我们将这种设计定义为以上下文为中心的状态表示，其中长期任务信息与短期推理信号交织在同一序列中。这种交织会导致信息保留与推理稳定性之间存在固有的权衡，尤其是在长时程任务中。

3.2 持久状态外化

为将长期记忆与有界推理上下文解耦，我们引入一种显式的持久化状态表示， S_t 并使其随时间演变:

$$S_t = \mathcal{F}_t, \tag{2}$$

其中， $\mathcal{F}_t$ 表示在步骤 t存储于智能体工作区中的文件和结构化产物集合。这些产物可能包括中间结果、计划、摘要、数据集或生成的代码，并共同构成任务进展的权威记录。

T持久状态通过智能体动作所诱导的状态转移算子不断演化 s:

$$\mathcal{F}_{t+1} = \mathcal{T}(\mathcal{F}_t, a_t), \tag{3}$$

其中， $\mathcal{T}$表示文件创建、修改或删除。重要的是， $\mathcal{F}_t$ 不受上下文窗口限制，并且可以随着任务复杂度和持续时间的增加而增长。

3.3 有界推理上下文重构

在每个步骤 t，智能体构造有界推理上下文 c_t^{bounded}，方法是查询持久状态:

$$c_t^{\text{bounded}} = g(\mathcal{F}_t, a_{t-k:t-1}), \tag{4}$$

其中， $a_{t-k:t-1}$ 表示由最近的 k 个动作组成的固定大小窗口，而 $g(\cdot)$是一个确定性的上下文构造函数。在 InfiAgent 中， k 是一个较小的常数（例如 $k = 10$），从而确保相对于任务时域而言， $|c_t^{\text{bounded}}| = \mathcal{O}(1)$。

这种表述保证智能体的推理上下文始终严格有界 t，同时通过持久化状态 $\mathcal{F}_t$隐式保留完整的任务历史。与基于摘要的方法不同，权威状态中的信息不会被丢弃；相反，每一步都会通过状态检查动态确定信息的相关性。

3.4 比较与启示

与上下文中心型智能体（将所有任务状态直接编码在提示中）和检索增强型智能体（将部分记忆外置，但会把检索到的文本重新注入上下文）相比，所提出的文件中心型抽象将持久化状态视为一等对象。通过有界接口重构推理上下文，InfiAgent 消除了上下文无限增长，并减少了长期产物与短期决策之间的相互干扰。

因此，推理错误不太可能随时间推移而隐式累积，且智能体可以扩展执行时域，而不会增加底层语言模型的认知负荷。

4 方法论：InfiAgent 框架

在第 3 节文件中心型状态表述的基础上，我们将 InfiAgent 描述为一个具体系统，用以实现持久化状态外置和有界上下文重构。本节聚焦于架构与实现方案，而非重新定义状态抽象。图1概述了整体情况。

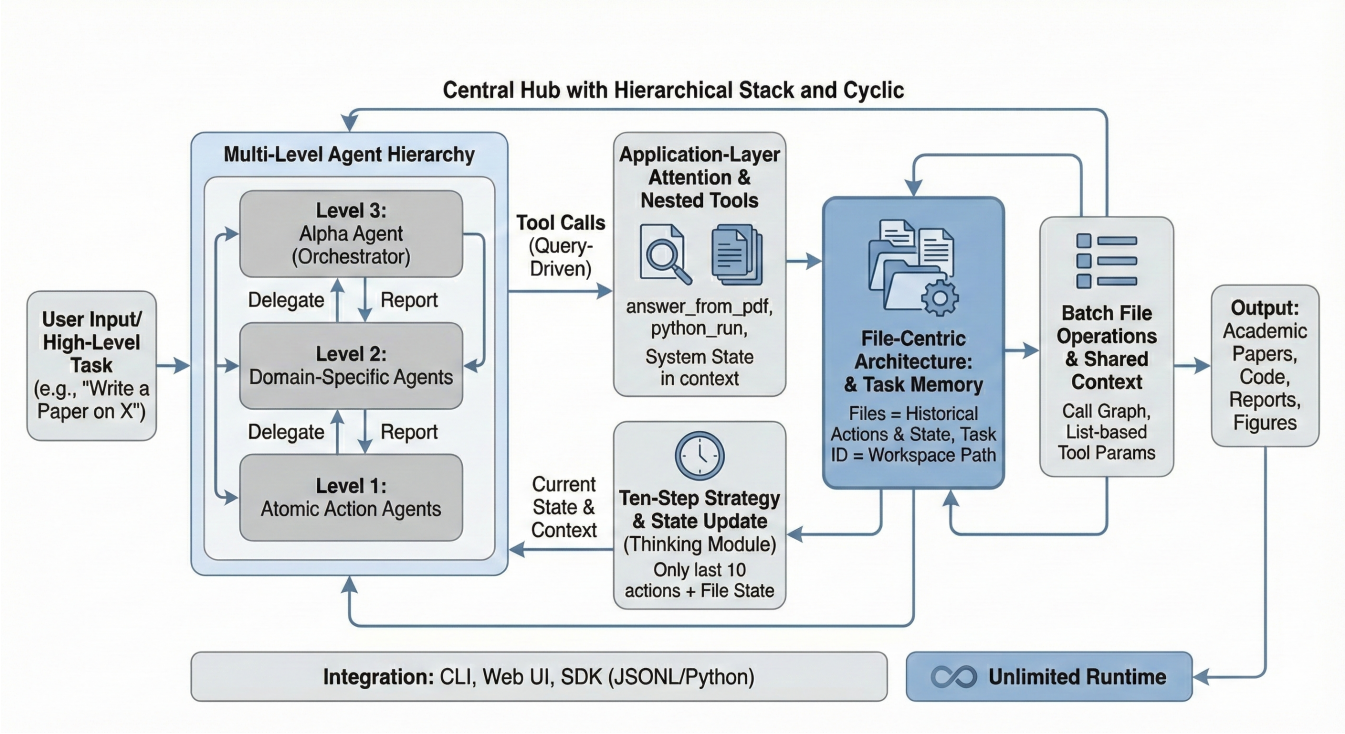

图1：**InfiAgent 框架**。InfiAgent 在以文件为中心的持久状态之上实现了分层执行栈。文件充当权威的任务记忆，而外部注意力机制则在有界推理上下文之外处理大型文档。周期性状态整合根据工作区快照刷新智能体的上下文。

4.1 以文件为中心的状态管理和任务记忆

InfiAgent 将持久化状态 $\mathcal{F}_t$ 具体化为文件系统上的结构化工作区。每项任务都会分配一个专用工作区目录，用于存储计划、中间产物、工具输出和验证日志。该工作区是任务进展的权威记录，并在不同执行步骤和会话之间持续保存。

为支持有界推理，InfiAgent 维护一个包含近期操作的定长缓冲区，并将其与工作区快照结合，在每个步骤构建智能体的推理上下文。在实践中，该缓冲区较小（例如包含 10 次操作），既保持短期执行连贯性，又不会增加上下文大小。

周期性状态整合。为支持长时程执行，InfiAgent 会定期将近期进展整合到持久状态中。在固定间隔内，工作区中存储的高级计划和进度标记会得到更新，随后使用更新后的状态快照刷新推理上下文。该机制在实践中实现了有界上下文重构，同时将详细的任务产物保存在持久化存储中。

4.2 多级智能体层级

InfiAgent 将智能体组织成树状层级结构（有向无环图）：

- **第 3 级（Alpha 智能体）：** 负责高层规划并将用户请求分解为子任务的编排器。它是决策树的根节点。
- **第 2 级（领域智能体）：** 包括编码智能体、数据收集智能体或论文撰写智能体等专门智能体。它们负责执行 Alpha 智能体委派的特定工作流。
- **第 1 级（原子智能体）：** 负责执行特定工具操作的智能体，例如网页搜索或文件输入输出。

这种层级结构支持边界清晰的串行执行。高层智能体将低层智能体作为可调用工具（智能体即工具），从而避免了扁平多智能体系统中智能体争相执行时常见的“工具调用混乱”。

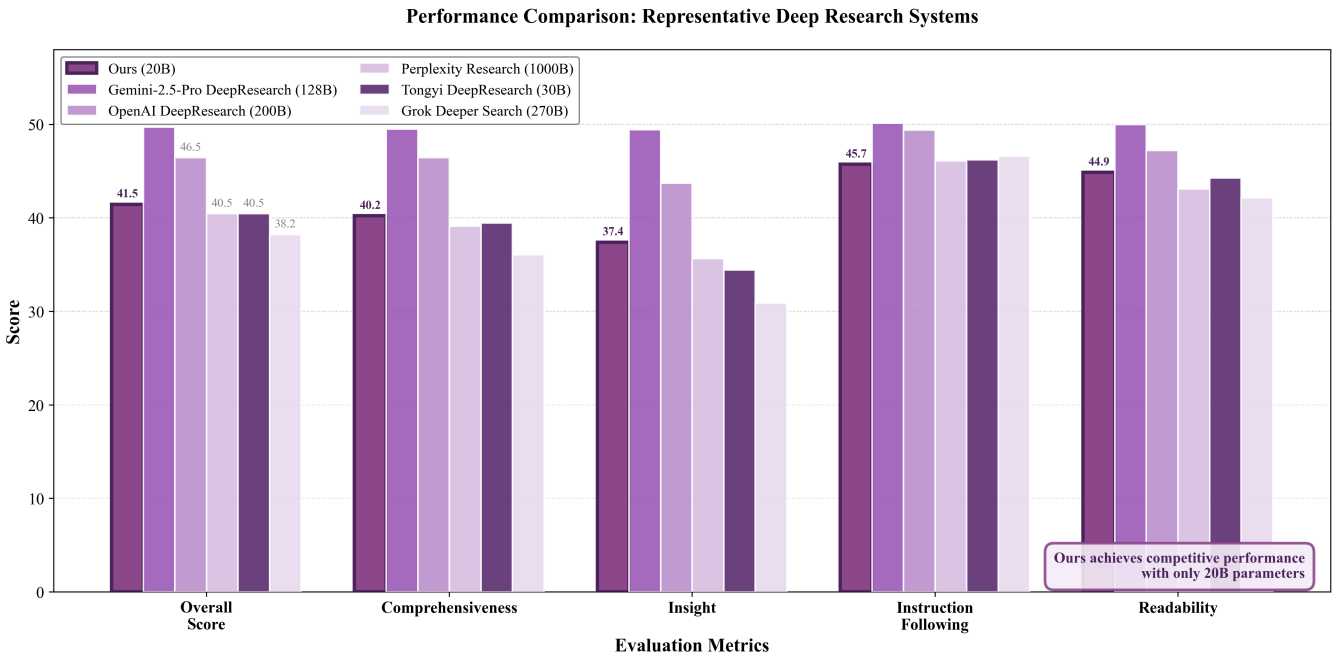

图 2：**DeepResearch 上的组件级对比**。 分数按评估维度细分。InfiAgent 在指令遵循和可读性方面表现出色，这与结构化状态管理和输出控制密切相关。

4.3 外部注意力流水线

为了处理海量信息（例如阅读 80 篇论文）而不造成上下文膨胀，InfiAgent 使用外部注意力流水线。当智能体需要从文档中获取信息时，它不会将文档加载到自己的上下文中，而是调用专用工具（例如从 PDF 获取答案）。该工具会启动一个临时、隔离的大语言模型进程来查询文档，并且只返回提取出的答案。

$$C_{main} \leftarrow C_{main} \cup Tool(Query, Document) \tag{5}$$

这实际上将“阅读”成本转移到了工具层，使主智能体的认知负荷保持较低，并专注于决策。该机制如同一个应用层注意力头，从海量外部数据源中筛选出相关信息。

5 实验

我们的实验旨在评估显式状态外化能否提升大语言模型智能体在长时程任务中的稳定性和可靠性。我们不以短时交互中的峰值表现为优化目标，而是关注在任务时长较长、文档集合规模较大且需要反复使用工具的情况下，执行是否足够鲁棒。

我们从两个互补维度评估 InfiAgent。首先，使用 DeepResearch 基准 [3]，这一标准化评估来衡量其通用研究能力，该基准旨在评估多步研究质量。其次，我们设计了一项长时程文献综述任务，直接对智能体在数百个执行步骤中的稳定性进行压力测试。在所有实验中，我们在适用情况下使用相同的模型骨干和相当的提示预算，并报告多次运行的平均结果，以降低方差。

5.1 DeepResearch 基准

我们在 DeepResearch 基准 [3], 上评估 InfiAgent。该基准旨在评估智能体开展多步研究的能力，研究过程涉及信息收集、综合和结构化报告。基准从四个维度对智能体进行评分：全面性、洞察力、指令遵循和可读性。

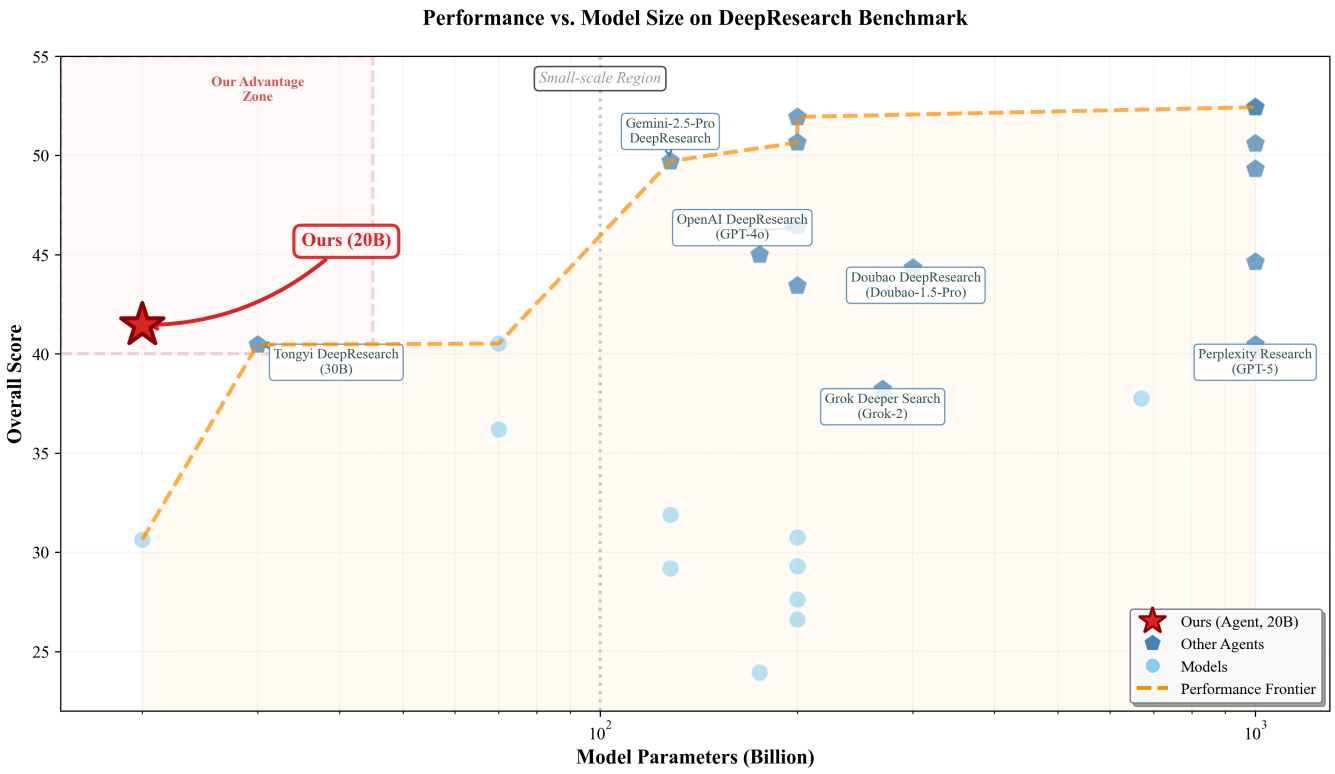

图 3：**DeepResearch 上的性能与模型规模**。 在同一基准测试中评估时，InfiAgent（20B）取得了与规模更大的专有智能体相当的性能，这表明其效率与性能之间的权衡有所改善。

设置。 我们使用一个拥有 200 亿参数的开源模型（gpt-oss-20b）配置 InfiAgent，且不进行任何特定任务微调。为单独考察智能体架构的影响，我们遵循标准基准测试协议，并尽可能采用默认任务指令。为确保不同系统之间的结果可比，所有报告的分数均直接取自基准测试评估。

总体性能。 图 3 展示了 DeepResearch 得分与模型规模的关系。InfiAgent 使用 20B 模型取得了 41.45 的总分，相较于依赖规模大得多的模型的系统，处于更有利的效率前沿。尽管不同智能体的绝对性能有所差异，这一结果表明，在长篇研究任务中，架构设计可以部分弥补模型规模的差距。

组件分析。 图 2 展示了各评估维度的得分细分。InfiAgent 在指令遵循和可读性方面表现尤为出色。我们认为，这一表现得益于明确的以文件为中心的状态和结构化执行流水线，它们有助于遵循任务指令并保持输出格式一致。在洞察力和全面性方面，其表现可与更大规模的模型相媲美，这表明稳定的状态管理有助于在长期交互中持续推理。

总体而言，这些结果表明，即使使用较小的骨干模型，显式外化智能体状态也能提升多步骤研究行为的可靠性和效率。

5.2 长期文献综述任务

为评估长时程稳定性，我们设计了一项文献综述任务，要求智能体遍历大量论文。每次运行都会提供一份包含 80 篇学术论文的列表，并要求智能体：（i）阅读每篇论文；（ii）撰写简短摘要；（iii）分配相关性评分。这项任务考验智能体在数百个步骤中持续使用工具并跟踪状态的能力。

评估指标（覆盖率）。我们关注的是任务完成可靠性，而非摘要质量。具体而言，我们衡量**覆盖率**，其定义为智能体能够生成非空摘要的论文数量

基于论文内容（即摘要中至少包含一条超出标题/元数据范围的内容陈述）。我们报告相同设置下重复运行的最大值、最小值和平均覆盖率。

设置	模型	Max	Min	Avg
主要结果（采用以文件为中心的状态；InfiAgent 与基线对比）				
InfiAgent	GPT-OSS-20B	**80**	15	67.1
InfiAgent	Gemini-3-Flash	**80**	**80**	**80.0**
InfiAgent	Claude-4.5-Sonnet	**80**	**80**	**80.0**
Claude Code	Claude-4.5-Sonnet	80	11	29.1
Cursor	Claude-4.5-Sonnet	5	0	1.0
Cursor	Gemini-3-Flash	1	0	0.1
消融（移除以文件为中心的状态；压缩的长上下文提示）				
无文件状态（压缩上下文）	GPT-OSS-20B	7	1	3.2
无文件状态（压缩上下文）	Gemini-3-Flash	25	20	21.1
无文件状态（压缩上下文）	Claude-4.5-Sonnet	77	11	27.7

表 1：**长时程任务可靠性（覆盖率）与消融**。在 80 篇论文文献综述任务中的覆盖率。上：以文件为中心的状态下的主要结果（InfiAgent 与基线）。下：消融以文件为中心的状态，并将其替换为压缩的长上下文提示。我们报告重复运行中的最大值/最小值/平均值覆盖率。

结果。表 1 显示，InfiAgent 始终能够实现较高的覆盖率：使用更强的骨干模型时可处理完整的 80 篇论文，即使使用 20B模型也能保持相当高的覆盖率。这一表现与所提出的以文件为中心的状态抽象一致，该抽象使智能体能够遍历较长的文档列表，而无需依赖不断增长的提示历史。

相比之下，基线智能体的平均覆盖率显著较低，且不同运行之间的方差较大。人工检查表明，失败通常表现为提前终止、跳过项目，或生成主要复述标题而非论文内容的摘要。

消融分析。 移除以文件为中心的状态并依赖压缩的长上下文提示，会导致所有模型的覆盖率显著下降（表 1 的底部区块）。即使使用更强的骨干模型，消融变体仍表现出较低的平均覆盖率和较大的运行间方差。这支持了以下观点：明确的持久化状态外置是长时程稳定性的关键因素，无法仅通过长上下文压缩完全替代。

比较的局限性。 我们并不声称绝对优于专有系统。商业智能体可能采用未公开的优化或终止启发式策略，在完整性与延迟或成本之间进行权衡。我们的目标是突出长时程行为上的定性差异：在延长的执行时程下，持久状态外化与更高且更稳定的覆盖率相关。

5.3 案例研究：真实世界应用与专家盲审

为验证该框架的能力，我们实现了**InfiHelper**，这是一个由 InfiAgent 框架驱动的半通用智能体。InfiAgent 提供底层架构（以文件为中心的状态、层级结构等），而 InfiHelper 则是专为复杂知识工作设计的具体实现。

InfiHelper 的多元应用场景 InfiHelper 并不局限于单一领域；它是一款多功能助手，能够基于同一核心架构处理各种真实世界任务：

- **计算生物学**：InfiHelper 成功开展了“干实验”，例如模拟细胞外基质（ECM）的蛋白质组成并预测相互作用。
- **物流与运营**：同一智能体为物流公司执行自动化轮班排班，并在复杂约束条件下优化劳动力分配。
- **学术研究**：InfiHelper 充当自动化研究助理，能够端到端地开展科学发现，从文献综述到论文起草。

学术成果的专家盲审 为评估复杂成果的质量，我们聚焦于学术研究场景。我们要求 InfiHelper 撰写完整篇幅的研究论文，并将论文提交给具有 EI/IEEE 会议经验的专家组成的盲审评审组。结果表明，InfiHelper 生成的论文达到了**人类水平的质量**，评审者认为其符合标准学术会议的录用标准。评审者特别指出，实验具有逻辑连贯性，且 LaTEX 格式正确；这些特点得益于 InfiAgent 框架稳定的状态管理和严格的输出约束。这表明，在 InfiAgent 的支持下，InfiHelper 能够有效地支持不同领域的严谨知识工作，有时甚至可以实现自动化。

6 讨论

显式状态外化无法解决的问题。尽管所提出的以文件为中心的状态抽象显著提升了长时程稳定性，但它本身并不会增强底层语言模型的推理能力。如果骨干模型得出了错误的中间结论或生成了有缺陷的产物，这些错误仍可能被写入持久化状态，并在后续步骤中传播。将持久化状态与有限上下文分离，虽然便于检查并有助于纠正，却无法消除在高风险场景中采用验证机制或人工监督的必要性。

长上下文不能替代持久化状态。第 5.2 节的消融结果凸显了长上下文与持久化状态之间的重要区别。即使使用上下文窗口较大的模型，以压缩的长上下文提示取代以文件为中心的状态，也会导致覆盖率大幅下降，并增加不同运行结果之间的方差。这表明，仅仅增加上下文长度不足以确保长时程执行的可靠性，而且显式状态外化发挥的作用在性质上不同于上下文扩展。

效率与延迟权衡。与单次处理、以上下文为中心的智能体相比，外化状态并实施分层执行会带来额外开销。文件系统操作、周期性状态整合以及串行化的智能体执行可能会增加延迟，尤其是在需要快速响应的任务中。因此，所提出的框架更适用于长期运行、知识密集型的工作流，而非实时交互应用。探索异步更新和部分并行仍是未来工作的重要方向。

范围与泛化。 我们的实证评估聚焦于研究导向任务，涉及长篇文档处理和多步信息综合。尽管这些设置揭示了长时程失效模式，但它们并不能充分代表其他智能体场景，例如响应式对话、具身交互，或外部状态快速变化的环境。还需在更广泛的任务范围内进行评估，以衡量其普适性。

更广泛的影响。 尽管存在这些局限，结果仍表明，长时程智能体应遵循一项通用设计原则：应将持久任务状态视为一等对象，并使其与语言模型的有界推理上下文相区分。这种分离有助于系统地分析智能体行为随时间的变化，并为未来在验证、纠正和协作式多智能体系统方面的研究奠定基础。

7 局限性

尽管 InfiAgent 在长时程任务中展现出强大的能力，但仍存在一些局限性。首先，多层级智能体层级结构引入了**延迟开销**。随着智能体树深度的增加，任务委派和结果聚合所需的时间也会增长，从而可能影响实时响应能力。其次，尽管以文件为中心的状态管理缓解了上下文丢失问题，**幻觉累积**仍然是一项风险，尤其是

在极长任务中使用较小模型（例如 20B）时。如果子代理将错误信息写入文件系统，且未被验证机制捕获，下游代理可能会传播这一错误。第三，当前架构为确保状态一致性而严格执行串行执行，这意味着它**不支持并行处理**。这限制了系统在本质上可并行化的任务中的效率，例如同时进行文献综述和实验编码。

8 结论

InfiAgent 提出了一种从基于上下文的代理转向以文件为中心的代理的范式转变。通过将状态外置到文件系统中，并采用严谨的多层级体系，InfiAgent 实现了近乎无限的运行时长和高度稳定性。我们的结果表明，这种**无需训练的** 架构使较小的开源模型（20B）能够在复杂研究基准上与最先进的专有代理竞争，为构建更易于使用、可扩展的自主研究系统铺平了道路。

AI 使用声明

根据 ACL 政策，我们声明在准备本稿件的过程中使用了 AI 辅助。框架示意图（图1）由 Gemini Nano Banana 模型生成。本文经过 Gemini 1.5 Flash 润色，以提升表达清晰度并修正语法。所有科学论断、实验设计和结果均由人类作者核实。

参考文献

[1] Humza Naveed、Asad Ullah Khan、Shi Qiu、Muhammad Saqib、Saeed Anwar、Muhammad Usman、Naveed Akhtar、Nick Barnes 和 Ajmal Mian。大语言模型综合概述。*ACM*智能系统与技术汇刊，16(5):1–72，2025。[2] Khanh-Tung Tran、Dung Dao、Minh-Duong Nguyen、Quoc-Viet Pham、Barry O’ Sullivan 和 Hoang D Nguyen。多智能体协作机制：大语言模型综述。*arXiv*预印本 *arXiv:2501.06322*，2025。[3] Parisa Shojaee 等人。深度研究代理中的状态幻觉。*arXiv*预印本 *arXiv:2506.11763*，2025。[4] Elliot Meyerson、Giuseppe Paolo、Roberto Dailey、Hormoz Shahrzad、Olivier Francon、Conor F Hayes、Xin Qiu、Babak Hodjat 和 Risto Miikkulainen。零错误解决百万步大语言模型任务。*arXiv*预印本 *arXiv:2511.09030*，2025。[5] Haowei LIU 等人。PC-Agent：面向 PC 复杂任务自动化的分层多智能体协作框架。*arXiv*预印本，arXiv:2502.14282，2025。[6] D. J. Moore。分层多智能体系统分类：设计模式、协调机制与工业应用。*arXiv*电子预印本，arXiv:2508.12683，2025。[7] X. Qu。人工智能代理综述：拓展技术及其他领域的可能性。*arXiv*电子预印本，arXiv:2508.11957，2025。[8] Wenhao LI 等人。多智能体强化学习中的语义对齐任务分解。*arXiv*预印本，arXiv:2305.10865，2023。[9] Xinye CAI 等人。用于进化多目标优化的网格约束分解方法。*IEEE*进化计算汇刊，22(4):564–577，2017。[10] Honguk WOO、Gwangpyo YOO 和 Minjong YOO。基于结构学习的非平稳环境强化学习任务分解。载于*AAAI*人工智能会议论文集，第8657–8665页，2022。[11] Xinjie ZHAO 等人。AGENTiGraph：面向交互式特定领域大语言模型聊天机器人的多智能体知识图谱框架。*arXiv*预印本，arXiv:2508.02999，2025。[12] Zhiheng Xi、Yiwen Ding、Wenxiang Chen、Boyang Hong、Honglin Guo、Junzhe Wang 等人。AgentGym：在多样化环境中进化基于大语言模型的智能体。*arXiv*预印本 *arXiv:2406.04151*，2024。

[13] Z Guan、X Kong、F Zhong 等人。Richelieu：面向人工智能外交的自进化大语言模型智能体。载于神经信息处理系统进展（*NeurIPS*），2024。

A 附录：详细实验结果

表 2 列出了 InfiAgent 和各种基线系统在 DeepResearch 基准上的完整评估结果。

模型/智能体	总体	Comp.	洞察力	指令遵循	阅读	引用准确率	有效引用	参数量（十亿）
tavily-research（GPT-5）	52.44	52.84	53.59	51.92	49.21	-	-	1000
thinkdepthai-deepresearch（GPT-5）	52.43	52.02	53.88	52.04	50.12	-	-	1000
cellcog（GPT-4o）	51.94	52.17	51.90	51.37	51.94	-	-	200
salesforce-air-deep-research（GPT-4o）	50.65	50.00	51.09	50.77	50.32	-	-	200
gensee-search-gpt-5（GPT-5）	50.60	50.06	50.76	51.31	49.72	32.94	21.06	1000
gemini-2.5-pro-deepresearch	49.71	49.51	49.45	50.12	50.00	78.30	165.34	128
langchain-open-deep-research（GPT-5）	49.33	49.80	47.34	51.05	48.99	34.74	22.44	1000
openai-deepresearch (GPT-4o)	46.45	46.46	43.73	49.39	47.22	75.01	39.79	200
claude-research（Claude-3.7-Sonnet）	45.00	45.34	42.79	47.58	44.66	-	-	175
kimi-researcher (Kimi-K2)	44.64	44.96	41.97	47.14	45.59	-	-	1000
doubao-deepresearch (Doubao-1.5-Pro)	44.34	44.84	40.56	47.95	44.69	52.86	52.62	300
langchain-open-deep-research (GPT-4o)	43.44	42.97	39.17	48.09	45.22	49.10	29.49	200
我们的 (gpt-oss-20b)	**41.45**	**40.22**	**37.39**	**45.72**	**44.87**	-	-	20
英伟达 AIQ 研究助手	40.52	37.98	38.39	44.59	42.63	-	-	70
通义 DeepResearch 30B-A3B	40.46	39.46	34.44	46.22	44.27	-	-	30
Perplexity 研究 (GPT-5)	40.46	39.10	35.65	46.11	43.08	82.63	31.20	1000
Grok 深度搜索 (Grok-2)	38.22	36.08	30.89	46.59	42.17	73.08	8.58	270
Sonar 推理专业版	37.76	34.96	31.65	44.93	42.42	45.19	9.39	670
Sonar 推理	37.75	34.73	32.59	44.42	42.39	52.58	13.37	70
Claude 3.7 Sonnet 搜索版	36.63	35.95	31.29	44.05	36.07	87.32	24.51	-
Sonar 专业版	36.19	33.92	29.69	43.39	41.07	79.72	16.75	70
gemini-2.5-pro-preview	31.90	31.75	24.61	40.24	32.76	-	-	128
gpt-4o-search-preview	30.74	27.81	20.44	41.01	37.60	86.63	5.05	200
sonar	30.64	27.14	21.62	40.70	37.46	76.41	10.68	20
gpt-4.1	29.31	25.59	18.42	40.63	36.49	89.85	4.27	200
gemini-2.5-flash-preview	29.19	28.97	21.62	37.80	29.97	-	-	128
gpt-4o-mini-search-preview	27.62	24.24	16.62	38.59	35.27	81.69	4.62	200
gpt-4.1-mini	26.62	22.86	15.39	38.18	34.49	84.54	4.10	200
claude-3-5-sonnet-with-search	23.95	21.28	16.20	32.41	29.87	94.06	9.35	175

表 2：DeepResearch 基准上的详细性能分解。全面性（Comp.）、洞察力、指令遵循（Inst. Fol.）和可读性（Read.）的得分均已归一化。参数数量根据公开报告进行估算。