

多智能体协作 workflow 综述

夏冰^{1,2}, 胡梦良¹, 袁天¹, 王永³, 赵迎利³, 吴厚坡³

(1. 中原工学院网络空间安全学院; 2. 河南省计算智能和智能系统重点实验室, 河南 郑州 450007;
3. 郑州埃文科技有限公司, 河南 郑州 450000)

摘要: 运行在多智能体系统中的 workflow, 将多个智能体能力、组织管理机制与通信协议组合成可执行的子任务序列, 对协作任务完成、生成结果、工作效率、管理模式产生不可忽视的影响。介绍多智能体 workflow 概念、workflow 分类、组成和生命周期, 总结对比典型多智能体 workflow 生成技术、workflow 优化技术、workflow 安全保障技术。研究指出, 当前多智能体 workflow 生成技术的核心是任务解耦, workflow 优化重点在于降低资源消耗和提升生成结果质量, 需增加智能体自身安全、引入安全通信协议以保障智能体 workflow 安全。

关键词: 多智能体系统; workflow 生成; workflow 优化; workflow 安全

DOI: 10.11907/rjdk.261001

扫描二维码阅读全文: 

中图分类号: TP181

文献标识码: A

文章编号: 1672-7800(2026)003-0001-08

A Review of Multi-Agent Collaborative Workflow

XIA Bing^{1,2}, HU Mengliang¹, YUAN Tian¹, WANG Yong³, ZHAO Yingli³, WU Houpo³

(1. School of Cyberspace Security, Zhongyuan University of Technology; 2. Henan Key Laboratory of Computational Intelligence and Intelligent Systems, Zhengzhou 450007, China; 3. Aiven Technology of Zhengzhou Co., Ltd, Zhengzhou 450000, China)

Abstract: Workflows running in multi-agent systems combine the capabilities of multiple agents, organizational management mechanisms, and communication protocols into executable sub-task sequences, exerting a significant influence on the completion of collaborative tasks, the generation of results, work efficiency, and management models. This paper introduces the concept of multi-agent workflows, presents the classification, composition, and life cycle of workflows, and summarizes and compares typical multi-agent workflow generation technologies, workflow optimization technologies, and workflow security guarantee technologies. The research indicates that the core of current multi-agent workflow generation technologies lies in task decoupling, while the focus of workflow optimization is on reducing resource consumption and improving the quality of generated results. It is necessary to enhance the security of agents themselves and introduce secure communication protocols to ensure the security of multi-agent workflows.

Key Words: multi-agent systems; workflow generation; workflow optimization; workflow security

0 引言

大语言模型 (Large Language Models, LLMs) 已成为通用人工智能领域的核心驱动力, 深刻影响全球产业发展。从单一的参数规模扩张, 转向理解、推理与跨模态能力的全面突破^[1], LLMs 所呈现的任务理解、规划和生成能力, 加速了 AI 智能体的落地发展, 这些具有自主思考与执行

能力的智能体在教育教学^[2]、医疗辅助^[3]和经济金融^[4]等任务中展现出显著的效能, 成为实现通用人工智能的关键路径。然而, 随着应用场景的不断拓展, 当面对现实世界中具有复杂性、动态性和不确定性的任务时, 单智能体的局限性日益凸显。单个智能体处理需要长思维链推理、多工具协同调用以及跨领域知识融合的复杂逻辑判断任务时, 受限于固定的上下文窗口大小, 导致关键信息在长期的交互中丢失^[5]。在应对多目标优化问题时, 单智能体

收稿日期: 2025-12-31

基金项目: 河南省科技攻关项目 (252102210217); 河南省重点研发专项项目 (251111212000)

作者简介: 夏冰 (1981-), 男, 博士, CCF 杰出会员, 中原工学院网络空间安全学院副教授、硕士生导师, 研究方向为网络安全、智能体与安全; 胡梦良 (2000-), 男, 中原工学院网络空间安全学院硕士研究生, 研究方向为智能体与安全; 袁天 (2000-), 男, 中原工学院网络空间安全学院硕士研究生, 研究方向为智能体与安全; 王永 (1981-), 男, 博士, 郑州埃文科技有限公司副高级工程师, 研究方向为多模态大模型性能优化与多智能体协同。本文通讯作者: 夏冰。

容易出现性能瓶颈。

为突破单智能体在复杂任务求解中的性能瓶颈,以及其过程稳健性与功能可扩展性局限,学术界与工业界开始将目光投向多智能体系统(Multi Agent Systems, MAS)的研究与应用。多智能体系统集成多个功能任务分工明确的智能体,通过任务解耦将高度复杂的任务分解为若干逻辑关联的子任务,这些子任务由具备差异化角色设定、专业知识及工具权限的异构智能体协同完成。这种基于角色分工与并行驱动的机制,不仅能发挥异构能力的互补优势,还通过智能体间多维信息交互与反馈校验,让多智能体系统能处理更加动态、复杂的任务,进而显著提升了系统在复杂任务推理过程中的性能^[6]。在多智能体系统中,智能体不再是孤立的执行单元,而是集感知、决策、行动、通信于一体的自治实体,在共享的上下文环境中通过 workflow 协作达成目标。这个运行在多智能体系统中的 workflow,将分散的智能体能力、组织管理机制与通信协议组合成可执行的任务序列,充当任务执行的桥梁,对多智能体系统的任务生成结果、工作效率、管理模式产生不可忽视的影响^[7]。

目前,针对多智能体系统的研究已取得丰硕成果,但其结构形式多样、缺乏统一规范。不同多智能体系统从智能体设计到编排的 workflow 相差很大,呈现出高度多样化的特征,导致研究者对不同技术路径的优劣以及适用场景存在一定认知局限。为此,本文系统梳理了当前典型的多智能体协作 workflow 体系,介绍了其概念、组成、生命周期,从 workflow 的结构、适应性、生成方式 3 个维度对 workflow 进行分类。并且,在此基础上对主流 workflow、优化技术和安全保障技术进行对比分析,为相关研究者提供参考。

1 多智能体 workflow 概述

1.1 概念

多智能体 workflow (Multi Agent Workflow, MAW) 是一个由多个基于大语言模型的自治智能体组成的协作系统,这些智能体通过预定义或动态生成的流程以共同完成任务。单智能体 workflow 聚焦内部推理,依赖长上下文、思维链和大语言模型能力,其独立完成整个任务,集感知、推理、决策、行动于一体。单智能体 workflow 的优势在于实现简单、资源消耗低、响应速度快、易部署。单智能体 workflow 适合简单问答、短文本生成或简单工具调用等轻量场景。多智能体 workflow 强调智能体间分工和交互,形成“团队”效应,如对话讨论、任务分配和优化反馈。多智能体系统中每个智能体通常具备固定角色和专业化能力,通过编排执行序列或任务解耦,将复杂任务分解为逻辑关联的执行步骤,在 workflow 牵引下由不同智能体并行或顺序负责执行,使得系统能够有效应对高度动态、不确定的复杂任务场景,如软件开发、金融分析、社会模拟或多模态推理等。尽管当

前多智能体 workflow 在通信冗余和协调复杂性上会带来更高的 Token 消耗和延迟,但其通过角色分工、并行执行与反馈优化突破了单智能体的性能瓶颈,反映了多智能体间的分工协作将催生生产力的跃升趋势。

1.2 分类

多智能体 workflow 可从组织结构、适应性、生成方式 3 个维度进行分类。

依据组织结构,多智能体 workflow 可分成层级式 workflow、对等式 workflow 和混合式 workflow。层级式 workflow 中,智能体依据角色可以划分不同层级,存在明确的管理者和执行者。管理者负责任务分解、协调与监控,执行者完成用户具体任务或子任务,如图 1 所示。对等式 workflow 中智能体地位平等、分布式交互,无中心节点,如图 2 所示;混合式 workflow 则同时包含层级式与对等式机制,以适应更复杂场景。

基于执行过程中的可适应性,多智能体 workflow 可分成静态 workflow 与动态 workflow。静态 workflow 在规划阶段就确定了节点拓扑、执行顺序与控制逻辑,执行阶段结构不再改变;动态 workflow 根据模型推理、错误检测、外部信息等实时状态,可在执行过程中调整结构,其最终执行路径难以预先确定。

依据生成方式,多智能体 workflow 可分成预定义结构 workflow 与自动生成 workflow。预定义结构 workflow 指在执行前通过标准操作流程(Standard Operating Procedure, SOP)或模板明确定义完整 workflow。自动生成 workflow 指在 workflow 执行前,无需预定义其 workflow 结构,系统在运行时通过用户任务、算法、大模型推理等自主产生 workflow 结构。

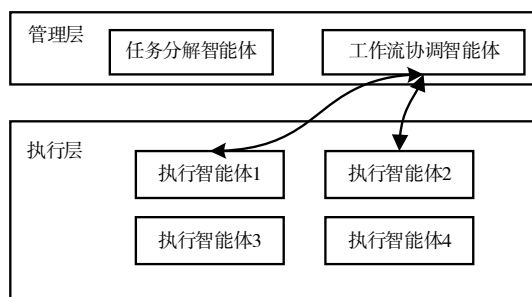


Fig. 1 Hierarchical multi-agent workflow

图 1 层级式多智能体 workflow

1.3 组成

(1) 智能体。它是构成多智能体 workflow 的核心基本单元,被封装为高度自治的实体,每个智能体都包含大脑、感知、行动三大基础部分^[8]。在多智能体 workflow 中,智能体可设置不同的智能体角色以实现分工。

(2) 依赖关系。它是指智能体间的逻辑关系和条件约束,可建模为图结构、顺序结构、会话群聊等。其中,图结构包括 DAG 图形式、自然语言描述建模的流程图形式等;顺序结构为子节点序列依次顺序执行的结构;会话群聊表示子节点间无明确逻辑依赖关系,多智能体间可自主进行

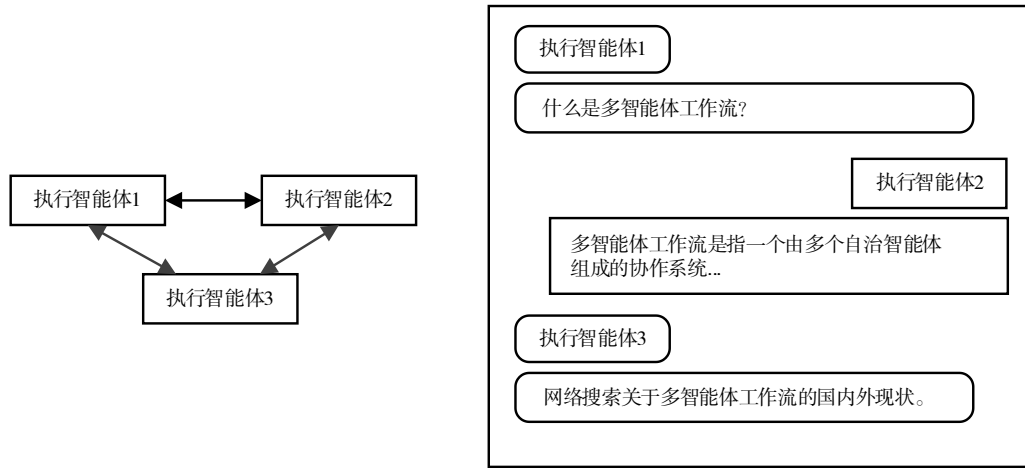


Fig. 2 Peer-to-peer multi-agent workflow

图 2 对等式多智能体工作流

会话交互。

(3)协作机制。包括智能体调度和管理、优化迭代、安全防护等机制。智能体调度和管理机制通过统一或分布式的控制策略确保工作流中智能体协同工作,以保证工作流按预期高效推进。优化迭代机制通过系统对智能体行为、通信方式或工作流结构等进行持续调整和改进,实现多智能体工作流性能的持续提升。安全防护机制指通过安全防护手段保障了工作流的运行安全和可靠性。

(4)工具。它是多智能体工作流中智能体扩展能力的外部接口,允许智能体调用 API、数据库或第三方服务以执行具体操作。工具支持包括代码执行环境如 Python 解释器、数据检索模块如 RAG 知识库以及领域特定插件如金融分析工具或医疗诊断 API 等。

(5)协议。多智能体工作流对协议的支持,如 Anthropic 推出的模型上下文协议 (Model Context Protocol, MCP),旨在标准化 LLMs 与外部数据源、工具及系统的双向连接。其核心采用客户端—服务器架构,以 AI 智能体或应用作为 MCP 客户端,通过 JSON-RPC 等机制与 MCP 服务器通信,实现工具自动发现、上下文动态注入、权限控制、实时数据交互与工具调用。在多智能体系统中, MCP 支持跨智能体高效共享外部知识,并处理跨模态数据的整合。相比传统碎片化集成, MCP 极大提升了智能体的工具使用能力、可扩展性与安全性^[9]。Google 推出的 A2A 协议旨在为异构 AI 智能体提供标准化互操作“通用语言”,解决跨厂商、跨框架的多智能体协作难题。它基于 HTTP、JSON-RPC 和 SSE 构建,支持智能体动态发现、任务委托、实时状态推送、异步协作与可追溯交互^[10]。

1.4 生命周期

多智能体工作流的生命周期通常包括规划过程和执行过程,执行期间往往还支持反馈过程和人类干预过程,如图 3 所示。

1.4.1 规划过程

规划指多智能体工作流在任务执行阶段前的思考分

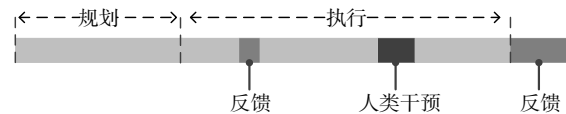


Fig. 3 The life cycle of a workflow

图 3 工作流的生命周期

解阶段。通常涉及任务分解和角色分配,由规划智能体将整体目标任务拆分为多个彼此存在联系的子任务,并分配给合适的智能体。例如,在 AutoGen 框架中,规划通过定义智能体角色并编程交互行为加以实现^[11]。在一个股票价格绘图的任务中,规划阶段包括智能体讨论安装包的需求,然后分配执行步骤。同样,在 MetaGPT 中,规划由 Product Manager 智能体生成结构化的产品需求文件^[12]。在 OWL 框架中,由任务规划智能体通过分解工具对整体目标任务进行拆分,并为每个子任务分配唯一 ID,同时记录其依赖关系^[13]。具备依赖关系的子任务形成一种工作流结构。

1.4.2 执行过程

执行指多智能体工作流执行任务阶段。智能体根据规划生成的工作流操作子任务,可能涉及工具调用或外部交互,并最终返回结果。例如,在 AutoGen 中,执行包括 Writer 智能体生成代码、Safeguard 智能体检查安全等^[11]。MetaGPT 的执行则通过 Engineer 智能体编写代码,并在共享消息池中发布结构化输出^[12]。在 OWL 框架中,协调智能体通过协调 Workers 工作以完成任务^[13]。协调智能体从队列中提取可执行子任务,并将其分配给智能体 Workers 执行,Workers 通过自身能力完成子任务并返回任务状态。

1.4.3 反馈过程

在工作流执行中或结束后,通过一定方式对工作流进行改进的策略称为反馈阶段。结合规划结果、执行状态或执行结果等评估数据,多智能体系统通过反馈机制纠正错误或迭代优化。反馈阶段通常位于多智能体工作流执行中断阶段或执行阶段后。在 Aflow 中,优化器通过蒙特卡

洛树算法对工作流进行迭代优化^[14]。在 OWL 中,当任务处理失败后,协调智能体根据失败恢复策略进行重试、重规划、重分解、创建新节点^[13]。在 PEER 中,系统设计 Review 智能体,通过判断结果是否满足用户要求,对计划、执行阶段进行修改以实现反馈和迭代优化^[15]。

1.4.4 人类干预过程

人类用户可以实时参与,提供反馈或纠正,以增强多智能体系统的可控性和可靠性。该机制允许人类在关键点介入工作流,如批准代码执行或解决歧义。在 AutoGen 中,通过 UserProxyAgent 智能体,用户可以设置为每次智能体生成响应前都暂停等待人类输入、仅在任务可能结束时征求确认或全自动模式^[11]。在 LangGraph 中,开发者在图结构中设置中断点 breakpoints,执行到指定节点时自动暂停并等待人类输入、审查或修改状态后恢复运行,支持细粒度控制和安全审批,确保复杂工作流中的人工可控性^[16]。

2 多智能体工作流研究现状

2.1 多智能体工作流生成技术

本文选取若干主流多智能体工作流框架进行对比分析,包括 CAMEL^[17]、AutoGen^[11]、MetaGPT^[12]、AgentVerse^[18]、OWL^[13]、MegaAgent^[19]、EvoAgentX^[20]、LangGraph^[16]、Coze^[21]、CrewAI^[22]、Agno、Google ADK。这些框架覆盖了从早期对话式协作到自动化生成、生产级编排的技术发展轨迹,代表了当前多智能体工作流的主要设计范式。

为便于读者分析对比,本文从系统名称、智能体角色、工作流分类、工作流依赖关系、应用场景、人类干预、反馈机制、工具支持、实验等维度,全面对比了当前主流多智能体协作工作流框架,具体如表 1 所示。其中,“/”表示未指明。表 2 对多智能体工作流分类的编码进行了说明,分类的编码组合规则为 3 个维度依次拼接,例如“H-D-Pre”表示层级式+动态+预定义结构的工作流。

CAMEL 通过 Inception Prompting 为智能体分配角色,如用户与助手,然后引导其自主生成任务细节并进行多轮对话,直至自然完成任务^[17]。该框架强调涌现行为与简单性,无需预定义复杂结构,仅依赖角色提示激发 LLM 社会协作。其实现门槛低,但缺乏结构化控制、易产生幻觉与任务偏离。

AutoGen 提供了高度灵活的对话式多智能体框架,支持开发者通过代码或自然语言定义多种交互模式,如两人对话、群聊。核心流程为:UserProxyAgent 发起任务,其他智能体通过消息传递迭代响应,支持代码执行与工具调用。该框架内置 human-in-the-loop 与反馈循环^[11]。该方案异步执行高效、易于人为干预。

MetaGPT 模拟软件公司团队,采用严格预定义 SOP 驱

动工作流,Product Manager 输出需求文档、Architect 设计架构、Project Manager 任务分配、Engineer 实现代码、QA Engineer 测试验证。各角色输出标准化中间文档,形成强顺序依赖^[12]。该方案在端到端软件开发中表现卓越,代码质量高、可重复性强,但灵活性低、难以扩展至其他类型任务。

AgentVerse 引入“生成—决策—执行—评估”的模块化循环,支持动态生成专家智能体并选择水平/垂直通信结构,实现涌现社会行为与多轮迭代优化。适用于文本推理、工具使用与具身智能任务^[18]。该方案在复杂任务中表现出色,但组队和生成质量的不确定性会增加执行轮次,导致时间和资源开销增大。

OWL 提出分层多智能体框架,通过将规划与执行彻底解耦,实现高效跨领域适应。其核心流程为:Planner 智能体递归分解任务、Coordinator 智能体管理分配、Worker 智能体调用领域工具执行,支持反馈迭代。训练阶段仅优化 Planner 的强化学习,显著提升了其泛化能力^[13]。这种方案使系统面临新领域时只需添加 Worker 即可适配,模块化易扩展,但多次面对相同复杂任务时,每次生成质量波动较大。

MegaAgent 实现了无需预定义 SOP 的大规模自主协作,Admin Agent 根据任务复杂度动态分解、生成工作流。同时,面对复杂任务时可自主生成子智能体并组建团队,支持并行执行和团队嵌套,支持监控反馈与错误重试^[19]。该框架在游戏开发、政策模拟等复杂任务中展现出色的扩展性与效率,但大规模的生成往往导致资源消耗较大和幻觉增加。

EvoAgentX 提供自动化工作流生成与进化框架,从任务提示自动构建多智能体结构,并集成 TextGrad、AFlow、MIPRO 等算法迭代优化提示、工具与拓扑^[20]。其适用于多跳推理、代码生成与真实世界任务,显著提升了现有智能体性能。但同时,其计算开销大、优化收敛依赖评估质量。

LangGraph 采用基于图的编排范式,开发者手动构建状态图,支持循环、分支、持久状态与中断恢复^[16]。该方案有极高的灵活性与控制力,无缝集成 LangChain 生态,因此被广泛应用,适用于生产级复杂工作流与人机混合流程。

Coze 作为可视化低代码平台,提供拖拽式工作流编辑器与多智能体模式,支持节点连接、插件集成与 RAG 知识库。适用于聊天机器人、企业自动化与快速原型开发。学习门槛低、插件生态丰富、调试直观。但开源版功能受限,复杂大规模多智能体协调需手动优化,性能依赖底层 LLM。

CrewAI 允许用户先通过 YAML 文件或代码定义智能体角色、工具和任务目标。启动后,管理者智能体将复杂任务分解为子任务,再根据智能体的角色适配性分配任务,同时支持人类干预和系统反馈。若出现异常,系统会自动协调执行逻辑和工具使用策略^[22]。CrewAI 能满足简易、快速构建多智能体自动化的需求,但大规模智能体集群调度能力略显不足。

Table 1 Typical workflow comparative analysis
表 1 典型工作流对比分析

系统名称	智能体角色	分类	依赖关系	应用场景	人类干预	反馈	工具支持	实验
PEER	Plan、Execute、Express、Review	H-D-Pre	顺序循环	金融问答领域	/	是	是	FinQA 数据集
AgentVerse	Expert Agent、Evaluator、etc	M-D-Pre	图结构	文本推理、编码、社会模拟	/	是	是	文本理解与推理任务、Humaneval
AutoGen	ConversableAgent、AssistantAgent、UserProxyAgent	H-S-Pre	会话群聊	复杂任务解决、代码生成、问答	是	是	是	HumanEval、MBPP、GSM8K 等
CAMEL	User Agent、Assistant Agent、自定义角色	P-S-Auto	会话群聊	社会模拟、复杂推理	/	是	是	代码生成、对话式任务完成、HumanEval
LangGraph	agent Node	/	图结构	生产级 LLM 应用	是	是	是	/
EVOagentX	动态生成的自定义智能体	P-D-Auto	图结构	多跳推理、代码生成、数学问题求解	是	是	是	HotPotQA、MBPP、MATH、GAIA
OWL	Planner、Coordinator、worker	H-D-Auto	图结构	真实场景任务、网页浏览	是	是	是	GAIA、AssistantBench、WebArena
MetaGPT	Product Manager、Architect、Project Manager 等	H-S-Pre	顺序结构	软件开发、项目管理	是	是	是	HumanEval、MBPP;SoftwareDev(自定义软件开发任务集)
MegaAgent	Boss Agent、Admin Agent、ordinary agents	H-D-Auto	图结构	复杂协调任务、游戏开发、世界模拟	/	是	是	MBPP、HumanEval、MATH、GSM8K;
Coze	自定义节点	P-S-Pre	图结构	聊天机器人开发、客服自动化、生产力助手等现实应用场景	是	是	是	/
crewAI	Manager Agent、Worker Agent、Delegator Agent	H-D-Auto	图结构	医疗辅助诊断、客户服务咨询、软件开发等	是	是	是	HumanEval、TeamFormationBench、SoftwareDev
Agno	Research Agent、Analysis Agent、Final Agent、WorkflowAgent	M-D-Pre	图结构	金融问答、多模态任务、企业级多智能体系统部署、	/	是	是	工具测试、多模态测试类型实验
Google ADK	Execution Manager Agent、Plan Execution Agent 等	H-D-Pre	图结构	代码 / 数学任务、大规模企业智能体调度	是	是	是	BigQuery Agent、GEPA 等

Table 2 Description of multi-agent workflow classification coding
表 2 多智能体工作流分类的编码说明

维度	含义	编码
组织结构	层级式	H
	对等式	P
	混合式	M
执行过程适应性	静态	S
	动态	D
生成方式	预定义结构	Pre
	自动生成	Auto

Agno 核心围绕智能体构建、工作流编排、监控迭代环节展开。工作流以预定义为核心模式,执行前需定义完整工作流规则,包括智能体角色、执行顺序、交互规则、控制逻辑。该框架集成 RAG 知识库、多模态工具调用。其工作流能力丰富、稳定性强,适用于文档问答、知识分析、多模态推理等复杂场景,但难以适配高度动态的开放任务场景。

Google ADK (Agent Development Kit) 是谷歌推出的生产级多智能体框架,旨在简化从简单任务到复杂多 Agent 系统的构建、评估与部署全流程。该框架内置开发界面,集成 Google 生态工具及自定义工具,通过预定义的工作流程和动态工作流调整完成任务,降低了复杂多智能体系统的研

发门槛,可快速构建企业级智能体应用。但其工作流多依赖预定义结构,自动生成能力相对有限,限制了其灵活性。

综上可知,多智能体工作流生成技术的核心优势在于自主性,标志着从传统静态脚本向动态角色协作的进步。CAMEL 和 AutoGen 奠定对话式交互基础,通过简单的角色提示和异步消息传递,实现多智能体间的自然涌现协作,无需复杂预定义结构即可激发智能体的角色行为。MetaGPT 和 OWL 引入标准化输出机制,前者模拟软件工程 SOP 流程以确保高质量代码生成,后者通过“规划—协调—执行”的分层解耦实现跨域适应性。进一步地,MegaAgent 和 LangGraph 推动规模化自适应,前者动态生成嵌套团队支持并行复杂任务,后者基于图状态管理提供生产级控制力。CrewAI、Agno 和 Google ADK 面向企业级多智能体协作需求构建起完备的业务流程和社区。整体而言,这些技术将智能体从孤立的单体能力跃升为高效的智能化工厂,显著提升了智能体对复杂任务的处理与容错能力。

2.2 工作流优化技术

目前主流的工作流框架仍存在生成结果质量不稳定的问题,缺乏可靠的冲突调解机制与质量门控,幻觉风险在多轮交互中被指数级放大,造成资源严重浪费,因而工

作流优化技术成为研究热点。多智能体工作流的优化方法主要分为两种思路:①在保证精度的情况下降低资源消耗和提高效率;②提升工作流生成结果的质量。

当前,多智能体工作流的资源消耗面临显著挑战,相比于单智能体系统,多智能体工作流中智能体之间在不断地进行交互与信息传递,从而导致执行时间与Token消耗的指数级增长。Wang等^[23]指出,在多智能体框架下,工作流的执行都要求参与的智能体之间持续进行信息交换,以实现协同与合作的有效性。这种信息传递贯穿任务输入至输出阶段,导致其存在明显冗余,从而系统在多轮通信与处理环节中会产生显著的Token消耗,对整体计算资源与响应时间造成负担。除通信与协同所消耗的资源外,多智能体系统在工作过程中采用的额外策略也会导致资源需求显著增加,如智能体结果生成策略、最优路径选择、资源分配策略制定等。文献[24]通过实验表明,在同一任务情境下,相较于单智能体系统,多智能体系统的总体资源消耗往往更高,且不同的任务完成路径策略会带来不同程度的资源消耗,显示出路径选择对计算与通信负荷的影响具有可观的差异性。Kim等^[25]指出,虽然单智能体的整体效率较低,但其效率损失往往较小,相对地,在资源充足、工具繁多的多智能体环境中,智能体之间的协同成本会被放大,从而增加实现任务目标所必需的消耗。Su等^[26]进一步强调,现有的多智能体工作流往往倾向于采用统一架构覆盖所有场景下的任务,尽管该方式在准确性等性能指标上表现优异,但过度依赖重型流水线及高频的LLM调用与工具链,导致在处理简单查询时存在明显的Token开销冗余与响应延迟。

多智能体工作流资源消耗优化可从多个角度入手。基于智能体的路由选择,Yue等^[27]指出多智能体系统可以选择较小的LLM,以更低的资源成本实现不低于最大LLM的性能效果,并提出多智能体动态路由算法并进行了验证。智能体之间的高效协同对工作流性能的提升至关重要,该问题的核心挑战在于,如何在多个条件约束下寻找最优或近似最优的分配策略,这需要在任务需求与优先级、智能体能力与可用资源、环境约束与动态性等维度之间实现动态平衡。然而,这些因素在不同的时间点可能存在显著差异,因此建立一套适应力强且高效的多智能体任务分配机制尤为必要^[28]。此外,有研究对工作流中的各种数据指标进行量化评估,例如智能体的工作贡献、协作效果、上下文内容等,对不必要或效果差的部分进行剪枝,从而节省消耗^[29]。

此外,工作流效率提升也是重要优化方向。在面对用户提出的问题时,静态工作流往往采用相同的流程,这导致简单的请求也会消耗大量时间成本,因此针对不同复杂度问题可以采用不同的策略,从而提高响应速度。同时,还可以结合请求在工作流中的延迟敏感度和所在阶段,引入精细化的优先级调度机制^[30]。将对整体完成时间影响

更大的关键请求置于更高优先级,以减少端到端排队时间。在此基础上,进一步结合智能体间依赖关系与资源占用特性进行全局协调,例如优先调度处于关键路径上的推理请求、延迟敏感的用户交互型请求等,使系统能够在高并发场景下仍保持稳定、可预期的响应性能。

工作流的生成结果可以通过人为判断其质量,而近些年的研究则更倾向于让工作流在工作过程中自动优化,达到不断进化的效果。AFlow通过蒙特卡罗树算法和软混合概率选择,并结合工作流生成经验持续优化工作流的结构,提升工作流和生成结果的质量^[14]。文献[31]提出让多智能体在推理与回答过程中进行辩论,减少智能体可能产生的幻觉和错误问题,提升生成结果的准确性。文献[32]在工作流的工作节点中采用多层架构,每层包含多个智能体,每个智能体在生成回答时都会将上一层所有智能体的输出作为辅助信息进行推理,从而提高了生成结果的质量。

3种工作流优化技术比较情况如表3所示。可以看出,当前工作流优化技术研究的重心是闭环自进化,通过引入类似AFlow的结构化搜索、多智能体辩论和动态路由等机制,初步实现了工作流从静态预定义执行向智能自适应编排的跃迁。不足之处在于,当工作流试图追求自动化的进化能力时,底层却过度依赖尚不成熟的强化学习算法。在结果反馈稀疏的复杂业务场景中,这些算法往往难以收敛,优化行为可能会导致性能退化与决策波动。同时,工作流缺乏跨框架标准化评估基准,不同的优化方法间难以比较优劣,导致实际部署中资源节省往往被额外调试成本抵消。

Table 3 Workflow optimization technology comparison

表3 工作流优化技术比较

技术名	技术特点	优势	相关文献
减少资源消耗	侧重于计算成本的控制与效率最大化,通过精细化分配算力、剪除冗余环节以实现降本增效	降低工作流的边际成本,帮助工作流快速实现生产应用的落地	[27]、[28]、[29]、[31]、[32]
效率提升	侧重于工作流各方面的效率提升,通过引入路由选择与精细化的优先级等机制提升响应速度等指标	减少工作流的冗余耗时,确保相应性能的稳定性和可预期性	[26]、[30]、[33]、[34]
质量优化	侧重于结果的准确性与结构的进化,通过增加交互深度、引入对抗或自我修正机制以减少幻觉和错误	增强工作流的能力上限,让工作流获得长期的自适应进化能力	[29]、[35]、[36]、[37]

2.3 工作流安全保障技术

除智能体自身存在的安全问题外,多智能体特性也暴露了广泛的攻击面,提示注入是智能体领域内最常见的攻击手段之一^[30]。针对单个智能体,危险的输入内容可能会让智能体生成非法内容或泄露隐私信息;而在多智能体系统中,攻击维度的复杂性显著增加,威胁不仅源于外部攻

击者的直接注入,还可能源于系统内部恶意智能体所产生的对抗性输出,形成链式攻击。除针对文本的攻击外,攻击者也从多智能体的协作特性入手,例如女巫攻击^[33],攻击者可以伪造合法智能体从而实现对工作流的支配,或是耗尽工作流的资源,攻击者还可以劫持工作流中的智能体实行中间人攻击^[34],通过向其他节点发送恶意内容或是拦截合法内容进行替换,从而操控工作流生成内容,导致整个工作流瘫痪。多智能体中还存在对齐问题,缺乏监管的长时间运行^[35]、任务相关知识缺口^[36]、智能体或用户偏好都可能导致工作流偏移原始目标,而主动干预与引导机制的缺失也加剧了工作流目标偏移问题^[37]。

针对多智能体工作流安全,目前业界主要包含 AI for Security、Security for AI、AI Safety 3 种方向,表 4 给出了工作流安全保障技术对比。AI for Security 的特点是利用 AI 赋能并强化安全能力,如在工作流中增加检测智能体,主动检测工作流可能出现的问题并发出预警,及时纠正智能体目标偏移^[38-40]。当智能体无法确定自己的理解与实际任务目标是否符合时,还可以求助专门配置的引导智能体进行校准。Security for AI 侧重于通过一系列安全机制增强智能体及其运行环境的防护能力,避免其在训练、推理和交互过程中成为攻击入口或被恶意操控,在多智能体系统中,除针对单个智能体的加固,更关键的是防范工作流中的通信威胁。针对多智能体系统采用的协议,如 MCP 和 A2A,需要设计匹配的安全防御机制,包括加密通信信道、消息完整性验证以及动态密钥交换机制等^[10,41-42]。AI Safety 旨在确保智能体以可靠且符合人类利益的方式运行,防止发生非故意的意外伤害。在多智能体工作流的执行过程中,需要设置防护措施,防止智能体在信息交换中产生目标偏移或价值偏移。例如,Ravindran 等^[43]提出的道德锚定系统 MAS(Moral Anchor System, MAS)框架,通过价值状态的实时贝叶斯推断、用于预测漂移趋势的 LSTM 网络,以及支持自适应干预的人本治理层,实现对智能体价值漂移的检测、预测与缓解。

目前,对智能体安全机制的研究仍聚焦于根据攻击或威胁增加相应防御,这种方式固然可以提高工作流的鲁棒性与安全性,但同时会带来额外的安全成本。并且,其对于种类繁多、层出不穷的新威胁无法做到面面俱到,遇到未预设防御的攻击时仍会造成破坏。关于工作流安全保障,未来研究方向更需放在异常的精准定位和精确回溯上,并致力于实现安全和资源消耗之间的平衡。

3 结语

本文系统梳理了当前典型的多智能体协作工作流体系,介绍了其概念、分类、组成、生命周期,对典型工作流生成技术进行了对比研究和归纳分析。并且,基于多智能体工作流所面临的挑战,从多智能体工作流的优化方法和安

Table 4 Workflow security assurance technologies comparison
表 4 工作流安全保障技术比较

技术方向	技术特点	优势	相关文献
AI for Security	利用 AI 的认知与推理能力,赋能强化工作流的安全防御	从业务逻辑层面识别出理解错误或行为失控,实时监控工作流的状态,在破坏发生前进行干预或求助,提高了系统的容错性	[38]、[39]、[40]、[44]
	针对多智能体系统		
	采用安全措施,确保在工作过程中的安全	防止个人隐私和敏感数据在系统工作过程中泄露,确保系统完整性。例如,在基础设施层面,构建安全下,仍能保持预期的行为边界,不被恶意操纵。通过严格的身份验证和密钥管理,防止恶意节点伪装接入工作流	
Security for AI	使用道德锚定系统、可解释 AI 等技术,确保智能体的目标与人类价值观一致,防止非故意的行为偏差引发灾难性后果	能够有效预防目标漂移导致的意外伤害,而非仅仅防范人为攻击。确保大规模多智能体系统在复杂交互环境下不产生群体性的逻辑失控。通过对偏见、伦理和透明度的治理,构建公众对智能体系统的长期信任	[42]、[45]、[46]、[47]、[48]、[49]、[50]
AI Safety			

全保障技术方面作了对比分析。多智能体工作流的挑战主要在于如何提升工作流执行效率和保障工作流安全。通过本文研究,以期对未来智能体开发、智能体协作、智能体互联研究提供参考价值。

参考文献:

[1] WANG L, MA C, FENG X, et al. A survey on large language model based autonomous agents [J]. *Frontiers of Computer Science*, 2024, 18 (6) : 186345.

[2] LIU Z N. Educational agents enable high school physics teaching: innovative practice and application exploration [J]. *Digital Teaching in Primary and Secondary Schools*, 2025(4) : 88-91.

刘泽宁. 教育智能体赋能高中物理教学: 创新实践与应用探索 [J]. *中小学数字化教学*, 2025(4) : 88-91.

[3] LI P F, WU J Y, ZHENG Y W, et al. A review of research and applications of large language model-driven medical agents [J]. *Chinese Journal of Health Information Management*, 2025, 22(2) : 179-186, 194.

李鹏飞, 吴静依, 郑悦闻, 等. 大语言模型驱动的医疗智能体研究与应用综述 [J]. *中国卫生信息管理杂志*, 2025, 22(2) : 179-186, 194.

[4] WU Y F. Applied research and practice of financial agents [Z]. Beijing: Huaxia Bank Co., Ltd., 2024-05-22.

吴永飞. 金融智能体应用研究与实践 [Z]. 北京: 华夏银行股份有限公司, 2024-05-22.

[5] ZHANG Y, SUN R, CHEN Y, et al. Chain of agents: large language models collaborating on long-context tasks [J]. *Advances in Neural Information Processing Systems*, 2024, 37: 132208-132237.

[6] GUO T, CHEN X, WANG Y, et al. Large language model based multi-agents: a survey of progress and challenges [DB/OL]. <https://arxiv.org/abs/2402.01680v1>.

[7] MALDONADO D, CRUZ E, TORRES J A, et al. Multi-agent systems: a survey about its components, framework and workflow [J]. *IEEE Access*, 2024, 12: 80950-80975.

[8] XI Z, CHEN W, GUO X, et al. The rise and potential of large language

- model based agents: a survey [DB/OL]. <https://arxiv.org/abs/2309.07864>.
- [9] KRISHNAN N. Advancing multi-agent systems through model context protocol: architecture, implementation, and applications [DB/OL]. <https://arxiv.org/abs/2504.21030>.
- [10] RAY P P. A Review on agent-to-agent protocol: concept, state-of-the-art, challenges and future directions [DB/OL]. <https://www.techrxiv.org/users/913189>.
- [11] WU Q, BANSAL G, ZHANG J, et al. Autogen: enabling next-gen LLM applications via multi-agent conversations [C]//Beijing: The First Conference on Language Modeling, 2024.
- [12] HONG S, ZHUGE M, CHEN J, et al. MetaGPT: meta programming for a multi-agent collaborative framework [C]//Vienna: The International Conference on Learning Representations, 2023.
- [13] HU M, ZHOU Y, FAN W, et al. Owl: optimized workforce learning for general multi-agent assistance in real-world task automation [DB/OL]. <https://arxiv.org/abs/2505.23885>.
- [14] ZHANG J, XIANG J, YU Z, et al. Aflow: automating agentic workflow generation [DB/OL]. <https://arxiv.org/abs/2410.10762>.
- [15] WANG Y, LI X, WANG B, et al. Peer: expertizing domain-specific tasks with a multi-agent framework and tuning methods [DB/OL]. <https://arxiv.org/abs/2407.06985>.
- [16] LangChain. Langgraph: stateful multi-actor workflow library for LLM applications [EB/OL]. <https://github.com/langchain-ai/langgraph>.
- [17] 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.
- [18] CHEN W, SU Y, ZUO J, et al. AgentVerse: facilitating multi-agent collaboration and exploring emergent behaviors [C]//Vienna: The International Conference on Learning Representations, 2024.
- [19] WANG Q, WANG T, TANG Z, et al. MegaAgent: a large-scale autonomous LLM-based multi-agent system without predefined SOPs [C]//The Findings of the Association for Computational Linguistics, 2025: 4998–5036.
- [20] WANG Y, LIU S, FANG J, et al. Evoagentx: an automated framework for evolving agentic workflows [C]//Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2025: 643–655.
- [21] ContributorsCoze. Coze: an open framework for building agentic applications [EB/OL]. <https://github.com/coze-dev/coze>.
- [22] ZHANG L, JI Z, CHEN B. Crew: facilitating human-ai teaming research [DB/OL]. <https://arxiv.org/abs/2408.00170>.
- [23] WANG Q, TANG Z, JIANG Z, et al. AgentTaxo: dissecting and benchmarking token distribution of LLM multi-agent systems [C]//Singapore: ICLR 2025 Workshop on Foundation Models in the Wild, 2025.
- [24] WANG H, ZHAO S, WANG J, et al. Beyond frameworks: unpacking collaboration strategies in multi-agent systems [DB/OL]. <https://arxiv.org/abs/2505.12467>.
- [25] KIM Y, GU K, PARK C, et al. Towards a science of scaling agent systems [DB/OL]. <https://arxiv.org/abs/2512.08296>.
- [26] SU J, LAN Q, XIA Y, et al. Difficulty-aware agentic orchestration for query-specific multi-agent workflows [DB/OL]. <https://arxiv.org/abs/2509.11079>.
- [27] YUE Y, ZHANG G, LIU B, et al. Masrouter: learning to route llms for multi-agent systems [DB/OL]. <https://arxiv.org/abs/2502.11133>.
- [28] FANG Z, MA T, HUANG J, et al. Efficient task allocation in multi-agent systems using reinforcement learning and genetic algorithm [J]. *Applied Science*, 2025, 15: 1905.
- [29] LIU Z, ZHANG Y, LI P, et al. A dynamic LLM-powered agent network for task-oriented agent collaboration [C]//Beijing: The First Conference on Language Modeling, 2024.
- [30] CHEN J, SHI J, CHEN Q, et al. Kairos: low-latency multi-agent serving with shared LLMs and excessive loads in the public cloud [DB/OL]. <https://arxiv.org/abs/2508.06948>.
- [31] ZHANG G, YUE Y, LI Z, et al. Cut the crap: an economical communication pipeline for llm-based multi-agent systems [DB/OL]. <https://arxiv.org/abs/2410.02506>.
- [32] YI X, ZHOU Z, CAO C, et al. From debate to equilibrium: belief-driven multi-agent LLM reasoning via bayesian nash equilibrium [DB/OL]. <https://arxiv.org/abs/2506.08292>.
- [33] CHEN W, YUAN J, QIAN C, et al. Optima: optimizing effectiveness and efficiency for llm-based multi-agent system [C]//The Findings of the Association for Computational Linguistics, 2025: 11534–11557.
- [34] AMAYUELAS A, YANG J, AGASHE S, et al. Self-resource allocation in multi-agent LLM systems [DB/OL]. <https://arxiv.org/abs/2504.02051>.
- [35] DU Y, LI S, TORRALBA A, et al. Improving factuality and reasoning in language models through multiagent debate [C]//Vienna: The Forty-first International Conference on Machine Learning, 2023.
- [36] WANG J, WANG J, ATHIWARATKUN B, et al. Mixture-of-agents enhances large language model capabilities [DB/OL]. <https://arxiv.org/abs/2406.04692>.
- [37] SONG A, AZMAN A. Leveraging symmetry in multi-agent code generation: a cross-verification collaboration protocol for competitive programming [J]. *Symmetry*, 2025, 17(10): 1660.
- [38] JIN J D, HUANG Z, HU Z Y, et al. Advanced phishing email detection method based on agent workflows [J]. *Journal of Communications*, 2024, 45(S2): 59–68.
金建栋, 黄正, 胡占宇, 等. 基于智能体工作流的高级钓鱼邮件检测方法 [J]. *通信学报*, 2024, 45(S2): 59–68.
- [39] SONG C, MA L, ZHENG J, et al. Audit-LLM: multi-agent collaboration for log-based insider threat detection [DB/OL]. <https://arxiv.org/abs/2408.08902>.
- [40] TRINH T, DANESH M H, KHANH N X, et al. Getting by goal misgeneralization with a little help from a mentor [DB/OL]. <https://arxiv.org/abs/2410.21052>.
- [41] SINGH A, EHTESHAM A, KUMAR S, et al. A survey of the model context protocol (MCP): standardizing context to enhance large language models (LLMs) [DB/OL]. <https://www.preprints.org/manuscript/202504.0245>.
- [42] KONG D, LIN S, XU Z, et al. A survey of LLM-driven ai agent communication: protocols, security risks, and defense countermeasures [DB/OL]. <https://arxiv.org/abs/2506.19676>.
- [43] RAVINDRAN S K. Moral anchor system: a predictive framework for AI value alignment and drift prevention [DB/OL]. <https://arxiv.org/abs/2510.04073>.
- [44] LIN R. Thelma: threat hunting enhanced language models for hunt automation [C]//Xiamen: 2024 IEEE Conference on Communications and Network Security, 2024.
- [45] HUANG K, NARAJALA V S, YEOH J, et al. A novel zero-trust identity framework for agentic ai: Decentralized authentication and fine-grained access control [DB/OL]. <https://arxiv.org/abs/2505.19301>.
- [46] KOOHESTANI R. Agentguard: runtime verification of AI agents [DB/OL]. <https://arxiv.org/abs/2509.23864>.
- [47] GARZON S R, VAZIRY A, KUZU E M, et al. AI agents with decentralized identifiers and verifiable credentials [DB/OL]. <https://arxiv.org/abs/2511.02841>.
- [48] WU S, SHU K. Memory in LLM-based multi-agent systems: mechanisms, challenges, and collective intelligence [EB/OL]. <https://www.researchgate.net/publication/398392208>.
- [49] JAHN F, MUSKALLA Y, DARGASZ L, et al. Breaking up with normative-monolithic agency with GRACE: a reason-based neuro-symbolic architecture for safe and ethical AI alignment [DB/OL]. <https://arxiv.org/abs/2601.10520>.
- [50] ABIRI G. Public constitutional AI [DB/OL]. <https://arxiv.org/abs/2406.16696>.