基于大语言模型的多智能体系统异常综述(特邀)

张珑耀[1],温东新[1],马庄宇[1],舒燕君[1],李庆[1,2],刘明义[1],左德承[1]
(1.哈尔滨工业大学计算学部,黑龙江 哈尔滨 150001;2.江苏自动化研究所,江苏 连云港 222006)

摘　要: 基于大语言模型(LLM)的多智能体系统(MAS)虽在处理复杂任务方面展现出巨大潜力,但其分布式特性与交互不确定性易引发多样化异常,威胁系统可靠性。为系统化识别并分类此种异常,进行全面综述。研究选取 7 个代表性 MAS 及相应数据集,收集 13 418 段运行轨迹,采用 LLM 初步分析与专家人工校验相结合的方法进行数据分析。构建一个涵盖模型理解感知异常、智能体交互异常、任务执行异常和外部环境异常 4 个层级的细粒度异常分类框架,并结合典型案例揭示各类异常产生的内在逻辑与外部诱因。统计分析显示:模型理解感知异常占比最高,其中"上下文幻觉"和"任务指令误解"是主要问题;智能体交互异常占 16.8%,"信息隐瞒"是主因;任务执行异常占 27.1%,主要表现为"决策重复出错";外部环境异常占 18.3%,以"记忆冲突"为主。此外,模型理解感知异常作为根源性诱因,引发其他层级的异常,凸显了提升模型基础能力的重要性。此分类和根源分析旨在为构建高可靠的基于 LLM 的 MAS 提供理论支撑与实践参考。

关键词: 大语言模型;智能体;多智能体系统;异常统计;异常分类
中图分类号: TP18　　**文献标志码:** A　　**DOI:** 10.19678/j.issn.1000-3428.0252754

A Review of Anomaly in Large Language Model-Based Multi-Agent Systems (Invited)

ZHANG Longyao[1], WEN Dongxin[1], MA Zhuangyu[1], SHU Yanjun[1], LI Qing[1,2], LIU Mingyi[1], ZUO Decheng[1]
(1. Faculty of Computing, Harbin Institute of Technology, Harbin 150001, Heilongjiang, China;
2. Jiangsu Automation Research Institute, Lianyungang 222006, Jiangsu, China)

【Abstract】 Large Language Model (LLM)-based Multi-Agent System (MAS) has demonstrated significant potential in handling complex tasks. Their distributed nature and interaction uncertainty can lead to diverse anomalies that threaten system reliability. This paper presents a comprehensive review, identifying and classifying these anomalies systematically. Seven representative multi-agent systems and their corresponding datasets are selected, accounting for 13 418 operational traces, and a hybrid data analysis method is employed, combining preliminary LLM analysis with expert manual validation. A fine-grained, four-level anomaly classification framework is constructed, encompassing the following anomalies: model understanding and perception, agent interaction, task execution, and external environment. Typical cases are analyzed to reveal the underlying logic and external causes of each type of anomaly. Statistical analysis indicates that model understanding and perception anomalies account for the highest proportion, with "context hallucination" and "task instruction misunderstanding" being the primary issues. Agent interaction anomalies represent 16.8%, primarily caused by "information concealment". Task execution anomalies account for 27.1%, mainly characterized by "repetitive decision errors". External environment anomalies account for 18.3%, with "memory conflicts" as the predominant factor. In addition, the model perception and understanding of anomalies often act as root causes, triggering anomalies at other levels, highlighting the importance of enhancing fundamental model capabilities. These classification and root cause analyses aim to provide theoretical support and practical reference for building highly reliable LLM-based multi-agent systems.

【Key words】 Large Language Model (LLM); agent; Multi-Agent System (MAS); anomaly statistics; anomaly classification

0　引言

在人工智能技术的浪潮中,多智能体系统(MAS)以其分布式协作与动态决策能力,已成为智能客服、自动驾驶、供应链管理等复杂场景的核心解决方案[1]。尽管如此,传统基于强化学习的 MAS 在处理复杂动态需求时,仍存在响应僵化、协作效率低、容错能力不足等瓶颈[2]。近年来,大语言模型

基金项目: 国家重点研发计划(2024YFB4506000);国家自然科学基金(61202091, 62171155)。
作者简介: 张珑耀(CCF 学生会员),男,硕士,主研方向为边缘计算容错技术、多智能体系统异常分析;温东新、马庄宇、舒燕君(通信作者)、李庆、刘明义、左德承,博士。
收稿日期: 2025-07-14　　**修回日期:** 2025-10-13　　**E-mail:** yjshu@hit.edu.cn

(LLM)取得了突破性进展,其凭借强大的语义理解、上下文推理与生成能力,显著提升了单个智能体认知水平[3],并重构了多智能体协作范式,实现了从任务规划到执行的全流程智能化升级,为复杂动态环境下的自主决策提供了全新可能[4]。

然而,基于 LLM 的 MAS 由于其生成结果的固有不确定性以及在动态环境中频繁交互所带来的复杂性,使得其在运行过程中总是会发生各种异常情况[5]。这些异常不仅威胁系统的可用性,更可能引发连锁性失效风险。为深入理解异常的产生机制与表现形式,并依其类别设计针对性的容错策略,系统化地识别与分类异常已成为提升 MAS 鲁棒性的关键挑战。

目前,已有研究尝试对 MAS 中的异常进行分类,并提出了一些具有启发性的框架。然而,这些分类方法存在维度单一、粒度较粗的问题。CEMRI 等[6]提出一种基于智能体系统异常发生时间(执行前/中/后)的分类方法。该方法为后续更细致、更深入的分类研究奠定了理论基础,但以时间维度划分无法揭示异常产生的具体原因与作用环节,因此难以满足精准定位需求。WANG 等[7]提出的分类方法主要从智能体内与智能体间两个层次进行区分,将推理、规划、行动等错误大多归结为模型幻觉的表现。这种划分虽然为异常分类研究提供了一个整体视角,但在实践中过于笼统,难以支持精确的异常诊断。因此,构建层次化、细粒度的异常分类体系仍是研究重点。

鉴于上述分类方法在维度与粒度上的局限,缓解异常必须依托于更精细的系统设计变更,本文提出了一个涵盖模型理解感知、智能体交互、任务执行和外部环境四大层级的细粒度异常分类框架。本文的主要贡献在于:1)提出了一个全新的、基于 MAS 运行核心环节的层次化异常分类体系;2)基于多智能体的大规模真实运行 Trace 数据,对各类异常进行了量化统计与根源分析,揭示异常产生的内在逻辑与外部诱因;3)展望未来工作,为 MAS 异常的根因分析及智能化修复提供理论与实践指导,助力系统可用性与自治能力提升。

1 基于 LLM 的 MAS

近年来,随着人工智能的快速发展,LLM 展现出强大的语言理解与生成能力,逐渐成为 MAS 中的核心组件。相比传统依赖规则驱动或轻量模型的 MAS,基于 LLM 的系统在环境感知、任务协作与推理能力方面取得了显著提升。智能体之间的交互模式、任务分工及整体智能水平也因此得以增强。为系统探讨此类系统的构建与运行机制,本节将从 MAS 的层次化架构与异常处理关注两个方面展开分析,揭示其关键组成、运行逻辑及当前异常研究的最新进展,为后续章节的异常分析奠定基础。

1.1 MAS 架构

基于 LLM 的 MAS 通过集成多个具备语言理解与生成能力的智能体,实现对复杂任务的协同处理。系统架构通常围绕工具使用、自主决策、长期记忆与智能体间通信等核心能力展开。每个智能体不仅需要具备感知环境和独立行动的能力,还应能在多轮交互中保持上下文一致性,响应协同计划,并根据外部输入或其他智能体的反馈灵活调整自身行为。

目前已有部分研究针对基于 LLM 的 MAS 提出了层次化的架构划分逻辑[8-9],以刻画各组成部分在任务执行过程中的角色与关系。在此基础上,本文进一步细化并提出了一个自上而下分为 4 个层级的系统架构,包括用户层、任务规划层、智能体层和模型层。各层协同工作,构建出一个可处理复杂任务、多智能体协同决策的智能系统,其架构如图 1 所示。

用户层(User Layer)作为系统入口,负责接收用户输入的任务提示与目标定义。例如,用户提出“分析某公司近五年财报并生成可视化报告”的需求,系统需通过自然语言解析明确核心目标与约束条件。该层不仅要求精准理解用户意图,还需将抽象任务转化为可执行的语义框架,为后续任务分解提供基础。

任务规划层(Task Planning Layer)负责将宏观目标拆解为具体子任务,并确定执行优先级与依赖关系。例如,上述财报分析任务可划分为“数据采集”“指标计算”“图表生成”等子任务,并规划其顺序(如采集需先于计算)。该层关键在于动态适应性:结合历史经验与 LLM 推理能力,优化子任务划分与执行路径,避免僵化规划带来的偏差与资源浪费。

智能体层(Agent Layer)是系统的执行中枢,由多个异构智能体组成,各自承担特定角色(如数据采集员、分析师、可视化工程师),并配备相应工具(如爬虫、统计库、绘图接口)与记忆模块。智能体通过通信协议(如广播或协商机制)实现协同。记忆模块至关重要:短期记忆维护当前任务上下文,长期记忆存储跨任务知识图谱,支持智能体持续学习与错误修正。

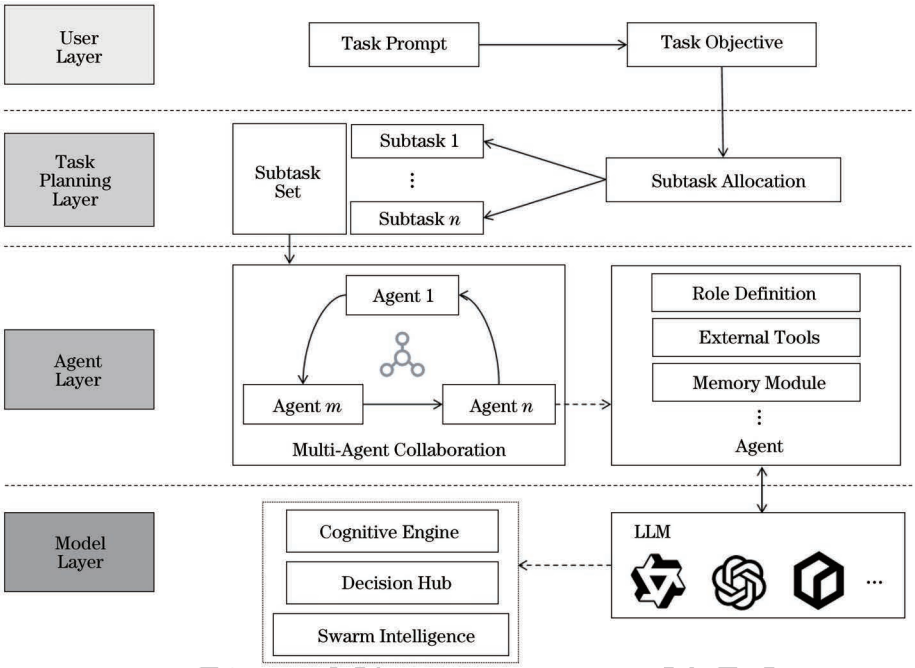

图 1 基于 LLM 的 MAS 基本架构

Fig. 1 Basic architecture of LLM-based MAS

模型层(Model Layer)作为底层支撑,融合 LLM 的语义理解能力与群体智能机制。认知引擎负责解析需求、生成建议并修正逻辑错误;决策中枢基于智能体反馈与环境状态优化策略(如资源分配、异常优先级);群体智能机制(如强化学习、博弈模型)则用于优化协作、减少冲突与循环依赖。例如,系统可通过监控智能体状态动态调整分配策略,或在工具异常时启动备用方案。

通过分层设计与协同机制,基于 LLM 的 MAS 可有效理解用户意图、拆解任务,并借助智能体协作与 LLM 赋能,自动完成复杂任务,提升系统智能与执行效率。

1.2 MAS 异常

MAS 异常是指系统在任务执行过程中因感知、理解或协作等环节出错,导致行为偏离预期目标的现象。随着基于 LLM 的 MAS 在自动化协作和复杂任务处理中的广泛应用,系统面临的异常类型也日益多样化。本节将从两个维度展开:一是回顾学术界在 MAS 异常研究方面的进展与不足;二是梳理工程实践中典型系统的应对策略与成效。

学术研究方面,HUANG 等[10]系统评估了多种智能体结构的鲁棒性,并提出输出质疑机制与审校智能体,有效提升了代码生成和数学任务中的稳定性;BARBI 等[11]关注“单点失败”问题,在行动预测阶段引入监控与干预机制,防止系统性能崩溃;SUNG 等[12]提出 VeriLA 框架,用于验证代理故障,通过人类标准和自动分析模块提升异常可解释性。EPPERSON 等[13]总结构建基于 LLM 的 MAS 的三大挑战:错误难定位、缺乏调试支持和工具配置难,并开发了交互式调试工具 AGDebugger,以提供消息编辑、历史重置和可视化导航等功能。总体来看,学术界的探索多集中于提升系统整体鲁棒性或应对特定异常,缺乏细粒度、系统化的异常分类体系。

在工程实践中,CAMEL 框架[14]通过任务重分配与智能体重启机制,实现了智能体角色与任务规划的动态调整,有效缓解了智能体失效带来的影响。Magentic-One 架构[15]通过双循环机制保障任务连续性与系统鲁棒性。随着系统规模与复杂度提升,智能体行为监控与异常检测愈发重要。AgentOps[16]等工具实现了对交互轮数、中间消息与错误率等指标的实时监控,为异常检测与系统可靠性评估提供了强有力的支撑。这些工程手段弥补了理论模型的局限,推动了异常应对实践的发展。

2 分析方法

为了系统性地识别、理解并归类基于 LLM 的 MAS 在实际运行过程中可能出现的各类异常,采用如图 2 所示的异常分析流程,分为数据收集和数据分析两个环节,旨在通过一系列结构化的步骤,确保异常分析的全面性与准确性。本章将详细阐述此流

程。数据收集环节通过选取代表性系统与数据集来广泛收集运行轨迹(Trace)。在数据分析环节,首先运用 LLM 进行初步自动化异常识别与分类,并结合专家人工校验与深度分析,以保障结果的可靠性。在此基础上,进一步构建异常分类框架并进行统计,确定各类异常的分布比例。

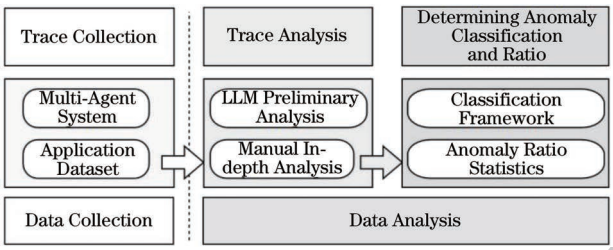

图 2 基于 LLM 的 MAS 异常分析流程

Fig. 2 Process of anomaly analysis in LLM-based MAS

2.1 数据收集

为确保所收集数据的广泛代表性与多样性,选取了当前领域内具有显著特点、架构各异且具备广泛应用前景的 7 个 MAS。同时,匹配了 7 个涵盖不同领域和任务类型的公开数据集,旨在模拟真实世界中 MAS 可能遭遇的复杂应用场景。本研究选取的 MAS 覆盖了从任务规划、协作开发到通用助理等不同应用方向,表 1 给出了具体的智能体系统和对应数据集,其中共收集到 13 418 段智能体在解决问题时产生的 Trace[17]数据,涵盖任务分解、子任务分配、智能体的思考/推理过程、执行的行动、调用的工具及工具反馈结果等环节,其中存在大量系统运行失败、行为异常或效率低下的案例,为理解智能体行为逻辑与决策路径提供了直观依据。

表 1 智能体系统与对应的测试数据集

Table 1 Agent systems and corresponding datasets

Agent System	Dataset	Application
AG2[18]	GSM-Plus[19]	Solving Math Problems
AppWorld[20]	AppWorld BenchMark (Test-C)	Simulating Mobile Application Operations
HyperAgent[21]	SWE-Bench Lite[22]	Software Engineering Tasks
MagenticOne[15], OpenManus[23]	GAIA[24]	General AI Assistant
MetaGPT[25], ChatDev[26]	MMLU[27], ProgramDev[6]	Large-Scale Tasks and Program Development

2.2 数据分析

为系统性地对大规模、高复杂度的 MAS 运行轨迹数据中的异常进行分析,采用了融合 LLM 分析与人工校验的混合方法。完整 Trace 数据的结构化字段如图 3 所示(彩色效果见《计算机工程》官网 HTML 版,下同),其中主要涵盖任务信息(任务编号、任务内容)、智能体交互信息(智能体角色、智能体动作、消息内容)以及任务结果评估(是否成功、标准答案)等内容。判定 Trace 中存在异常主要基于两个核心标准:一是任务执行最终产生错误的结果;二是即便任务结果最终正确,但分析 Trace 发现智能体在执行过程中存在显著偏离预期逻辑、低效、冗余或潜在风险的行为(如无谓循环、关键信息缺失下的盲目尝试、违反约束条件等)。

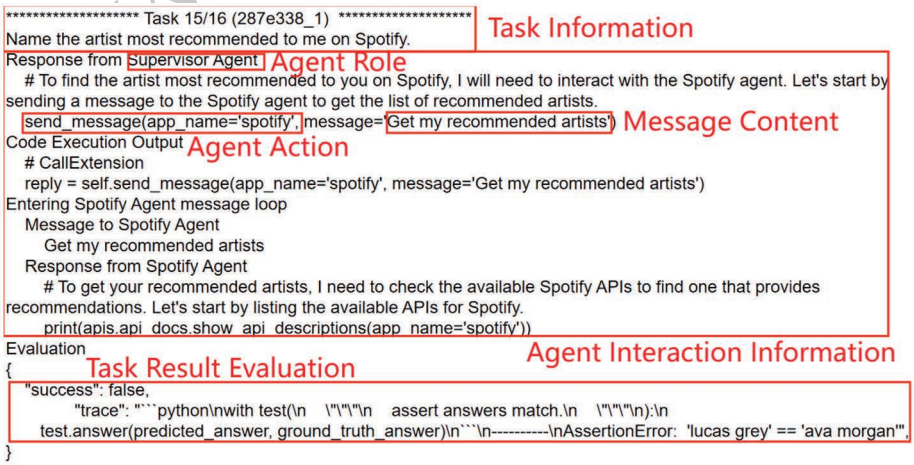

图 3 Trace 数据结构化字段构成

Fig. 3 Composition of structured fields in Trace data

在数据分析环节,研究使用 Gemini 2.5 Pro[28]对 Trace 数据进行自动化标注。具体而言,以图 4 为例,将 Trace 数据作为输入,依据预先定义的异常判定标准构建提示词,引导 Gemini 模型对 Trace 进

行系统分析。模型的输出包括两个部分：一是判断该 Trace 是否存在异常并给出违背的判定标准；二是对识别出的异常进行初步类型归类与原因推断，为后续人工复核与分类奠定基础。

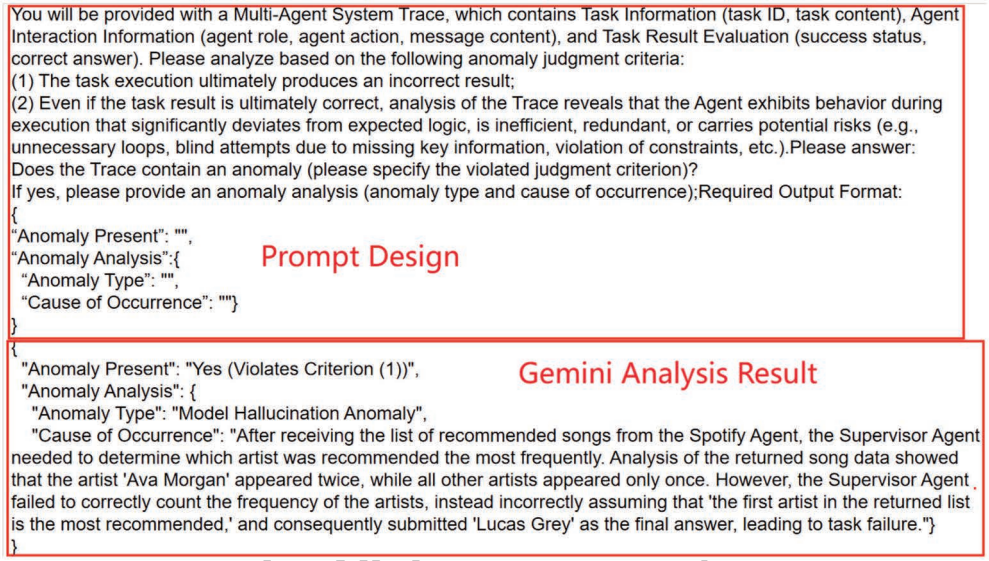

图 4 基于 LLM 的 Trace 分析方法

Fig. 4 LLM-based Trace analysis methodology

随后，本文的两位作者对 LLM 标记的案例进行逐一审核，判断 LLM 分析结果的正确性，对存在争议的案例进行第三位作者复核，保证分析的准确性。表 2 给出了最终得到的智能体系统的两种异常行为的次数和异常总数。

表 2 智能体系统异常行为统计

Table 2 Agent systems anomalous behavior statistics

Anomaly Behavior	Count
Result Error	5 675
Abnormal Output	2 066
Total Anomalies	7 741

为了确定异常的分类与比例，使用层次化的异常分类体系，根据异常分析的结果，结合异常在 MAS 中的层级（模型层、智能体层、任务规划层、用户层）将异常分为 4 个大类。每个大类中，再按照异常的表现特征细分为若干小类。

研究采用了在定性数据分析中广泛使用的开放式编码（Open Coding）方法[29]，结合 LLM 给出的初步分析结果通过手工标注的方法来对异常样本进行分类。本文的两位作者先选取一些代表性样本进行预分类，以熟悉各类异常特征并统一理解标准。在预分类阶段，主要选择在相关工作中经常被讨论且具有代表性的异常，例如模型幻觉、记忆丢失等。在预分类完成之后，两位作者开始对所有的异常样本进行“双盲”独立分类。在分类过程中，如果一名作者发现一个样本不符合现有的异常类别，该作者会与另一名作者协商，并根据需要增加一个新的类别。

最后，两名作者比较他们的分类结果，分类结果的一致性通过 Cohen Kappa 系数[30]来衡量，表 3 是不同 Kappa 系数的一致性程度。两位作者的手工标注结果的 Kappa 系数是 0.84，这代表着两人对异常的标注有极佳的一致性程度。对于标注不一致的样本，由第三位作者进行仲裁，通过迭代讨论解决冲突，最终达成共识。对于一个智能体异常可能产生于多个层次的重叠情况，则相应地将这个异常归类到多个层次上。

表 3 Kappa 系数代表的一致性程度

Table 3 Kappa coefficient for degree of agreement

Kappa Coefficient	Degree of Agreement
<0.40	Poor
$[0.40, 0.60]$	Moderate
$[0.61, 0.80]$	Good
>0.80	Excellent

由于本文的研究重点在于 MAS 异常类型的划分与体系构建，因此未进一步展开异常间复杂因果关系的刻画。整体分类立足于 MAS 运行的核心环节，遵循从个体到协作、从内部逻辑到外部依赖的递进式分析逻辑，系统性地归纳了不同来源和表现形式的异常。

3 MAS 异常的分类

MAS 在复杂环境中的应用日益广泛，但其分布式、自治性和动态交互的特性也带来了多样化的异

常。本章将 MAS 中常见的异常类型进行详细统计和分类，并结合实际案例进行阐述。

MAS 的异常分类需从系统运行的核心环节逐层展开，其逻辑可划分为 4 类：模型自身能力缺陷、智能体间交互冲突、任务执行过程偏差以及外部环境制约，如图 5 所示。

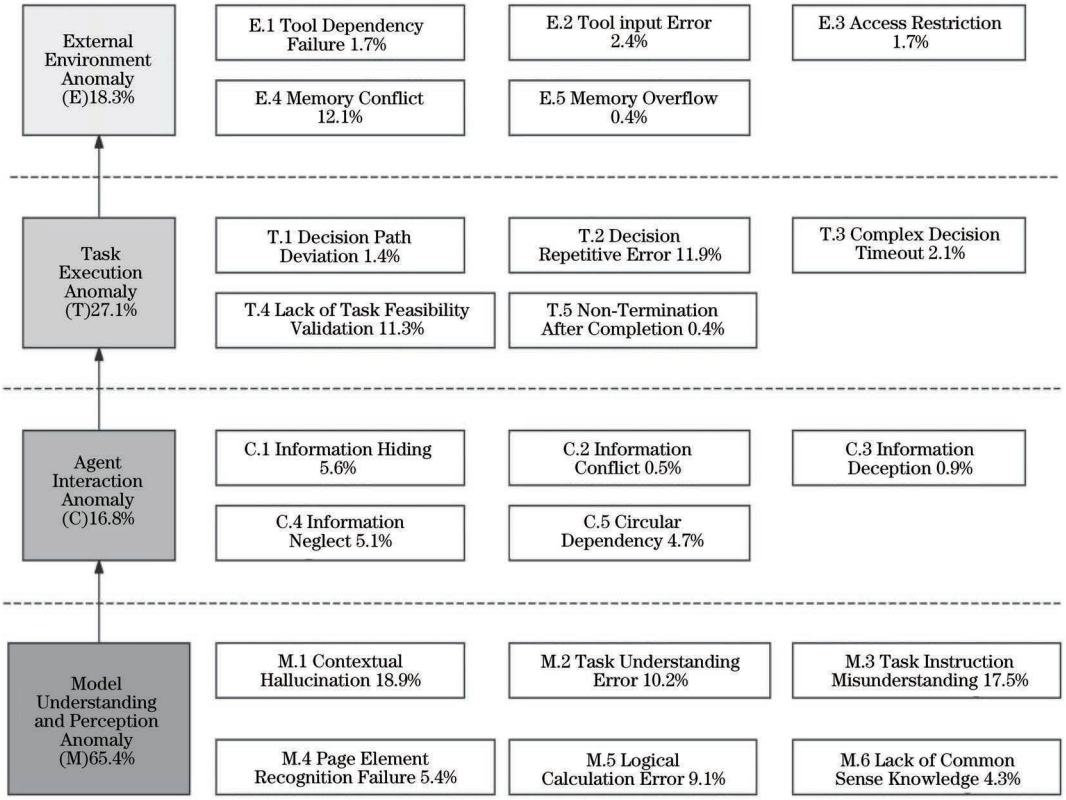

图 5 基于 LLM 的 MAS 异常类型

Fig. 5 Anomaly types in LLM-based MAS

最底层是模型理解感知异常，源于 LLM 在语义理解、任务识别或推理能力上的局限，体现智能体个体层面的能力瓶颈；其上是智能体交互异常，聚焦多智能体协作中信息传递不畅、冲突未解或循环依赖等问题，反映协作机制的不健全；再往上是任务执行异常，指系统在具体任务推进中出现策略失当、目标偏离或资源耗尽等流程控制类问题，是协作行为的直接输出层；最高层则是外部环境异常，强调系统与工具、服务、记忆等外部要素交互中的障碍，体现外部依赖的不稳定性。这一分类路径既覆盖了智能体系统运行的各个关键环节，也揭示了异常问题在层层递进中的演化逻辑，为异常的系统性识别、定位和修复提供了理论基础。

模型理解感知异常层级占据了最大的比重(65.4%)，其中“上下文幻觉”(18.9%)和“任务指令误解”(17.5%)是主要的认知偏差，这反映出 LLM 智能体在复杂语义理解方面仍面临瓶颈。此外，“逻辑计算错误”也占到 9.1%，表明基础推理能力仍有提升空间，而“页面元素识别失败”和“缺少常识知识”的占比分别为 5.4%和 4.3%。

智能体交互异常层级占总体的 16.8%，其中“信息隐瞒”(5.6%)和“信息忽略”(5.1%)是导致协作失效的主要诱因，这表明智能体之间信息共享的透明度和有效性至关重要。同时，“循环依赖”占 4.7%，而“信息冲突”和“信息欺骗”的比例较低，分别为 0.5%和 0.9%。

任务执行异常层级则占据 27.1%的比例，主要表现为“决策重复出错”(11.9%)和“任务可行性缺少验证”(11.3%)，这凸显了智能体在动态决策过程中的稳定性和前期验证环节的不足。其他如“复杂决策超时”(2.1%)和“完成后不终止”(0.4%)的异常也偶发性存在。

最后，外部环境异常占比 18.3%，其中“记忆冲突”问题最为突出，达到 12.1%，表明在复杂环境中管理和同步记忆状态是一个亟待解决的关键挑战。其他如“工具依赖故障”(1.7%)、“工具输入错误”(2.4%)和“访问受限”(1.7%)也构成了外部环境制约的一部分，而“记忆溢出”的发生率最低，仅为 0.4%。

总结：这种量化统计揭示了异常分布的幂律特征——少数高频异常贡献了系统故障的绝大部分，

为后续异常的精准定位与有效的修复提供了基础。

4 MAS 异常分析

4.1 模型理解感知异常

模型理解感知异常主要指智能体在感知环境、理解问题或执行指令过程中，由于其内部模型或认知能力的局限性而导致的错误。

上下文幻觉(M.1)表现为智能体在信息缺失时未能主动询问用户或获取更多信息，而是直接做出不准确或错误的假设。例如，当智能体被问及“A 病了一段时间，在此期间他每个月瘦了 10 磅。如果他最后的体重是 70 磅，请问他最初的体重是多少?”时，在缺少病程时长信息的情况下，它假设病了 70 个月给出了答案。

问题理解错误(M.2)是指智能体对问题的语义理解出现偏差，导致其后续处理或输出与原问题不相关或完全错误。例如，智能体被问及“A 每次跑步的速度是多少?”时，它可能错误地理解了“速度”的语义，转而去计算了“跑步距离”。

任务指令误解(M.3)定义为智能体对任务目标或指令的解释不准确，未能正确理解其核心意图。如图 6 所示，指令为“在 Spotify 播放器上不断切换到上一首歌曲，直到找到 Luna Starlight”。智能体尝试调用 previous_song API，但遭遇“未授权访问”的错误。这表明智能体未能识别“登录授权”为指令执行的隐性前提，导致任务失败。

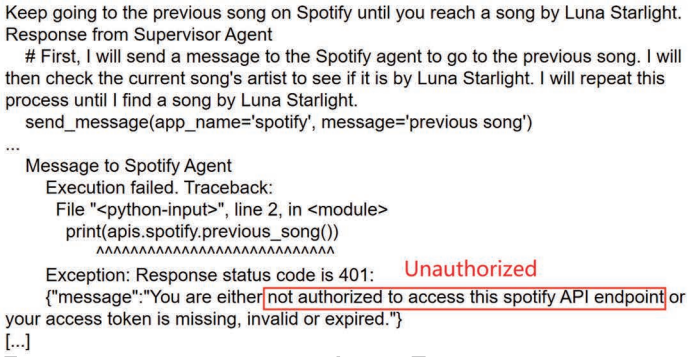

图 6 任务指令误解

Fig.6 Task instruction misunderstanding

页面元素识别失败(M.4)指的是智能体在与图形用户界面(GUI)或网页进行交互时，无法准确识别并定位到如输入框、按钮等关键的用户界面元素，从而导致其预期的操作无法执行。

逻辑计算错误(M.5)指智能体在执行数学计算或逻辑推理步骤时出现错误，导致结果不正确。这包括在计算金钱剩余时，智能体将“消费总金额的 3/5，还剩余 2/5”错误地计算为“剩余 1/5”。此外，在统计推荐列表中最常被推荐的艺术家时，智能体可能存在逻辑缺陷，错误地认为推荐列表中第一个艺术家就是最常被推荐的，而根本没有统计出现的频率。

缺乏常识知识(M.6)指智能体在推理过程中，因缺少基本的常识性信息而导致判断失误或错误行为。例如，在计算被子布料需求时，智能体可能不知道被子有两面需要布料这样的常识性信息。

总结：模型理解感知异常的根源在于模型自身的理解与知识能力的双重不足。前者导致其难以正确解析复杂、模糊或干扰性输入，后者则源于知识体系的不完整或错误。因此，提升模型感知与推理的准确性，需要从增强其语义理解能力和补全高质量知识入手。

4.2 智能体交互异常

智能体交互异常主要发生在 MAS 内部，智能体之间进行信息交换、协商和协作时。

信息隐瞒(C.1)指智能体在交互过程中未能主动共享关键信息，导致其他智能体无法做出正确判断或行动。如图 7 所示，为获取 Spotify 播放器的音乐库信息，主控智能体请求 Spotify 智能体调用相关应用程序接口(API)。后者指出需提供 access_token，但未说明如何获取。双方多次信息缺失与误解，最终导致任务失败。

信息冲突(C.2)指智能体之间在交互中产生意见或决策上的矛盾，且未能有效解决。例如，执行智能体已经多次报告任务无法执行，但是管理智能体仍然忽视这些反馈，继续发出重复指令，导致两者之间发生持续冲突。

信息欺骗(C.3)指智能体间在交互中传递虚假、不准确或具有误导性的信息，试图欺骗其他智能体。例如，在计算“12 月 31 日到 7 月 19 日有多少天”时，智能体没有进行计算就直接返回了 170 天(实际天数大于 170)，从而导致最终计算结果错误。

信息忽略(C.4)指智能体未能识别、接收或处理其他智能体发出的重要指令或信息,导致任务执行偏离预期。例如,主控智能体要求将计算结果放入\boxed{}中输出。但辅助智能体尽管计算正确,却将结论语句放入框中,未遵循输出格式要求,反映其对关键信息的忽略。

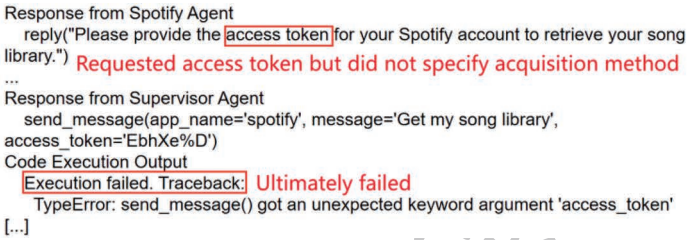

图 7 信息隐瞒

Fig.7 Information hiding

循环依赖(C.5)是指智能体之间或智能体与工具之间形成相互等待的关系,导致任务无法推进,陷入无效循环。如图 8 所示,当智能体在信息不足时重复向其他智能体或工具提出相同的请求,而无法获得新的有效信息,从而产生重复的对话和操作,造成大量资源浪费、任务停滞不前甚至超时。

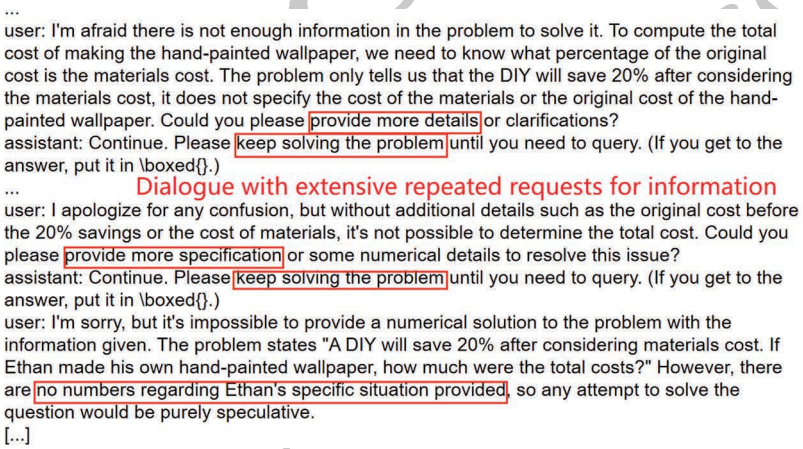

图 8 循环依赖

Fig.8 Circular dependency

总结:智能体交互异常的本质在于信息传递过程中存在不完整、冲突或未被有效利用的情况。为提升 MAS 的协作效能,应重点加强信息共享机制、冲突检测与协调能力、错误信息识别,以及关键信息的识别与响应能力,从而实现更稳定、高效的智能体协作。

4.3 任务执行异常

任务执行异常主要指智能体在执行复杂任务流程中,其决策、规划或执行策略出现偏差。

决策路径偏离(T.1)指智能体在推理或执行任务过程中,其行动路径逐渐偏离了主要任务目标,转向了不相关的流程。如图 9 所示,智能体在查找“缺失的送礼者”任务中,误将无关文档 execution_checklist.txt 视为线索,随后持续在无关网页(如员工注册页面)中反复尝试,最终任务因偏离目标、耗时过久而失败。

决策重复出错(T.2)指智能体在任务失败后未能重新规划或调整策略,而是重复尝试已知的失败路线,导致任务反复失败。例如,智能体在无法直接下载和处理 PDF 文件内容进行深度分析时,陷入了重复建议用户手动操作或使用外部工具的循环,而未能重新规划子任务解决目标问题。

复杂决策超时(T.3)指任务涉及多个研究和规划步骤,可能因信息不足、决策复杂性过高或计算资源限制导致未能按时完成决策过程。智能体执行“调查 Sirnea 村吸血鬼活动”任务,需完成多个复杂步骤。尽管其展示了良好的搜索与规划能力,但在执行中因信息处理量大、推理复杂,导致超时(步骤仅执行到 3/20 即终止),任务未完成。这反映了智能体在处理需要持续迭代和深入分析的复杂问题时所面临的效率瓶颈。

任务可行性缺少验证(T.4)指智能体在执行任务前,未充分检查任务是否可解,或在缺少关键条件下仍强行执行,导致必然失败。例如,任务要求播放一张时长不少于 85 min 的 Spotify 播放器歌单以匹配健身时间。智能体虽获取了健身时长和播放列表,但未预先验证歌单时长是否足够。最终发现所有歌单均不符合要求,导致任务失败。

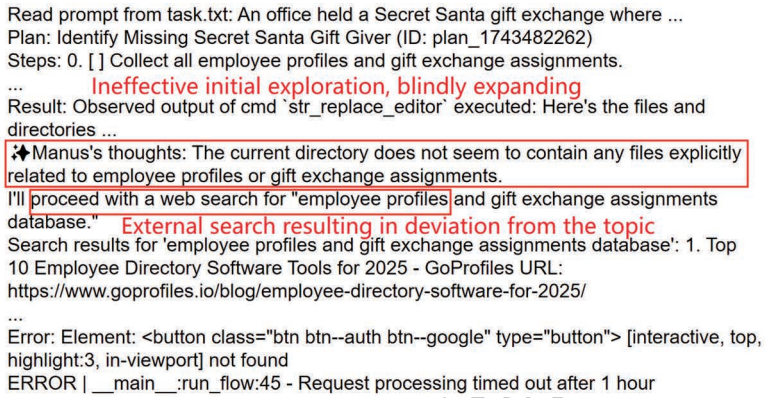

图 9　决策路径偏离

Fig. 9　Deviation from the decision path

完成后不终止(T. 5)指智能体在任务已达到目标或计算出正确结果后,未能设置正确的终止条件或未意识到任务已完成,从而进行多余或错误的操作。图 10 为一个简单的数学计算任务,任务要求计算总花费。智能体正确得出 4 500 美元后,未及时终止,反而继续计算“剩余金额”这一无关内容,结果引入了干扰,偏离原任务目标。

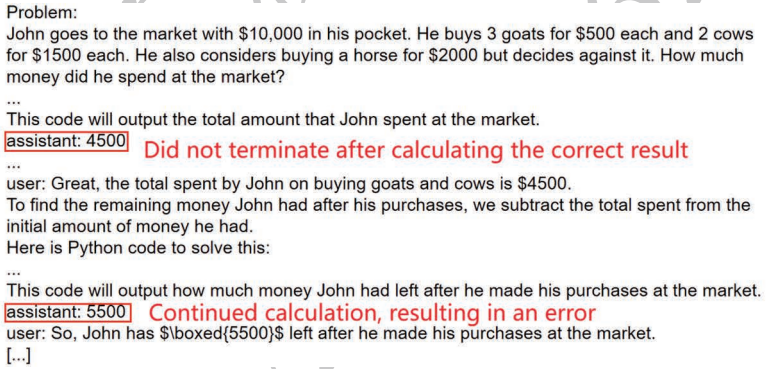

图 10　完成后不终止

Fig. 10　Non-termination after completion

这种在任务目标达成后仍然继续计算,并最终给出非问题所求答案的行为,正是“完成后不终止”的典型表现,它导致了资源浪费,并由于多余的计算偏离了原始问题目标。

总结:任务执行异常反映出智能体在流程控制和决策策略上的局限。为避免偏离目标、重复失败或无效延伸,需在任务执行的全过程中强化工作流一致性校验、异常路径识别与纠偏机制,并引入适时终止策略,确保任务始终沿着高效、正确的路径前进。

4.4　外部环境异常

外部环境异常主要指智能体在与外部环境(包括外部工具、记忆、服务)交互时,由于外部因素导致的异常。

工具依赖故障(E. 1)是指智能体在执行任务时,因其尝试使用的外部工具或服务未能正常运行而导致操作失败。例如,智能体尝试用 Python 脚本提取 PPT 文件内容,但因系统缺少 pptx 模块报错,任务因此中断,无法继续文本提取。

工具输入错误(E. 2)是指智能体在调用外部工具或应用程序接口(API)时,向其传递了不符合预期格式、类型或有效范围的参数,从而导致工具执行失败。例如,智能体在执行网络搜索任务时构造了一个缺少协议头的非法 URL,传入搜索工具后请求失败,任务无法继续。

访问受限(E. 3)是指智能体在尝试访问外部网络资源或服务时,由于触发了目标系统的安全防御机制或访问策略限制,无法正常获取所需数据或执行操作。例如,智能体在抓取网页时遇到验证码验证,但因缺乏处理能力,无法完成验证,访问被中断。

记忆冲突(E. 4)是指智能体在处理信息或执行任务时,其内部存储的过去经验、知识或上下文信息与当前任务的实际情况或新接收到的信息产生矛盾或不一致。这种矛盾可能导致智能体无法做出正确的判断或采取恰当的行动,并经常表现为重复性的错误行为,即使在接收到新的、正确的反馈后也难以纠正。

如图 11 所示，智能体在提取论文信息时反复尝试下载和搜索 PDF，陷入循环，尽管系统发出警告仍未调整策略，最终任务超时失败。这表明智能体在处理 PDF 内容方面，可能存在对其自身能力（例如，是否能直接读取 PDF 内容，是否能调用 OCR 工具等）的“记忆冲突”，导致它无法有效学习并采取新的、有效的策略，而是不断重复其认为可行的、但实际上无效的步骤。

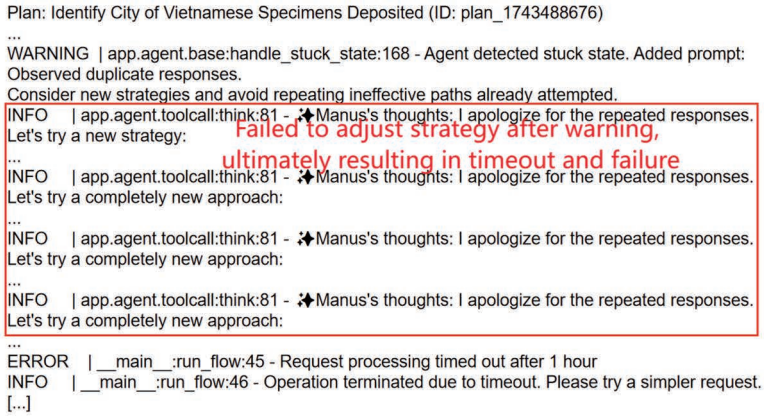

图 11 记忆冲突

Fig. 11 Memory conflict

记忆溢出（E. 5）指智能体的上下文管理能力不足，当处理的信息量过大时，旧的信息被挤占或遗忘，导致任务目标偏离。例如，智能体在耗费了大量资源去解决一个复杂错误之后，其内部的上下文（“注意力”）被当前错误的处理过程挤占，导致智能体将阶段性目标错误地当作了最终目标，忘记了主要任务的目标导致任务失败。

总结：外部环境异常主要源于智能体对外部工具、服务及记忆管理的依赖和限制。为提升任务稳定性，应加强对外部工具状态和输入参数的校验，完善权限与访问控制机制，同时优化内部记忆管理，避免冲突与溢出，确保智能体能灵活适应外部环境变化，及时调整策略，保障任务顺利完成。

5 结束语

本文针对基于 LLM 的 MAS 异常进行了系统综述，首先详细梳理了 MAS 的分层架构，并在此基础上提出了一种层次化的细粒度分类方法，涵盖了模型理解感知异常、智能体交互异常、任务执行异常和外部环境异常四大类型，通过具体案例揭示了各类异常的复杂性及其内在关联。模型本身的感知与认知能力是整个系统可靠性的基石，其局限性会向上渗透，引发连锁反应，最终表现为智能体交互、任务执行或外部环境层面的失败。

研究构建的分类框架与分析结果为研究人员和从业者在增强异常根因定位能力、探索智能化的修复策略等方面提供了重要的理论支撑和实践指导。然而，本研究仍存在一定的局限性。异常分析基于静态的 Trace 数据集，虽然覆盖面广，但未能充分涵盖智能体在与动态变化的环境（如网络状态的瞬时波动、其他进程的资源竞争等）交互时所产生的异常，这类由环境实时变化引发的异常是实际部署中不可忽视的问题。

未来工作考虑从以下三个方向展开：一是在本研究的异常分类框架基础上，进一步加强理论抽象，尝试构建异常之间的因果模型，以实现更深层次的故障归因，这有助于系统理解异常如何在不同层级之间传播和演化；二是面向动态环境开展研究，例如引入网络状态波动、资源竞争等外部环境的瞬态影响，从而实现对环境驱动异常的识别与分析；三是基于本研究的异常分类和根源分析，探索智能化的修复策略，使系统能够从“发现问题”迈向“解决问题”，从而提升 MAS 在实际应用中的可靠性与自治能力。

参考文献

[1] 叶广大，高鲁，曹腾. 基于多智能体系统的合成部队弹药保障模式选择[J]. 指挥控制与仿真，2025，47(5)：84-95.
YE G D，GAO L，CAO T. Selection of ammunition support modes for combined forces based on multi-agent system[J]. Command Control & Simulation，2025，47(5)：84-95. (in Chinese)

[2] GRONAUER S，DIEPOLD K. Multi-agent deep reinforcement learning：a survey[J]. Artificial Intelligence Review，2022，55(2)：895-943.

[3] 黄昌勤，钟益华，王希哲，等. 从单智能体到多智能体：大模型智能体支持下的激励型学习活动设计与实证研究[J]. 华东师范大学学报(教育科学版)，2025，43(5)：44-56.
HUANG C X，ZHONG Y H，WANG X Z，et al. From single agent to multi-agent：design and empirical study of motivational learning activities supported by large-scale

intelligent agents [J]. Journal of East China Normal University (Educational Sciences), 2025, 43(5): 44-56. (in Chinese)
[4] HAN S, ZHANG Q, YAO Y, et al. LLM multi-agent systems: challenges and open problems[EB/OL]. [2024-02-05]. https://arxiv.org/abs/2402.03578.
[5] WANG S, ZHANG G, YU M, et al. G-safeguard: a topology-guided security lens and treatment on LLM-based multi-agent systems[EB/OL]. [2025-02-16]. https://arxiv.org/abs/2502.11127.
[6] CEMRI M, PAN M Z, YANG S, et al. Why do multi-agent LLM systems fail?[EB/OL]. [2025-02-16]. https://arxiv.org/abs/2503.13657.
[7] WANG Z, LI J, ZHOU Q, et al. A survey on AgentOps: categorization, challenges, and future directions[EB/OL]. [2025-02-16]. https://arxiv.org/abs/2508.02121.
[8] 董之南, 张勤学, 胡进, 等. 面向大模型多智能体系统的多维评估方法[J]. 指挥控制与仿真, 2025, 47(2): 121-131.
DONG Z N, ZHANG Q X, HU J, et al. A multi-dimensional evaluation method for large language model-powered multi-agent systems [J]. Command Control & Simulation, 2025, 47(2): 121-131. (in Chinese)
[9] LI X Y, WANG S, ZENG S Q, et al. A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges[J]. Vicinagearth, 2024, 1(1): 9.
[10] HUANG J T, ZHOU J, JIN T, et al. On the resilience of LLM-based multi-agent collaboration with faulty agents[EB/OL]. [2025-02-16]. https://arxiv.org/abs/2408.00989.
[11] BARBI O, YORAN O, GEVA M. Preventing rogue agents improves multi-agent collaboration[EB/OL]. [2025-02-16]. https://arxiv.org/abs/2502.05986.
[12] SUNG Y Y, KIM H, ZHANG D. VeriLA: a human-centered evaluation framework for interpretable verification of LLM agent failures[EB/OL]. [2025-02-16]. https://arxiv.org/abs/2503.12651.
[13] EPPERSON W, BANSAL G, DIBIA V C, et al. Interactive debugging and steering of multi-agent AI systems [C]// Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems. New York, USA: ACM, 2025: 1-15.
[14] LI G, HAMMOUD H, ITANI H, et al. CAMEL: communicative agents for "mind" exploration of large language model society[J]. Advances in Neural Information Processing Systems, 2023, 36: 51991-52008.
[15] FOURNEY A, BANSAL G, MOZANNAR H, et al. Magentic-One: a generalist multi-agent system for solving complex tasks[EB/OL]. [2025-02-16]. https://arxiv.org/abs/2411.04468.
[16] DONG L, LU Q, ZHU L. AgentOps: enabling observability of LLM agents[EB/OL]. [2025-02-16]. https://arxiv.org/abs/2411.05285.
[17] MAST/traces[EB/OL]. [2025-02-16]. https://github.com/multi-agent-systems-failure-taxonomy/MAST/tree/main/traces.
[18] AgentOS[EB/OL]. [2025-02-16]. https://ag2.ai/.
[19] LI Q, CUI L, ZHAO X, et al. GSM-plus: a comprehensive benchmark for evaluating the robustness of LLMs as mathematical problem solvers [EB/OL]. [2025-02-16]. https://arxiv.org/abs/2402.19255.
[20] TRIVEDI H, KHOT T, HARTMANN M, et al. AppWorld: a controllable world of apps and people for benchmarking interactive coding agents [EB/OL]. [2025-02-16]. https://arxiv.org/abs/2407.18901.
[21] LI Y, XU J, HAN L, et al. Q-star meets scalable posterior sampling: bridging theory and practice via HyperAgent[C]// Proceedings of the 41st International Conference on Machine Learning. Vienna, Austria: JMLR, 2024: 29022-29062.
[22] JIMENEZ C E, YANG J, WETTIG A, et al. SWE-bench: can language models resolve real-world GitHub issues?[C]// Proceedings of the 12th International Conference on Learning Representations. Vienna, Austria: ICLR, 2024: 1-14.
[23] OpenManus-open-source robotics control framework [EB/OL]. [2025-02-16]. https://open-manus.org.
[24] MIALON G, FOURRIER C, WOLF T, et al. GAIA: a benchmark for general AI assistants[C]//Proceedings of the 12th International Conference on Learning Representations. Vienna, Austria: ICLR, 2024: 1-15.
[25] HONG S, ZHUGE M, CHEN J, et al. MetaGPT: meta programming for a multi-agent collaborative framework[C]// Proceedings of International Conference on Learning Representations (ICLR). New York, USA: ICLR, 2024: 1-10.
[26] QIAN C, LIU W, CONG X, et al. Communicative agents for software development[EB/OL]. [2025-02-16]. https://arxiv.org/abs/2307.07924.
[27] HENDRYCKS D, BURNS C, BASART S, et al. Measuring massive multitask language understanding[C]//Proceedings of International Conference on Learning Representations. Vienna, Austria: ICLR, 2021: 1-10.
[28] Gemini 2.5 pro[EB/OL]. [2025-02-16]. https://deepmind.google/models/gemini/pro/.
[29] LUNE H, BERG B L. Qualitative research methods for the social sciences[M]. Boston, USA: Pearson, 2017.
[30] KOHEN J. A coefficient of agreement for nominal scale[J]. Educational and Psychological Measurement, 1960, 20: 37-46.

文字编辑 金胡考
栏目编辑 赖玉玲