第 51 卷 第 11 期 自 动 化 学 报 Vol. 51, No. 11
2025 年 11 月 ACTA AUTOMATICA SINICA November, 2025

大语言模型的工具使用综述

郑逸宁[1] 余 镇[1] 李不凡[1] 杨 捷[1] 殷林琪[1] 印张悦[1] 袁枫烨[1]
魏海洋[1] 陆嘉昊[1] 方世成[1] 陈 爽[1] 邱锡鹏[1]

摘 要 大语言模型因其强大的生成和理解能力受到广泛关注, 但在获取实时信息和执行复杂计算上仍存在局限性. 为使其更好地响应用户需求, 赋予大语言模型工具使用能力成为当下的研究热点. 首先, 明确大语言模型工具使用的基本概念, 并按照时间顺序梳理工具使用的发展脉络. 随后, 总结与工具使用相关的数据集和技术方法, 并分析其在智能体和具身智能等领域的应用. 最后, 梳理大语言模型工具使用领域未来的研究重点与发展方向.

关键词 大语言模型, 工具使用, 工具增强生成, 智能体, 具身智能

引用格式 郑逸宁, 余镇, 李不凡, 杨捷, 殷林琪, 印张悦, 袁枫烨, 魏海洋, 陆嘉昊, 方世成, 陈爽, 邱锡鹏. 大语言模型的工具使用综述. 自动化学报, 2025, **51**(11): 2371−2386

DOI 10.16383/j.aas.c240793 **CSTR** 32138.14.j.aas.c240793

Survey of Tool Use in Large Language Models

ZHENG Yi-Ning[1] YU Zhen[1] LI Bu-Fan[1] YANG Jie[1] YIN Lin-Qi[1] YIN Zhang-Yue[1] YUAN Feng-Ye[1]
WEI Hai-Yang[1] LU Jia-Hao[1] FANG Shi-Cheng[1] CHEN Shuang[1] QIU Xi-Peng[1]

Abstract Large language models have garnered significant attention due to their strong generative and comprehension capabilities. However, they still face limitations in accessing real-time information and performing complex calculations. To better address user needs, empowering large language models with tool use capabilities has become a major topic of current research. Firstly, we clarify the fundamental concepts of tool use in large language models and then trace its developmental trajectory in chronological order. Subsequently, we summarize the datasets and technical approaches related to tool use, and analyze their applications in fields such as agent and embodied intelligence. Finally, we present an outlook on future research priorities and development directions in the field of tool use in large language models.

Key words Large language models, tool use, tool-augmented generation, agent, embodied intelligence

Citation Zheng Yi-Ning, Yu Zhen, Li Bu-Fan, Yang Jie, Yin Lin-Qi, Yin Zhang-Yue, Yuan Feng-Ye, Wei Hai-Yang, Lu Jia-Hao, Fang Shi-Cheng, Chen Shuang, Qiu Xi-Peng. Survey of tool use in large language models. *Acta Automatica Sinica*, 2025, **51**(11): 2371−2386

大语言模型 (Large language models, LLM) 凭借其卓越的语言理解与生成能力, 在众多领域中表现优异. 然而, 由于模型内在的局限性, 大语言模型在执行实时信息获取、精密计算等复杂任务时仍存在明显的不足. 通过引入搜索引擎、代码执行模块等外部工具, 大语言模型在复杂任务处理上的性能得以显著提升, 工具使用 (Tool use) 因此逐渐成为一个重要且专门的研究方向.

本文从工具使用的基本定义、发展路径、数据与评测、能力获取、扩展场景与发展前景六个方面对这一领域进行系统性探讨:

收稿日期 2024-12-13 录用日期 2025-04-22
Manuscript received December 13, 2024; accepted April 22, 2025
本文责任编委 左旺孟
Recommended by Associate Editor ZUO Wang-Meng
1. 复旦大学计算与智能创新学院 上海 200433
1. College of Computer Science and Artificial Intelligence, Fudan University, Shanghai 200433

1) 基本定义: 介绍大语言模型工具使用的基本概念, 并将其提炼为三种核心能力;

2) 发展路径: 按照时间顺序梳理工具使用的起源与演进路径, 重点分析代码式工具与接口式工具两种主要范式在近年来的研究趋势与技术规律;

3) 数据与评测: 系统地总结与工具使用相关的各种数据集与评测方法, 归纳不同数据集所覆盖的工具使用能力及其在任务评估中的作用;

4) 能力获取: 按照大语言模型在工具使用中展现的不同能力, 逐一阐述获取这些能力的方法, 包括训练策略、技术手段与实际效果;

5) 扩展场景: 探讨工具使用在工具增强生成之外的扩展应用场景, 着重讨论智能体和具身智能两个领域的实践与潜在发展方向;

6) 发展前景: 结合当前研究现状, 总结大语言模型工具使用的技术发展方向及其未来应用前景,

分析这一领域未来研究面临的技术挑战与突破点.

本文就大语言模型工具使用的关键研究要素和主要技术问题进行回顾和综述, 以帮助读者深入了解这一领域的最新发展及未来展望.

1 工具使用的定义

在大语言模型领域, 工具使用被视为一种重要的模型能力. 具体来说, 工具是能够独立于模型本身运行并返回结果的软件或硬件模块. 如图 1 所示, 模型通过生成特定语法的工具指令 (Tool command), 由外部系统识别并执行相应的工具调用, 随后对调用结果进行进一步处理. 这些工具指令也被称为工具调用指令 (Tool invocation command). 一种常见的工具指令形式是将工具视为函数, 由模型生成函数名及其参数以实现调用. 在这种情况下, 工具使用也被称为函数调用 (Function calling)[1−3].

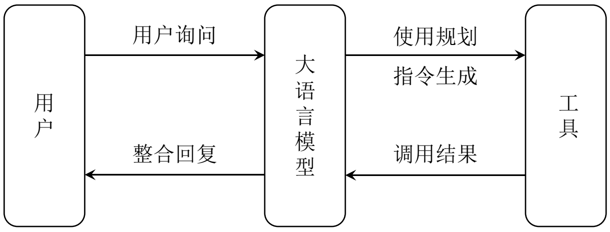

图 1 大语言模型工具使用的典型流程

Fig. 1 A typical procedure for tool use in large language models

通过工具使用, 大语言模型可以显著提升其能力. 例如, 在数值计算等需要精确结果的任务中, 工具使用能够有效地增强模型的精确计算能力[4−6]; 在通用对话模型中, 工具使用可以用于实时信息获取或生成定制化内容[1, 7−9], 这些方法通常被统称为工具增强生成 (Tool-augmented generation). 除了工具增强生成外, 大语言模型的工具使用能力还广泛应用于智能体 (Agent) 和具身智能 (Embodied intelligence) 领域, 用于生成智能体或机器人的具体动作. 例如, 工具使用能力支持智能体调用外部环境接口以完成复杂任务. 具体而言, 大语言模型的工具使用能力可以划分为三部分: 工具使用的规划能力、工具指令的生成能力和询问回复的生成能力. 本文将对这些能力分别进行定义和深入讨论.

工具使用的规划能力. 在工具使用过程中, 大语言模型首先需要判断用户询问是否需要使用工具, 这一能力尤为重要[1, 10]. 对于无需工具支持的询问, 强行使用工具可能导致用户体验的下降. 在解决复杂问题时, 模型可能需要调用多个工具, 而这些工具之间往往存在一定的关联性. 例如, 如图 2 所示, 工具 3 的参数依赖于工具 1 的返回结果, 而工具 2 的调用则与工具 1 和工具 3 相互独立. 在这种情况下, 模型对工具使用的规划能力至关重要. 为提升多工具协作中的效率, 一些研究在生成工具指令前对工具的使用流程进行明确规划[11−13]; 另一些研究则设计能够动态调整工具使用策略的系统, 通过反复尝试调用工具以优化任务的完成效果[14−16]. 此外, 实用性更高的对话模型通常在对话过程中隐式完成工具使用的规划, 根据上下文适时调用工具, 从而实现更自然的交互[1, 7−9].

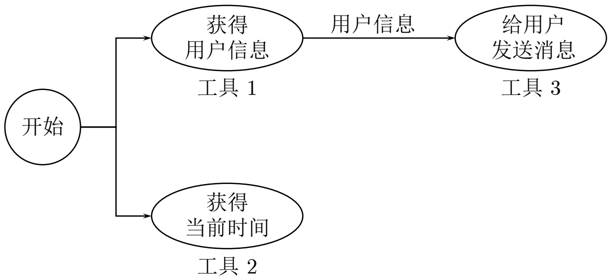

图 2 多工具调用示例

Fig. 2 An example of multiple tool invocations

工具指令的生成能力. 在完成工具使用规划后, 大语言模型需要生成工具指令以完成具体的工具调用. 工具指令是由模型生成的一段内容, 采用特殊语法撰写, 能够被外部系统识别和解析. 在早期研究中[4−6], 工具指令通常以特殊格式的纯文本形式呈现, 外部系统通过分析模型一次性生成的完整输出 (以结束符结尾) 来提取指令. 然而, 这种纯文本形式在流式输出场景中存在一定的局限性, 因为用户可能会在系统解析之前看到模型输出的工具指令语法, 从而影响了用户体验. 为解决这一问题, 后续研究[7−9] 引入包含特殊词符的工具指令格式. 这种设计不仅减少了工具指令与自然语言的混淆, 还支持外部系统以流式方式实时提取指令, 提高了指令解析的效率与用户体验的流畅性. 目前, 在通用工具使用任务中, 使用以 JSON 格式组织的工具指令较为普遍[8, 14, 17]. 该格式通常包含工具名称和参数信息, 能够更高效地支持工具调用与参数传递.

询问回复的生成能力. 在完成一次或多次工具调用后, 大语言模型需要整合工具调用的结果, 生成简明且连贯的自然语言回复. 早期研究表明, 当上下文中提供相关信息时, 大语言模型通常能够较好地利用这些信息. 然而, 当某些工具返回的结果过长时, 模型可能无法有效提取关键信息. 这可能导致生成的回复过于冗长或复杂, 影响了用户体验. 为解决这一问题, 可以借助外部工具对原始返回结果进行简化或摘要处理, 将核心内容提炼出来供模

型使用. 这种方法不仅能够缩短模型的输入长度, 还能提升回复的简洁性和可读性, 从而改善了用户的体验.

根据模型在推理过程中是否需要依赖上下文中的额外信息, 大语言模型的工具使用可以分为内生工具使用 (Intrinsic tool use) 和情境工具使用 (In-context tool use). 在内生工具使用中, 模型通过训练掌握一组固定工具, 并在推理时依靠其内生能力直接生成工具指令[5–6, 18]. 这种方法实现较为简单, 因为无需在推理阶段额外依赖上下文信息. 然而, 它的局限性在于模型只能调用训练数据中包含的工具, 难以适应训练过程中未见过的新工具. 相比之下, 情境工具使用要求模型在推理时依赖上下文提供的工具相关信息, 从而动态调用上下文中指定的候选工具. 情境工具使用的研究主要分为两类: 一类工作[4, 11, 15] 利用如 ChatGPT 等模型强大的指令遵循能力, 通过在推理阶段向模型提供候选工具的信息, 实现动态工具调用; 另一类工作[8, 14, 19–20] 通过泛化性训练, 能够让模型基于上下文中提供的工具信息生成调用指令. 这种泛化性训练方法能够让模型调用其在训练过程中未曾见过的工具, 显著提升其工具使用的灵活性与适应性.

2　工具使用的发展路径

工具使用与大语言模型相结合的研究最早由 Google 的 Parisi 等[5] 在 2022 年 5 月提出, 其研究成果命名为 TALM. 该研究首次将搜索引擎引入自然语言问答任务, 并将四则运算作为工具用于数学问答任务, 同时在参数规模介于 220 M 至 3 B 的 T5 系列模型[21] 上进行有监督微调. 研究结果初步揭示, 大语言模型在工具使用能力上的表现会随着模型参数规模的增加而逐步增强. 随后, 2022 年 11 月, Gao 等[4] 提出 PAL, 通过引入 Python 代码解释器完成数学推理任务, 进一步扩展了工具使用的应用场景, 并利用情境学习方法赋予 Codex 模型[22] 工具使用能力. 2023 年 3 月, OpenAI 发布闭源的大语言模型 GPT-4[1], 首次引入插件功能, 允许用户在对话中选择最多三个候选工具进行调用. 这一创新显著提升了大语言模型的工具扩展能力. 2023 年 4 月, 复旦大学推出开源模型 MOSS[7], 该模型通过在上下文中提供工具的描述及功能启用信息, 构建能够满足用户个性化需求的对话式语言模型.

随后, 关于大语言模型工具使用的研究逐渐形成两个主要范式. 其一是代码式工具使用, 主要以 Python 函数或神经网络模块作为大语言模型使用的工具[11, 23–25], 也有以代码解释器为工具、生成完整代码作为参数的研究[4, 26]; 其二是接口式工具使用, 重点研究基于应用程序编程接口 (Application programming interface, API) 的工具使用[10, 14, 19–20], 也包含使用接口操作手机应用[27–30]. 表 1 按时间顺序总结大语言模型工具使用研究中的关键工作.

在代码式工具使用中, HuggingGPT[11] 和 ViperGPT[23] 均于 2023 年 3 月提出. 两者通过调用神经网络或数据处理模块完成视觉问答任务, 但实现策略存在显著差异: HuggingGPT 使用配置文件作为工具调用指令, 功能更为多样, 可解决多种任务; ViperGPT 则采用 Python 代码作为工具调用指令, 专注于视觉问答任务. 后续研究[12, 24–26, 31–40] 普遍采用 Python 代码作为主要工具调用指令形式. 其中, LMTM[24] 提出一种策略, 通过语言模型生成并存储代码作为新工具.

在 2023 年, 许多接口式工具使用的数据集被提出[10, 14, 17, 19–20, 33], 为研究提供多种形式的数据支持. 其中, ToolLLM[14] 尽管存在数据质量不高的问题, 但依托 RapidAPI 平台[52], 是收集工具数量最多的数据集, 为后续研究奠定了坚实基础. 一些后续工作[2, 13, 50] 聚焦于提升数据质量和增强多工具之间的关联性, 同时也有研究基于已有数据集探索特定方向, 例如优化工具检索与调用规划[16, 35–44], 解决了模型输入长度过长的问题[32–41], 或通过课程学习和强化学习等方法提升了工具使用能力[33, 49]. 在 2024 年, 接口式工具使用的研究重点逐渐转向将手机应用作为工具的场景[27–30]. 这些研究探索语言模型与移动端应用结合的可能性, 不仅扩展了工具使用的应用场景, 还为未来发展提供新的方向和更广阔的前景.

在 2024 年, GPTs[1]、Gemini-1.5[53] 等闭源模型, 以及 Llama 3.1[8]、Qwen2.5[54]、DeepSeek-Coder-V2[55] 等开源模型, 在一定程度上融合代码式工具和接口式工具的使用能力. 以 GPTs 为例, OpenAI 允许用户通过配置网页智能体来选择是否启用 Python 代码执行、网页搜索和 DALL·E 图片生成等三种预设功能, 同时支持添加最多三个基于 Restful API 的自定义工具. 然而, 这些模型在不同接口式工具数据集上的表现仍存在较大差异, 大语言模型工具使用能力的泛化性研究依然是一个长期存在的挑战和重要研究方向.

纵观大语言模型工具使用的发展路径, 在一次对多个工具进行规划的场景下, 代码式工具使用在处理复杂问题时展现出更强的解决能力[12, 26, 34]. 这是由于代码式工具可利用通用编程语言自带的循环与条件语句, 表达复杂的多工具协作关系[36]. 而接口式工具使用在调用工具接口的同时, 还需额外设

表 1 大语言模型工具使用的发展

Table 1 The development of tool use in large language models

发布年份	名称	工具的类型	对话轮次	工具数量及关系	工具使用能力的获取方法
2022 年	TALM[5]	接口式工具	单次询问	多工具 (含复杂关系)	有监督微调
2022 年	PAL[4]	Python 解释器	单次询问	多工具	情境学习
2023 年	Toolformer[6]	接口式工具	单次询问	单工具	有监督微调
2023 年	GPT4-Plugin[1]	接口式工具	多轮对话	多工具	有监督微调 + 强化学习
2023 年	HuggingGPT[11]	神经网络模块	单次询问	多工具 (含复杂关系)	情境学习
2023 年	ViperGPT[23]	Python 函数	单次询问	多工具 (含复杂关系)	情境学习
2023 年	MOSS[7]	接口式工具	多轮对话	多工具	有监督微调
2023 年	API-Bank[19]	接口式工具	多轮对话	多工具	有监督微调
2023 年	APIBench[31]	Python 函数	单次询问	单工具	有监督微调
2023 年	GPT4Tools[15]	神经网络模块	多轮对话	多工具	情境学习
2023 年	ToolkenGPT[32]	接口式工具	单次询问	多工具 (含复杂关系)	有监督微调
2023 年	TRICE[18]	接口式工具	单次询问	多工具 (含复杂关系)	有监督微调 + 强化学习
2023 年	CRITIC[12]	Python 函数	单次询问	多工具	情境学习
2023 年	LATM[24]	Python 函数	单次询问	单工具	情境学习 + 创建工具
2023 年	CREATOR[25]	Python 函数	单次询问	多工具	情境学习 + 创建工具
2023 年	ToolBench[17]	接口式工具	单次询问	单工具	情境学习
2023 年	ToolAlpaca[20]	接口式工具	多轮对话	多工具	有监督微调
2023 年	ToolLLM[14]	接口式工具	单次询问	多工具	有监督微调
2023 年	Confucius[33]	接口式工具	单次询问	多工具	多阶段的有监督微调
2023 年	ToRA[26]	Python 解释器	单次询问	多工具 (含复杂关系)	有监督微调
2023 年	CRAFT[34]	Python 函数	单次询问	多工具 (含复杂关系)	情境学习
2023 年	MetaTool[10]	接口式工具	单次询问	多工具	情境学习
2023 年	ToolChain[35]	接口式工具	单次询问	多工具	情境学习 + 决策过程优化
2023 年	ToolTalk[36]	Python 函数	多轮对话	多工具 (含复杂关系)	情景学习
2023 年	CLOVA[37]	Python 函数	单次询问	多工具 (含复杂关系)	情境学习
2023 年	T-Eval[13]	接口式工具	多轮对话	多工具 (含复杂关系)	情境学习
2024 年	ToolEyes[38]	接口式工具	单次询问	多工具	有监督微调
2024 年	MLLM-Tool[39]	神经网络模块	单次询问	多工具 (含复杂关系)	有监督微调
2024 年	TroVE[40]	Python 函数	单次询问	多工具	情境学习 + 创建工具
2024 年	EasyTools[41]	接口式工具	单次询问	多工具	情境学习 + 工具文档压缩
2024 年	AnyTool[42]	接口式工具	单次询问	多工具	情境学习 + 检索过程优化
2024 年	SciToolBench[43]	Python 函数	单次询问	多工具	有监督微调
2024 年	ToolRerank[44]	接口式工具	单次询问	多工具	情境学习 + 检索过程优化
2024 年	STE[16]	接口式工具	单次询问	单工具	有监督微调 + 对错误反馈处理
2024 年	Seal-Tools[45]	接口式工具	单次询问	多工具 (含复杂关系)	有监督微调
2024 年	ToolPreference[46]	接口式工具	单次询问	多工具	有监督微调 + 偏好优化
2024 年	UltraTool[47]	接口式工具	多轮对话	多工具 (含复杂关系)	情境学习
2024 年	GTA[48]	接口式工具	单次询问	多工具 (含复杂关系)	情境学习
2024 年	Llama-3.1[8]	接口式工具	多轮对话	多工具	有监督微调 + 强化学习
2024 年	AppWorld[27]	手机应用	单次询问	多工具 (含复杂关系)	情境学习
2024 年	ShortcutsBench[28]	手机应用	单次询问	多工具	情境学习
2024 年	ToolSandbox[29]	手机应用	多轮对话	多工具 (含复杂关系)	有监督微调
2024 年	ToolACE[2]	接口式工具	多轮对话	多工具 (含复杂关系)	有监督微调
2024 年	StepTool[49]	接口式工具	单次询问	多工具	强化学习
2024 年	MTU-Bench[50]	接口式工具	多轮对话	多工具 (含复杂关系)	有监督微调
2024 年	ToolGen[51]	接口式工具	单次询问	多工具	有监督微调 + 工具文档压缩
2024 年	AndroidWorld[30]	手机应用	单次询问	多工具 (含复杂关系)	情境学习

计领域特定语言 (Domain-specific language, DSL) 以表示工具调用间的关联[38]. 在追求响应速度、需要并行调用多工具的工业场景中, 这种专门设计的 DSL 则可以比基于通用编程语言的代码式工具使用更加便捷[27−29]. 但在每一步仅调用单个工具, 并根据前一步的工具调用结果动态规划下一步行动的范式下, 代码式工具使用可以避免对 DSL 的依赖, 简化了复杂关系的表示.

3 工具使用的数据与评测

本节从多个维度对现有工具使用数据集的特点进行归类, 旨在为读者提供一个清晰而全面的概览. 同时, 我们还将分析并总结相关的评测方法, 探讨其在提升大语言模型工具使用能力中的作用.

3.1 数据集

自 Toolformer[6] 采用自监督方法构建首个工具使用的数据集以来, 越来越多的研究人员认识到将大语言模型与工具学习相结合所蕴含的巨大潜力, 并构造大量工具使用相关的数据集. 如表 2 所示, 我们从候选工具数量、数据规模、是否支持多轮对话、是否支持多工具使用以及多工具间的特殊关系等维度, 对 24 个工具使用数据集进行系统总结. 其中, "工具数量"指数据集中包含的工具总数量; "数据数量"指数据集中数据的总数量; "单轮"表示用户与大语言模型之间只进行单次交互; "多轮"指用户与大语言模型之间的交互是多轮对话的形式; "单工具"代表用户询问中只需调用单个工具; "多工具"代表用户询问中需要调用多个工具. 而对于多工具的数据, 我们还对其中的调用关系进行总结. 其中, "独立关系"表示多工具之间独立执行; "依赖关系"表示某些工具的执行需要依赖另一些工具执行后的结果; "嵌套关系"表示某些工具在执行的时候会调用其他工具. 除此之外, 我们还根据数据集的功能将其分为如下四个大类:

通用工具使用的数据集. 尽管赋予大语言模型工具使用能力能够显著拓展其应用范围, 但工具使用的复杂性仍限制模型在这一领域的表现, 亟须进

表 2 工具使用数据集概览

Table 2 The overview of tool use datasets

数据集	工具数量	数据数量	单轮	多轮	单工具	多工具	独立关系	依赖关系	嵌套关系
Toolformer[6]	5	12 500	√	×	√	×	×	×	×
API-Bank[19]	2 211	2 202	√	√	√	√	√	×	×
APIBench[31]	11 645	16 450	√	×	√	×	×	×	×
ToolBench[17]	232	2 764	√	×	√	×	×	×	×
ToolAlpaca[20]	426	3 938	√	√	√	√	√	×	×
RestBench[56]	94	157	√	×	√	×	×	×	×
ToolQA[57]	13	530	√	×	√	√	√	×	×
ToolLLM[14]	16 464	126 486	√	×	√	√	√	×	×
MetaTool[10]	199	21 127	√	×	√	√	√	√	×
TaskBench[58]	103	28 127	√	×	√	√	√	√	×
ToolTalk[36]	28	78	√	√	√	√	√	√	×
T-Eval[13]	15	533	×	√	×	√	√	√	×
ToolEyes[38]	568	382	√	×	√	√	√	×	×
UltraTool[47]	2 032	5 824	√	×	√	√	√	√	√
MLLM-Tool[39]	932	11 642	√	×	√	√	√	×	×
SciToolBench[43]	2 446	856	√	×	√	√	×	√	×
Seal-Tools[45]	4 076	14 076	√	×	√	√	√	√	√
ShortcutsBench[28]	1 414	7 627	√	×	√	√	√	×	×
GTA[48]	14	229	√	×	√	√	√	√	×
AppWorld[27]	457	750	√	×	√	√	√	√	√
ToolSandbox[29]	34	1 032	√	√	√	√	√	√	×
CToolEval[59]	398	6 816	√	×	√	√	√	√	×
ToolACE[2]	26 507	11 300	√	√	√	√	√	√	√
MTU-Bench[50]	136	159 061	√	√	√	√	√	√	√

一步改进. 为此, 大量专门用于评估和提升大语言模型工具使用能力的数据集应运而生. 其中, RestBench[56]、ToolLLM[14] 和 MetaTool[10] 聚焦于工具使用中的规划能力. 而 APIBench[31]、ToolBench[17] 和 ToolAlpaca[20] 更关注工具指令的生成, 包括工具选择和参数生成. 然而, 从用户询问中提取关键信息并生成具体参数, 是工具指令生成中的核心挑战. 为此, T-Eval[13]、TaskBench[58] 和 API-Blend[60] 在构建数据时特别强调对工具参数提取和处理能力的评估. 此外, API-Bank[19] 和 ToolEyes[38] 等数据集采用更为全面的设计, 涵盖工具使用能力评估的多个关键方面. GTA[48] 选择从现实世界中收集真实工具作为数据集中的工具, 贴近真实场景, 能够有效提升大语言模型在真实应用中的工具使用能力.

关注复杂关系的数据集. 目前, 一些常用的数据集[6, 17, 31, 56] 仅支持工具的单次调用, 而对包含多工具调用以及工具间复杂关系的关注较少. 然而, 现实中的工具调用往往涉及更复杂的参数结构, 并需要多个工具之间的协同配合. 因此, 生成多样化且高质量的工具调用数据对于提升大语言模型在实际应用中的表现至关重要. UltraTool[47]、SealTools[45]、ToolACE[2]和 MTU-Bench[50] 不仅考虑多工具使用的场景, 还涵盖工具间的多种调用关系, 包括并行调用、依赖调用和嵌套调用. 此外, 这些数据集还构建更贴近真实场景的工具调用多轮对话, 为模型处理复杂调用场景提供强有力的支持.

关注特定任务的数据集. 一些数据集专注于特定任务场景, 进一步挖掘大语言模型的工具使用能力. 例如, ToolTalk[36] 关注多轮对话交互, 开发一个数据集专门处理只能通过外部工具解决的模糊用户查询, 以提升模型在真实场景中的问答能力. SciToolBench[43] 针对五个科学领域中的挑战性问题, 提供通用且可扩展的工具支持, 其科学推理的性能超越 ChatGPT[1]. MLLM-Tool[39] 则结合大语言模型和多模态编码器, 使训练后的模型能够处理多模态输入指令, 并精准选择功能匹配的工具. 此外, AppBench[61]、ToolSandbox[29] 和 Shortcut-Bench[28] 专注于手机应用的调用, 探索大语言模型在移动端场景中的应用潜力. 这些数据集进一步扩展了工具使用的场景, 推动大语言模型在科学推理、多模态交互和移动端应用中的研究与实践.

关注鲁棒性的数据集. NoisyToolBench[62] 提出"按需询问"的方法, 以提升模型在处理不明确用户指令时的工具使用能力. ToolSword[63] 则重点分析大语言模型在工具使用过程中的安全挑战, 而 RoTBench[64] 通过设计五种不同级别的噪声环境, 在训练中引入环境多样性, 显著增强了大语言模型在工具学习中的鲁棒性. 需要注意的是, 某些 API 由于更新、访问限制或过期等原因往往存在不稳定性, 这会对评估结果的可重复性造成影响. 为此, StableToolBench[65] 引入虚拟 API 服务器, 结合缓存系统与 API 模拟器, 确保 API 在评估过程中的稳定性.

3.2 评测方法

大语言模型工具使用的根本目的是借助外部工具生成清晰、准确且符合用户预期的回复, 因此, 评价模型能力的核心在于对回复生成能力的评估. 此外, 还可以从更细粒度的角度对模型的整体工具使用表现进行评测, 例如: 指令生成能力、使用规划能力和工具检索能力. 在本节中, 我们将对这些评测维度依次进行介绍.

回复生成. 评估大语言模型工具使用的回复生成能力, 主要关注其输出是否满足用户需求并解决实际问题. 这一能力直接反映模型性能, 是评估用户体验和任务完成质量的核心指标. 高质量的回复不仅需要准确调用工具完成任务, 还需以清晰、符合用户预期的方式呈现结果. 评估方法包括使用 BLEU 分数[66] 和 ROUGE-L 分数[19, 67] 等自动化指标, 也可以通过能力较强大语言模型 (如 GPT-4[1]) 对生成的回复进行多维度的评分, 从而更全面地衡量模型回复的质量.

指令生成. 指令生成决定系统能否正确调用所需工具, 是大语言模型工具使用中仅次于回复生成的重要环节. 一些研究[14] 通过反复生成工具指令来提高最终正确率, 但这往往会带来无法接受的响应延迟, 因此衡量单次指令生成的正确性显得尤为重要. 要保证工具指令生成的正确性, 首先需要确保语法正确. ToolACE[2] 使用正则表达式检查指令格式, 从而快速验证语法一致性, 为自动化解析和执行提供保障. 在语法正确的基础上, 为更细致地分析模型能力, 还可以分别评估工具名称和调用参数的准确性. 对工具名称和调用参数的精准匹配 (Exact match, EM) 被视为最严格的评估标准[15]. 此外, 还可以通过计算工具调用的成功率 (Pass rate)[50, 56] 来评估工具指令生成的质量. 成功率反映生成指令在实际执行中完成任务的比例, 综合体现指令生成的准确性和任务完成的有效性.

使用规划. 在复杂需求中, 大语言模型的工具使用规划能力至关重要. 通过评估可以检验模型是否能准确判断用户查询是否需要工具辅助, 选择合适的工具, 并合理规划工具调用的顺序与逻辑. 一

种常用的评估方法是计算模型生成的规划与真实任务规划之间的最长公共子序列 (Longest common subsequence, LCS)[17, 68]. 虽然这种方法简单直观, 但仅关注字符或词序列的匹配, 忽略语义上的相似性. 并且, 最长公共子序列对顺序非常敏感, 无法容忍合理的步骤调整. 例如, 对于独立任务步骤的交换, 尽管对逻辑没有影响, 却可能导致得分下降. 此外, 在评估大语言模型的任务规划能力时, 一个关键因素是明确工具调用的顺序和逻辑[47, 69]. 许多任务需要经过复杂的多步骤规划, 包括调用多个工具、利用上下文信息或进行多轮对话. 模型能否高效完成这些任务, 很大程度上取决于其是否能合理拆解任务, 并正确安排各子任务的执行顺序. 因此, 一个工具调用任务可以形式化为一系列有序的子任务, 这种由多个具有复杂关系的子任务组成的序列通常被称为工作流 (Work flow)[70].

为更直观地表示多工具之间的依赖关系, 可以使用有向无环图 (Directed acyclic graph, DAG) 来表示工作流[58, 61, 70]. 一种使用 DAG 的评测指标[71]如下: 记模型预测和标准答案对应的图分别为 G_1 和 G_2. 对 $i=1$ 或 2, V_i 和 E_i 分别表示 G_i 的顶点集合和边集合. 通过对比 G_1 和 G_2 的顶点相似度 $Sim_{node}(G_1, G_2)$(式 1) 以及边相似度 $Sim_{edge}(G_1, G_2)$(式 2), 可以评估模型生成的任务规划是否合理.

$$Sim_{node}(G_1,\ G_2) = \frac{|V_1 \cap V_2|}{|V_1 \cup V_2|} \tag{1}$$

其中, $|V_1 \cap V_2|$ 表示图 G_1 和图 G_2 顶点集合的交集中顶点的数量; $|V_1 \cup V_2|$ 表示图 G_1 和图 G_2 顶点集合的并集中顶点的数量.

$$Sim_{edge}(G_1,\ G_2) = \frac{|E_1 \cap E_2|}{|E_1 \cup E_2|} \tag{2}$$

其中, $|E_1 \cap E_2|$ 表示图 G_1 和图 G_2 边集合的交集中边的数量; $|E_1 \cup E_2|$ 表示图 G_1 和图 G_2 边集合的并集中边的数量.

工具检索. 在情境工具使用中, 由于大语言模型的上下文长度有限, 通常需要通过检索来缩小候选工具的范围. 工具检索的结果直接决定任务能否被正确完成. 如果检索未能找到合适的工具, 可能导致任务无法解决 (缺少必要工具) 或结果不准确 (包含冗余工具). 工具检索的质量可以从两个方面评估: 第一, 是否检索出所有必要的工具; 第二, 是否避免检索出过多的工具. 假设 P_i 表示在任务 i 中检索出的工具集合, Q_i 表示任务 i 真实使用的工具集合, 任务总数为 N, 则在全部任务中工具检索的准确率 $Accuracy$ 可定义为式 (3).

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N}\frac{|Q_i \cap P_i|}{|Q_i|} \tag{3}$$

其中, $|Q_i \cap P_i|$ 是检索出的工具数量; $|Q_i|$ 是真实答案工具集合的大小. 此外, 还可以通过式 (4) 衡量在全部任务中工具检索的冗余率 $Redundancy$:

$$Redundancy = \frac{1}{N}\sum_{i=1}^{N}\frac{|P_i \setminus Q_i|}{|P_i|} \tag{4}$$

其中, $|P_i \setminus Q_i|$ 表示检索出的多余的工具数量; $|P_i|$ 是模型预测工具集合的大小.

4 工具使用能力的提升方法

目前, 即使先进的闭源模型 (如 ChatGPT[1]) 在 WebArena[72] 和 AndroidWorld[30] 等复杂工具应用场景中, 仍然面临指令遵循能力不足、上下文推理能力欠缺以及工具调用幻觉等问题. 因此, 大语言模型需要进一步提升工具使用能力以应对这些挑战. 现有的方法包括情境学习、监督微调、强化学习以及系统框架优化等. 接下来, 我们将在本节中对这些方法进行详细介绍.

情境学习. 情境学习通过在上下文中添加提示词、候选工具文档以及工具使用示例等信息, 提升了模型的工具使用能力. 它既能够将模型与具体工具集解耦, 使其根据当前工具信息快速切换工具集, 又可以对齐模型与工具的交互范式. PAL[4] 在上下文中同时输出代码片段和自然语言注释, 以增强计算和推理能力. CRITIC[12] 在生成回复时调用代码解释器和搜索引擎等工具, 并将结果加入上下文, 通过改正现有问题来改善回复质量. Kim 等[73] 提出基于提示词模板的“回复–批评–改进”流程, 利用情境学习能力提升回复的准确性. Wang 等[16] 则通过总结错误经验、自我反思, 并将其存储到外部记忆中, 在后续尝试中利用这些经验作为上下文信息, 避免了重复犯错.

监督微调. 此外, 在提升工具使用能力方面, 基于监督微调的训练方法也得到广泛应用. 相比于基于情境学习的方法, 经过微调后的模型具有内生工具使用能力, 具有更短的上下文长度, 响应速度和质量也都进一步提高. 在监督微调时, 一方面, 我们可以对工具使用数据的多样性进行增强[2, 26]. 工具使用数据的多样性包括工具多样性、任务多样性、文档风格多样性、调用格式多样性和用户询问多样性等. Liu 等[74] 将错误历史与当前执行结果同时维护在模型中, 避免了重复生成无效的工具指令. Abdelaziz 等[3] 将工具指令的生成过程分解为七个子任务, 包括三个高层次任务和四个低层次任务, 并在

子任务上微调以提升工具指令的生成能力. 另一方面, 我们可以对工具使用数据集的难度进行整体增强或动态适配, 包括从单工具调用到多工具调用的增强、从单轮对话到多轮对话的增强、从较易任务到较难任务的迭代训练等. TaskBench[58] 在工具调用依赖图中采样子图, 以生成更困难的多工具使用数据集. Confucius[33] 既设置多阶段学习, 从预热训练、同类别训练到跨类别训练, 又通过迭代的自我指导与内省反馈调整接下来的训练任务分布. GPT4Tools[15] 引入负面样本和上下文样本来增加数据集的复杂性和深度, 提高了模型对工具使用的理解和决策能力. AutoGLM[75] 和 WebRL[76] 都使用在线课程学习, 根据当前模型的能力动态调整训练任务的难度, 迭代提升智能体规划网页浏览任务的能力.

强化学习. 基于监督微调的训练方法存在一些缺点. 首先是监督微调的提升上限被专家轨迹所限制, 即可能难以跳出次优解, 错过更佳的工具解决方案; 其次, 对于高质量的工具学习数据集, 包括多样的任务和准确的调用轨迹, 其规模依赖于昂贵的人工标注, 难以大规模扩充; 最后, 基于监督微调的方法无法从工具使用的失败经验中学习, 也无法应对现实应用场景中的意外情况. 因此, ToolPlanner[77] 在不同粒度的 API 集合上, 将规划的相关性与准确性纳入奖励函数, 并基于得分高低配对正负样本, 以此训练模型偏好. 与此同时, 采用强化学习方法来提升工具使用的规划能力也带来一些新的挑战. 例如, 长动作序列中奖励信号稀疏, 或模型动作空间过大, 导致智能体训练效率低下; 工具使用环境中奖励函数建模复杂、奖励标志可能由于隐私或安全因素难以访问; 奖励劫持现象广泛存在, 导致智能体训练崩溃; 部分工具调用存在现实后果, 或者调用结果无法完全重现, 因此需要在高度定制的工具沙盒环境中进行训练等. Chen 等[78] 在 AppWorld[27] 中, 使用预定义的单元测试通过率作为结果奖励信号, 同时使用朴素的采样平均基线代替复杂的值网络, 对模型进行强化学习训练.

系统框架优化. 除了以上提升模型本身能力的方法外, 还可以通过优化整个工具调用框架以进一步提升大语言模型的工具使用能力. 首先, 在进行工具使用规划之前, 真实世界系统中大量的候选工具数量与大语言模型有限的上下文窗口之间存在冲突, 因此往往会先进行一次相关工具的检索[74, 79−80]. 工具检索完成后, 可选工具集合大幅缩小, 有助于减少上下文长度, 同时提高模型的工具使用准确率. 其次, 模型与工具系统的交互范式也可以进行优化. Liu 等[75] 以及 Huang 等[81] 在智能体的抽象规划和具体操作之间插入一个中间层, 强调进一步解耦智能体规划与操作的重要性. EasyTool[41] 通过将工具文档进行规范化处理, 降低了模型理解工具的难度. 最后, 系统框架能够为模型提供额外的信息增益. Xu 等[82] 强调可靠工具调用的重要性, 以及解决工具选择幻觉和调用幻觉的必要性, 并提出一个用来增强评估工具相关性和使用情况能力的对齐框架. ToolBench[14]、LLaVA-CoT[83]、ToolPlan-ner[77] 等工作引入搜索, 将工具调用看作图中状态节点间的转移, 使用束搜索、深度优先搜索、宽度优先搜索、蒙特卡洛树搜索等经典搜索算法引导模型在推理阶段进一步提升其工具使用能力[46, 77, 83−84].

5 工具使用的扩展场景

上述大语言模型在工具使用方面的应用场景主要集中于数学推理、知识问答和通用对话等自然语言处理任务. 然而, 除这些常规应用外, 其工具使用能力已进一步拓展至智能体和具身智能等领域. 在这些场景中, 模型借助工具生成具体的动作指令或交互策略, 为复杂任务的自动化执行及实际环境中的行为控制带来新的可能性.

5.1 工具使用与智能体

通用人工智能 (Artificial general intelligence, AGI) 的发展目标是构建能够在多领域自主学习并解决问题的智能体. OpenAI 和其他研究机构提出, 通用人工智能的发展可分为五个等级——L1: 聊天机器人, 具备基础对话能力; L2: 推理者, 能够像人类一样解决问题; L3: 智能体, 不仅能思考, 还能采取行动; L4: 创新者, 具备协助发明创造的能力; L5: 组织者, 能够完成复杂的组织协调工作. 在这一发展过程中, 工具使用是推动智能体发展的关键技术, 通过帮助智能体应对复杂挑战、提供闭环反馈、模仿专家轨迹, 避免引入冗余知识, 为智能体迈向更高等级奠定坚实基础.

工具使用助力智能体应对复杂任务. 传统智能体通常依赖预置知识或静态推理能力, 难以胜任开放世界中的复杂任务. 而工具使用通过赋予智能体动态调用外部工具的能力, 显著增强其在多样化场景中的适应性. Chen 等[85] 提出的视觉语言模型智能体, 通过调用预定义的动作工具集合, 在电子游戏环境中动态适应敌人的招式变化, 体现工具使用在增强智能体适应复杂开放场景能力中的关键作用. WebGPT[86] 通过调用搜索引擎工具, 使智能体能够实时检索和解析网页内容, 从动态信息中提取有效答案, 以解决复杂问题. WebShop[87] 利用网页

操作工具, 支持智能体模拟多步交互操作, 实现了复杂的商品搜索和购买任务, 展示了工具使用在实际应用中的巨大潜力. WebRL[76] 则通过智能体操控网页界面并执行预定义操作, 高效完成复杂的多步骤任务. SciAgent[43] 通过在科学推理中调用外部工具, 协助解决跨学科问题, 展现出优异的推理和分析能力.

工具使用为智能体提供闭环反馈环境. 在工具执行过程中, 智能体能够利用反馈信息反思自身行为, 并将其作为环境的奖励信号, 从而持续优化推理策略. StepTool[49] 通过闭环反馈机制, 使智能体在工具执行过程中有效利用反馈进行反思和行为优化, 并将反馈作为环境奖励信号, 从而显著提升其在动态环境中的性能和适应能力. 此外, Liu 等[74] 提出通过总结系统状态和推理历史, 分析工具调用的反馈信息, 优化当前策略并为下一步制定更精准的行动计划, 有效提高了智能体的决策效率. Wang 等[16] 进一步发现语言模型结合工具反馈机制的潜力, 智能体能够通过调用外部工具并分析反馈, 动态优化其任务规划和推理过程, 从而提升复杂任务的解决能力.

工具使用促进智能体模仿专家轨迹. 高质量的工具调用轨迹能够将复杂的规划问题具体化并逻辑化, 使智能体通过模仿学习掌握解决复杂问题的能力. 这些轨迹不仅为智能体提供明确的学习路径, 还有效降低了探索过程中的错误率. Qiao 等[88] 利用工具调用过程来模拟专家轨迹, 使智能体能够模仿专家的任务分解和规划方式, 从而优化全局任务的执行效果. AutoGLM[75] 利用工具调用轨迹进行行为克隆, 帮助智能体学习安卓应用中的具体操作步骤, 使其在复杂界面中表现出接近专家水平的能力.

工具使用避免智能体引入冗余知识. 工具使用通过将外部知识库封装为工具, 使智能体能够根据任务上下文动态调用所需的知识资源, 避免了因直接整合大量外部知识到模型中而可能引发的冗余与噪声问题. 这种方法不仅提高了模型的灵活性和适应性, 还使智能体在处理复杂任务时能够专注于相关信息, 从而提升了推理效率和响应质量. TOOL-ED[89] 提出将常识知识库封装为情感工具, 使模型能够根据对话上下文的需要, 按需查询用户情感的因果关系, 从而生成更精准且富有共情力的响应. 通过这种方式, 工具使用使智能体能够快速适应新的任务需求, 而无需重新训练或调整内部模型. 这不仅优化智能体的运行效率, 还提升了推理结果的针对性和可靠性.

工具使用降低智能体安全风险. 基于大语言模型技术的智能体在金融、医疗、自动驾驶等关键领域得到广泛应用. 然而, 在实际应用中, 这些智能体也逐渐暴露出隐私泄露、后门攻击、恶意代码生成与执行等安全风险. EIA[90] 指出通用网络智能体可能由于与被篡改的网站交互而导致用户隐私泄露. AgentPoison[91] 指出基于检索增强生成的智能体可能因知识库遭到恶意篡改而产生错误或有害输出. 针对上述问题, 研究者提出 GuardAgent[92] 框架, 其核心思想是通过利用大语言模型的逻辑推理能力, 将复杂的安全规则转化为可执行代码, 并通过调用外部工具来执行, 有效提升了智能体安全防护机制的精确性和可靠性. 工具的使用为智能体的安全风险防范提供一种有效的解决途径. 将复杂的安全需求转换为具体可执行的操作并通过工具实现, 不仅大幅提升了智能体的安全性, 还有效降低了运行成本和资源消耗. 这一方法为智能体领域未来的安全研究与实践提供重要的借鉴和指导.

综上所述, 工具使用在智能体的演化过程中扮演关键角色. 通过能力扩展、高效交互、动态适应和全面的问题求解支持, 工具使用为智能体的发展提供强大的技术支撑. 这一技术不仅弥补了传统模型的局限性, 还为实现通用人工智能奠定坚实基础.

5.2 工具使用与具身智能

传统的具身智能依赖于优化算法 (如抓取策略优化[93])、运动规划 (如路径生成[94]) 和轨迹规划 (如平滑运动轨迹生成[95]) 等技术, 专注于特定场景中的任务执行. 然而, 这些方法往往依赖于对环境的严格假设和任务的专用性, 缺乏泛化能力. 相比之下, 引入大语言模型显著提升了具身智能的灵活性和适应性. 凭借强大的理解与生成能力, 大语言模型能够迁移预训练阶段获取的知识, 降低对标注数据的依赖, 灵活应对复杂多变的环境与任务需求. 在基于大语言模型的具身智能框架中, 模型负责认知与任务规划, 工具执行具体的操作任务, 两者通过反馈机制共同构成闭环优化.

将自然语言指令转化为具体的任务步骤. 大语言模型在具身智能中的一个关键作用是将自然语言指令分解为具体的任务步骤. 例如, Code-as-Policies[96] 使用大语言模型根据自然语言命令生成机器人分层策略代码, 包括任务级动作、模块级动作、控制原语级动作和硬件接口级动作, 支持响应式策略和基于路径点的策略生成. 而 SayCan[97] 则利用大语言模型的语言理解与推理能力, 将用户指令分解为一系列子任务, 并结合技能库中已定义的能力, 通过可用性函数量化每项技能在当前环境下的成功

概率, 动态调整任务执行的优先级与决策流程, 从而实现高效任务分解与执行.

连接规划与物理执行. 工具在具身智能中被视为连接任务规划与物理执行的抽象接口, 涵盖高层动作函数、感知器的数据处理与图像识别, 以及算法与硬件的接口. 通过选择合适的工具, 大语言模型将语言规划转化为具体的物理动作. 例如, UniAff[98] 展示一种特殊的工具使用模式, 在该模式下, 工具仅作为被动执行的实体, 用于完成物理任务, 但不参与任务规划或动态反馈. 然而, 这种工具使用模式的局限性在于缺乏工具与智能体的深度交互, 难以动态适应复杂任务. 相比之下, Voxposer[99] 的框架展示工具与智能体深度结合的模式, 通过将感知接口与动作接口协同完成任务执行, 这一组合实现从语言规划到物理动作的无缝转换, 确保任务执行的高效性与可靠性. 在 ReKep[100] 系统中, 通过视觉感知器和动作规划模块等工具, 成功应对复杂任务的多阶段执行需求, 进一步展现了工具在具身智能中的核心作用.

大语言模型的引入为具身智能领域带来显著的突破, 其语言理解、任务分解和规划能力与工具的执行能力相辅相成, 不仅克服传统方法的泛化局限性, 还实现复杂任务的高效执行. 在未来, 如何进一步优化工具与大语言模型的交互深度, 并提升多模态任务的适应性, 将成为具身智能研究的关键方向.

6 未来与发展

本节将探讨当前工具使用领域亟待解决的技术难题、有待统一的领域规范, 以及具有潜力的发展方向. 未来的研究工作可围绕这些关键问题作进一步深入探索, 以推动该领域的持续发展.

6.1 高延迟对用户体验的影响

在大语言模型的工具使用场景中, 高延迟是影响用户体验的主要问题之一. 以 DFSDT[14] 为代表的一系列研究, 通过多次尝试调用不同工具以提高任务完成率. 然而, 这种策略的一个显著缺陷在于交互过程中的延迟大幅增加. 因此, 提高模型在单次工具调用中的生成准确性, 是一个需要长期关注的研究方向. 同时, 在某些情况下, 适时选择不调用工具, 也能够显著改善用户的整体体验.

在情境工具使用的任务中, 合理设计工具列表对于优化延迟至关重要. 通过精简工具文档内容和工具返回的结果[41], 分别可以减少模型在生成指令和生成回复过程中的输入, 减少推理时间. 对于那些工具列表固定且工具本身不经常更新的场景, 内生工具使用是一种有效的优化策略. 通过训练模型直接掌握固定工具的使用方式, 不仅可以减少模型输入, 还能够显著降低推理开销, 从而提高了整体效率. 这一方法尤其适用于对工具调用效率与稳定性要求较高的任务场景.

此外, 工具调用本身会不可避免地引入额外的时延. 尤其是在多工具调用的任务中, 工具的串行调用可能会进一步延长交互时间. 若能够实现多个工具的并行调用, 则可以在一定程度上提高处理效率并有效降低延迟.

6.2 数据集和评估的统一

如第 3 节所述, 工具使用领域已经积累大量数据集, 但不同数据集在工具的定义、描述方式及调用方式上存在较大差异. 目前, 开源的大语言模型通常只能在少量特定数据集上表现出较好的效果. 因此, 如何规范化工具使用领域的数据集, 充分评估并提升模型在工具指令生成、工具使用规划及工具使用整合回复等方面的能力, 已成为一个关键研究任务.

在评价指标方面, 现有工作同样缺乏统一的标准, 尤其是针对工具使用整合回复能力的评价. 使用 ROUGE[67]、BLEU[66] 等传统的定量指标无法有效反映回复的完成度, 而采用大语言模型进行评估的方法又可能因模型自身的偏好性而缺乏公平性. 相比之下, 依赖人类评估虽能提高评价的准确性, 但无法实现高效性与规模化. 因此, 开发统一而高效的评价指标是工具使用领域亟须解决的重要问题.

此外, 当前数据集中的大部分查询是通过大语言模型合成的, 只有少量数据集来源于真实用户的实际询问. 合成的查询可能无法准确反映人类的真实兴趣及其与模型交互的实际模式. 因此, 构建一个开源的交互系统, 并建立相应的标准, 可能会显著推动工具学习领域的发展.

6.3 幻觉缓解和意图完善

以外部知识库为代表的知识增强工具在帮助智能体缓解幻觉现象、增强对用户意图和情感的理解方面具有重要作用[89, 101−102]. 然而, 传统模型对固定知识库的依赖较强, 不加限制地引入外部知识可能导致噪声干扰, 从而影响了对话质量和模型性能. 目前, 对返回结果进行重排序[103] 得到广泛应用, 该步骤能够有效缓解模型在接收工具调用信息时产生的幻觉问题, 提升了回复质量. 此外, DMQR-RAG[104] 从改进查询本身出发, 提出四种重写查询策略和对应的自适应选择策略以提升召回的外部知识质量.

另一方面, 工具使用过程本身也引入幻觉. Xu 等[82] 将工具使用过程中的幻觉进行分类, 包括工具选择幻觉和工具使用幻觉, 强调消除工具使用幻觉的重要性. 在将来的工作中, 或许能将这些工作自然整合, 形成对齐用户意图、调用必需工具和召回有效信息的正反馈循环.

6.4 多模态工具使用

现有研究中, 大语言模型会通过使用工具处理语音、图像等多模态信息的输入[11, 23, 34–37], 以及生成语音、图像等多模态信息的输出[1, 7]. 然而, 目前多数大语言模型仍然以纯文本形式运作. 在未来的研究中, 具有多模态输入输出能力的模型是否能够引入全新的范式来完成多模态任务, 是一个非常值得关注的方向. 对于图文交错输入的模型, 一种常见的思路是将输入的图像信息映射到大语言模型的隐空间, 用若干图像词符进行表示, 模型生成工具使用指令的方式与纯文本模型的输出模式相似, 但允许工具的参数是图像词符. 在这种情况下, 要求工具能支持处理图像词符或者模型能对图像词符进行还原, 需要对整体系统进行设计. 而对于语音文本混合输入的模型, 不仅需要考虑将语音词符和文本词符映射到同一空间, 还需要考虑工具执行的时延对端到端语音系统流畅性的影响, 整体难度更上一层楼.

6.5 安全鲁棒的智能体构建

利用大语言模型的工具使用能力来构建具有安全性、鲁棒性的智能体是一个颇具潜力的发展方向. EIA[90]、AgentPoison[91] 等工作指出基于纯文本的智能体容易受到攻击, 而借助工具来增强智能体的安全性是一种可行的方案[92]. 但大语言模型的工具使用本身也存在一定鲁棒性[64] 和安全性[63] 的风险, 未来的研究需在确保工具使用的鲁棒性与安全性的前提下, 构建安全且稳健的智能体.

6.6 工具使用中的深度推理

近年来涌现出一些具备深度推理能力的大语言模型, 如 DeepSeek-R1[105] 和 QwQ[9], 它们在数学推理与代码生成等任务中表现出更优的效果[106]. 然而, 这类模型在复杂的工具使用场景中却存在明显不足: 一方面, 深度推理通常伴随更长的输出时间, 降低了模型在实际场景中的效率; 另一方面, 在部分工具使用任务中的性能甚至低于参数规模相近或更小的模型. 这一现象的内在原因以及如何有效地将深度推理能力迁移至工具使用任务中, 未来仍需深入探索与研究.

7 结束语

本文全面介绍大语言模型工具使用的发展. 从基本概念出发, 系统梳理其主要的研究进展, 并对与工具使用相关的数据集和评估方法进行分类与总结. 此外, 本文探讨获取工具使用能力的多种策略, 包括情境学习、监督微调、强化学习与系统框架优化等. 最后, 本文深入分析当前工具使用能力的拓展场景, 以及工具使用面临的挑战和未来的发展方向, 为相关领域研究人员提供参考.

References

1 Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Aleman F L, et al. GPT-4 technical report. arXiv preprint arXiv: 2303.08774, 2024.

2 Liu W W, Huang X, Zeng X S, Hao X L, Yu S, Li D X, et al. ToolACE: Winning the points of LLM function calling. arXiv preprint arXiv: 2409.00920, 2024.

3 Abdelaziz I, Basu K, Agarwal M, Kumaravel S, Stallone M, Panda R, et al. Granite-function calling model: Introducing function calling abilities via multi-task learning of granular tasks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing: Industry Track. Miami, USA: ACL, 2024. 1131−1139

4 Gao L Y, Madaan A, Zhou S Y, Alon U, Liu P F, Yang Y M, et al. PAL: Program-aided language models. In: Proceedings of the 40th International Conference on Machine Learning. Honolulu, USA: PMLR, 2023. 10764−10799

5 Parisi A, Zhao Y, Fiedel N. TALM: Tool augmented language models. arXiv preprint arXiv: 2205.12255, 2022.

6 Schick T, Dwivedi-Yu J, Dessi R, Raileanu R, Lomeli M, Hambro E, et al. Toolformer: Language models can teach themselves to use tools. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. New Orleans, USA: Curran Associates Inc., 2023. 68539−68551

7 Sun T X, Zhang X T, He Z F, Li P, Cheng Q Y, Liu X Y, et al. MOSS: An open conversational large language model. *Machine Intelligence Research*, 2024, **21**(5): 888−905

8 Grattafiori A, Dubey A, Jauhri A, Pandey A, Kadian A, Al-Dahle A, et al. The llama 3 herd of models. arXiv preprint arXiv: 2407.21783, 2024.

9 Qwen Team. QwQ-32B: Embracing the power of reinforcement learning [Online], available: https://qwenlm.github.io/blog/qwq-32b/, May 14, 2025

10 Huang Y, Shi J W, Li Y, Fan C R, Wu S Y, Zhang Q H, et al. MetaTool benchmark for large language models: Deciding whether to use tools and which to use. In: Proceedings of the 12th International Conference on Learning Representations. Vienna, Austria: OpenReview.net, 2024.

11 Shen Y L, Song K T, Tan X, Li D S, Lu W M, Zhuang Y T. HuggingGPT: Solving AI tasks with ChatGPT and its friends in hugging face. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. New Orleans, USA: Curran Associates Inc., 2023. Article No. 1657

12 Gou Z B, Shao Z H, Gong Y Y, Shen Y L, Yang Y J, Duan N, et al. CRITIC: Large language models can self-correct with tool-interactive critiquing. In: Proceedings of the 12th International Conference on Learning Representations. Vienna, Austria: OpenReview.net, 2024.

13 Chen Z H, Du W H, Zhang W W, Liu K K, Liu J N, Zheng M, et al. T-eval: Evaluating the tool utilization capability of large

language models step by step. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. Bangkok, Thailand: Association for Computational Linguistics, 2024. 9510−9529

14 Qin Y J, Liang S H, Ye Y N, Zhu K L, Yan L, Lu Y X, et al. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In: Proceedings of the 12th International Conference on Learning Representations. Vienna, Austria: OpenReview.net, 2024.

15 Yang R, Song L, Li Y W, Zhao S J, Ge Y X, Li X, et al. GPT4Tools: Teaching large language model to use tools via self-instruction. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. New Orleans, USA: Curran Associates Inc., 2023. Article No. 3149

16 Wang B S, Fang H, Eisner J, van Durme B, Su Y. LLMs in the imaginarium: Tool learning through simulated trial and error. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. Bangkok, Thailand: ACL, 2024. 10583−10604

17 Xu Q T, Hong F L, Li B, Hu C R, Chen Z Y, Zhang J. On the tool manipulation capability of open-source large language models. arXiv preprint arXiv: 2305.16504, 2023.

18 Qiao S F, Gui H H, Lv C F, Jia Q H, Chen H J, Zhang N Y. Making language models better tool learners with execution feedback. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). Mexico City, Mexico: ACL, 2024. 3550−3568

19 Li M H, Zhao Y X, Yu B W, Song F F, Li H Y, Yu H Y, et al. API-bank: A comprehensive benchmark for tool-augmented LLMs. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Singapore, Singapore: ACL, 2023. 3102−3116

20 Tang Q Y, Deng Z L, Lin H Y, Han X P, Liang Q, Cao B X, et al. ToolAlpaca: Generalized tool learning for language models with 3000 simulated cases. arXiv preprint arXiv: 2306.05301, 2023.

21 Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 2020, **21**(1): Article No. 140

22 Chen M, Tworek J, Jun H W, Yuan Q M, de Oliveira Pinto H P, Kaplan J, et al. Evaluating large language models trained on code. arXiv preprint arXiv: 2107.03374, 2021.

23 Surı́s D, Menon S, Vondrick C. ViperGPT: Visual inference via python execution for reasoning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). Paris, France: IEEE, 2023. 11854−11864

24 Cai T L, Wang X Z, Ma T Y, Chen X Y, Zhou D. Large language models as tool makers. In: Proceedings of the 12th International Conference on Learning Representations. Vienna, Austria: OpenReview.net, 2024.

25 Qian C, Han C, Fung Y, Qin Y J, Liu Z Y, Ji H. CREATOR: Tool creation for disentangling abstract and concrete reasoning of large language models. In: Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023. Singapore, Singapore: ACL, 2023. 6922−6939

26 Gou Z B, Shao Z H, Gong Y Y, Shen Y L, Yang Y J, Huang M L, et al. ToRA: A tool-integrated reasoning agent for mathematical problem solving. In: Proceedings of the 12th International Conference on Learning Representations. Vienna, Austria: OpenReview.net, 2024.

27 Trivedi H, Khot T, Hartmann M, Manku R, Dong V, Li E, et al. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Bangkok, Thailand: ACL, 2024. 16022−16076

28 Shen H Y, Li Y, Meng D S, Cai D Q, Qi S, Zhang L, et al. ShortcutsBench: A large-scale real-world benchmark for API-based agents. In: Proceedings of the 13th International Conference on Learning Representations. Singapore, Singapore: OpenReview.net, 2025.

29 Lu J R, Holleis T, Zhang Y Z, Aumayer B, Nan F, Bai H P, et al. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In: Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2025. Albuquerque, USA: ACL, 2025. 1160−1183

30 Rawles C, Clinckemaillie S, Chang Y F, Waltz J, Lau G, Fair M, et al. AndroidWorld: A dynamic benchmarking environment for autonomous agents. In: Proceedings of the 13th International Conference on Learning Representations. Singapore, Singapore: OpenReview.net, 2025.

31 Patil S G, Zhang T J, Wang X, Gonzalez J E. Gorilla: Large language model connected with massive APIs. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. Vancouver, Canada: NeurIPS, 2024.

32 Hao S B, Liu T Y, Wang Z, Hu Z T. ToolkenGPT: Augmenting frozen language models with massive tools via tool embeddings. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. New Orleans, USA: Curran Associates Inc., 2023. Article No. 1988

33 Gao S, Shi Z L, Zhu M H, Fang B W, Xin X, Ren P J, et al. Confucius: Iterative tool learning from introspection feedback by easy-to-difficult curriculum. In: Proceedings of the 38th AAAI Conference on Artificial Intelligence. Vancouver, Canada: AAAI Press, 2024. 18030−18038

34 Yuan L F, Chen Y Y, Wang X Y, Fung Y, Peng H, Ji H. CRAFT: Customizing LLMs by creating and retrieving from specialized toolsets. In: Proceedings of the 12th International Conference on Learning Representations. Vienna, Austria: OpenReview.net, 2024.

35 Zhuang Y C, Chen X, Yu T, Mitra S, Bursztyn V, Rossi R A, et al. ToolChain*: Efficient action space navigation in large language models with A* search. In: Proceedings of the 12th International Conference on Learning Representations. Vienna, Austria: OpenReview.net, 2024.

36 Farn N, Shin R. ToolTalk: Evaluating tool-usage in a conversational setting. arXiv preprint arXiv: 2311.10775, 2023.

37 Gao Z, Du Y T, Zhang X T, Ma X J, Han W, Zhu S C, et al. CLOVA: A closed-loop visual assistant with tool usage and update. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, USA: IEEE, 2024. 13258−13268

38 Ye J J, Li G Y, Gao S Y, Huang C S, Wu Y L, Li S X, et al. ToolEyes: Fine-grained evaluation for tool learning capabilities of large language models in real-world scenarios. In: Proceedings of the 31st International Conference on Computational Linguistics. Abu Dhabi, UAE: ACL, 2025. 156−187

39 Wang C Y, Luo W X, Dong S X, Xuan X H, Li Z X, Ma L, et al. MLLM-Tool: A multimodal large language model for tool agent learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). Tucson, USA: IEEE, 2025. 6678−6687

40 Wang Z R, Neubig G, Fried D. TroVE: Inducing verifiable and efficient toolboxes for solving programmatic tasks. In: Proceedings of the 41st International Conference on Machine Learning. Vienna, Austria: OpenReview.net, 2024.

41 Yuan S Y, Song K T, Chen J J, Tan X, Shen Y L, Ren K, et al. EASYTOOL: Enhancing LLM-based agents with concise tool instruction. In: Proceedings of the ICLR Workshop on Large Language Model (LLM) Agents. Vienna, Austria:

OpenReview.net, 2024.

42 Du Y, Wei F Y, Zhang H Y. AnyTool: Self-reflective, hierarchical agents for large-scale API calls. In: Proceedings of the 41st International Conference on Machine Learning. Vienna, Austria: OpenReview.net, 2024.

43 Ma Y B, Gou Z B, Hao J H, Xu R C, Wang S H, Pan L M, et al. SciAgent: Tool-augmented language models for scientific reasoning. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Miami, USA: ACL, 2024. 15701−15736

44 Zheng Y H, Li P, Liu W, Liu Y, Luan J, Wang B. ToolRerank: Adaptive and hierarchy-aware reranking for tool retrieval. In: Proceedings of the Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). Torino, Italia: ACL, 2024. 16263−16273

45 Wu M S, Zhu T, Han H, Tan C Y, Zhang X, Chen W L. Seal-tools: Self-instruct tool learning dataset for agent tuning and detailed benchmark. In: Proceedings of the 13th National CCF Conference on Natural Language Processing and Chinese Computing. Hangzhou, China: Springer, 2024. 372−384

46 Chen S J, Wang Y B, Wu Y F, Chen Q G, Xu Z, Luo W H, et al. Advancing tool-augmented large language models: Integrating insights from errors in inference trees. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. Vancouver, Canada: NeurIPS, 2024.

47 Huang S J, Zhong W J, Lu J Q, Zhu Q, Gao J H, Liu W W, et al. Planning, creation, usage: Benchmarking LLMs for comprehensive tool utilization in real-world complex scenarios. In: Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024. Bangkok, Thailand: ACL, 2024. 4363−4400

48 Wang J Z, Ma Z R, Li Y N, Zhang S Y, Chen C L, Chen K, et al. GTA: A benchmark for general tool agents. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. Vancouver, Canada: NeurIPS, 2024.

49 Yu Y Q, Wang Z F, Ma W Z, Guo Z C, Zhan J T, Wang S, et al. StepTool: A step-grained reinforcement learning framework for tool learning in LLMs. arXiv preprint arXiv: 2410.07745, 2024.

50 Wang P, Wu Y N, Wang N, Liu J H, Song X S, Peng Z Y, et al. MTU-Bench: A multi-granularity tool-use benchmark for large language models. In: Proceedings of the 13th International Conference on Learning Representations. Singapore, Singapore: OpenReview.net, 2025.

51 Wang R X, Han X D, Ji L, Wang S, Baldwin T, Li H N. ToolGen: Unified tool retrieval and calling via generation. In: Proceedings of the 13th International Conference on Learning Representations. Singapore, Singapore: OpenReview.net, 2025.

52 RapidAPI. RapidAPI: A platform for discovering and connecting to APIs [Online], available: https://rapidapi.com/, May 15, 2024

53 Georgiev P, Lei V I, Burnell R, Bai L B, Gulati A, Tanzer G, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv: 2403.05530, 2024.

54 Yang A, Yang B S, Hui B Y, Zheng B, Yu B W, Zhou C, et al. Qwen2 technical report. arXiv preprint arXiv: 2407.10671, 2024.

55 Zhu Q H, Guo D Y, Shao Z H, Yang D J, Wang P Y, Xu R X, et al. DeepSeek-coder-V2: Breaking the barrier of closed-source models in code intelligence. arXiv preprint arXiv: 2406.11931, 2024.

56 Song Y F, Xiong W M, Zhu D W, Wu W H, Qian H, Song M B, et al. RestGPT: Connecting large language models with real-world RESTful APIs. arXiv preprint arXiv: 2306.06624, 2023.

57 Zhuang Y C, Yu Y, Wang K, Sun H T, Zhang C. ToolQA: A dataset for LLM question answering with external tools. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. New Orleans, USA: Curran Associates Inc., 2023. Article No. 2180

58 Shen Y L, Song K T, Tan X, Zhang W Q, Ren K, Yuan S Y, et al. TaskBench: Benchmarking large language models for task automation. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. Vancouver, Canada: NeurIPS, 2024.

59 Guo Z S, Huang Y F, Xiong D Y. CToolEval: A Chinese benchmark for LLM-powered agent evaluation in real-world API interactions. In: Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024. Bangkok, Thailand: Association for Computational Linguistics, 2024. 15711−15724

60 Basu K, Abdelaziz I, Chaudhury S, Dan S, Crouse M, Munawar A, et al. API-BLEND: A comprehensive corpora for training and benchmarking API LLMs. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Bangkok, Thailand: ACL, 2024. 12859−12870

61 Wang H R, Wang R, Xue B Y, Xia H M, Cao J T, Liu Z M, et al. AppBench: Planning of multiple APIs from various APPs for complex user instruction. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Miami, USA: ACL, 2024. 15322−15336

62 Wang W X, Shi J L, Wang C Z, Lee C, Yuan Y L, Huang J T, et al. Learning to ask: When LLMs meet unclear instruction. arXiv preprint arXiv: 2409.00557, 2024.

63 Ye J J, Li S X, Li G Y, Huang C S, Gao S Y, Wu Y L, et al. ToolSword: Unveiling safety issues of large language models in tool learning across three stages. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Bangkok, Thailand: ACL, 2024. 2181−2211

64 Ye J J, Wu Y L, Gao S Y, Huang C S, Li S X, Li G Y, et al. RoTBench: A multi-level benchmark for evaluating the robustness of large language models in tool learning. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Miami, USA: ACL, 2024. 313−333

65 Guo Z C, Cheng S J, Wang H, Liang S H, Qin Y J, Li P, et al. StableToolBench: Towards stable large-scale benchmarking on tool learning of large language models. In: Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024. Bangkok, Thailand: ACL, 2024. 11143−11156

66 Papineni K, Roukos S, Ward T, Zhu W J. BLEU: A method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. Philadelphia, USA: Association for Computational Linguistics, 2002. 311−318

67 Lin C Y. ROUGE: A package for automatic evaluation of summaries. In: Proceedings of the Text Summarization Branches Out. Barcelona, Spain: Association for Computational Linguistics, 2004. 74−81

68 Bergroth L, Hakonen H, Raita T. A survey of longest common subsequence algorithms. In: Proceedings of the 7th International Symposium on String Processing and Information Retrieval. SPIRE 2000. A Curuna, Spain: IEEE, 2000. 39−48

69 Liu Y M, Peng X Y, Zhang Y W, Cao J N, Zhang X H, Cheng S, et al. Tool-planner: Dynamic solution tree planning for large language model with tool clustering. arXiv preprint arXiv: 2406.03807, 2024.

70 Qiao S F, Fang R N, Qiu Z S, Wang X B, Zhang N Y, Jiang Y, et al. Benchmarking agentic workflow generation. arXiv preprint arXiv: 2410.07869, 2024.

71 OpenMOSS. UnifiedToolHub [Online], available: https://github.com/OpenMOSS/UnifiedToolHub, May 14, 2025

72 Zhou S Y, Xu F F, Zhu H, Zhou X H, Lo R, Sridhar A, et al. WebArena: A realistic web environment for building autonomous agents. In: Proceedings of the 12th International Conference on Learning Representations. Vienna, Austria: OpenReview.net, 2024.

73 Kim G, Baldi P, McAleer S. Language models can solve computer tasks. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. New Orleans, USA: Curran Associates Inc., 2023. Article No. 1723

74 Liu Y L, Yuan Y L, Wang C W, Han J H, Ma Y Q, Zhang L, et al. From summary to action: Enhancing large language models for complex tasks with open world APIs. arXiv preprint arXiv: 2402.18157, 2024.

75 Liu X, Qin B, Liang D Z, Dong G, Lai H Y, Zhang H C, et al. AutoGLM: Autonomous foundation agents for GUIs. arXiv preprint arXiv: 2411.00820, 2024.

76 Qi Z H, Liu X, Iong I L, Lai H Y, Sun X Q, Sun J D, et al. WebRL: Training LLM web agents via self-evolving online curriculum reinforcement learning. In: Proceedings of the 13th International Conference on Learning Representations. Singapore, Singapore: OpenReview.net, 2025.

77 Wu Q Z, Liu W, Luan J, Wang B. ToolPlanner: A tool augmented LLM for multi granularity instructions with path planning and feedback. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Miami, USA: Association for Computational Linguistics, 2024. 18315−18339

78 Chen K, Cusumano-Towner M, Huval B, Petrenko A, Hamburger J, Koltun V, et al. Reinforcement learning for long-horizon interactive LLM agents. arXiv preprint arXiv: 2502.01600, 2025.

79 Kong Y L, Ruan J Q, Chen Y H, Zhang B, Bao T P, Shiwei S, et al. TPTU-v2: Boosting task planning and tool usage of large language model-based agents in real-world industry systems. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing: Industry Track. Miami, US: Association for Computational Linguistics, 2024. 371−385

80 Liu X K, Peng Z Y, Yi X Y, Xie X, Xiang L R, Liu Y C, et al. ToolNet: Connecting large language models with massive tools via tool graph. arXiv preprint arXiv: 2403.00839, 2024.

81 Huang W L, Abbeel P, Pathak D, Mordatch I. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In: Proceedings of the 39th International Conference on Machine Learning. Baltimore, USA: PMLR, 2022. 9118−9147

82 Xu H S, Zhu S, Wang Z H, Zheng H, Ma D, Cao R S, et al. Reducing tool hallucination via reliability alignment. arXiv preprint arXiv: 2412.04141, 2024.

83 Xu G W, Jin P, Li H, Song Y B, Sun L C, Yuan L. LLaVA-CoT: Let vision language models reason step-by-step. arXiv preprint arXiv: 2411.10440, 2024.

84 Koh J Y, McAleer S, Fried D, Salakhutdinov R. Tree search for language model agents. arXiv preprint arXiv: 2407.01476, 2024.

85 Chen P, Bu P, Song J, Gao Y, Zheng B. Can VLMs play action role-playing games? Take black myth Wukong as a study case. In: Proceedings of the NeurIPS Workshop on Open-World Agents. Vancouver, Canada: NeurIPS, 2024.

86 Nakano R, Hilton J, Balaji S, Wu J, Ouyang L, Kim C, et al. WebGPT: Browser-assisted question-answering with human feedback. arXiv preprint arXiv: 2112.09332, 2022.

87 Yao S Y, Chen H, Yang J, Narasimhan K. WebShop: Towards scalable real-world web interaction with grounded language agents. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. New Orleans, USA: Curran Associates Inc., 2022. Article No. 1508

88 Qiao S F, Fang R N, Zhang N Y, Zhu Y Q, Chen X, Deng S M, et al. Agent planning with world knowledge model. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. Vancouver, Canada: NeurIPS, 2024.

89 Cao H Y, Zhang Y Q, Feng S, Yang X C, Wang D L, Zhang Y F. TOOL-ED: Enhancing empathetic response generation with the tool calling capability of LLM. In: Proceedings of the 31st International Conference on Computational Linguistics. Abu Dhabi, UAE: Association for Computational Linguistics, 2025. 5305−5320

90 Liao Z Y, Mo L B, Xu C J, Kang M T, Zhang J W, Xiao C W, et al. EIA: Environmental injection attack on generalist web agents for privacy leakage. arXiv preprint arXiv: 2409.11295, 2025.

91 Chen Z R, Xiang Z, Xiao C W, Song D, Li B. AgentPoison: Red-teaming LLM agents via poisoning memory or knowledge bases. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. Vancouver, Canada: NeurIPS, 2024.

92 Xiang Z, Zheng L Z, Li Y J, Hong J Y, Li Q B, Xie H, et al. GuardAgent: Safeguard LLM agents by a guard agent via knowledge-enabled reasoning. arXiv preprint arXiv: 2406.09187, 2024.

93 Andrychowicz M, Baker B, Chociej M, Józefowicz R, McGrew B, Pachocki J, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 2020, **39**(1): 3−20

94 Kavraki L E, Svestka P, Latombe J C, Overmars M H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 1996, **12**(4): 566−580

95 Shen Z Y, Wilson J P, Harvey R, Gupta S. MRRT: Multiple rapidly-exploring random trees for fast online replanning in dynamic environments. arXiv preprint arXiv: 2104.11059, 2021.

96 Liang J, Huang W L, Xia F, Xu P, Hausman K, Ichter B, et al. Code as policies: Language model programs for embodied control. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). London, UK: IEEE, 2023. 9493−9500

97 Ichter B, Brohan A, Chebotar Y, Finn C, Hausman K, Herzog A, et al. Do as I can, not as I say: Grounding language in robotic affordances. In: Proceedings of the 6th Conference on Robot Learning. Auckland, New Zealand: PMLR, 2022. 287−318

98 Yu Q J, Huang S Y, Yuan X B, Jiang Z K, Hao C, Li X, et al. UniAff: A unified representation of affordances for tool usage and articulation with vision-language models. arXiv preprint arXiv: 2409.20551, 2024.

99 Huang W L, Wang C, Zhang R H, Li Y Z, Wu J J, Fei-Fei L. VoxPoser: Composable 3D value maps for robotic manipulation with language models. In: Proceedings of the 7th Conference on Robot Learning. Atlanta, USA: PMLR, 2023. 540−562

100 Huang W L, Wang C, Li Y Z, Zhang R H, Fei-Fei L. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. In: Proceedings of the 8th Conference on Robot Learning. Munich, Germany: PMLR, 2025. 4573−4602

101 Cai M X, Wang D L, Feng S, Zhang Y F. PECER: Empathetic response generation via dynamic personality extraction and contextual emotional reasoning. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Seoul, South Korea: IEEE, 2024. 10631−10635

102 Jin Q, Yang Y F, Chen Q Y, Lu Z Y. GeneGPT: Augmenting

large language models with domain tools for improved access to biomedical information. *Bioinformatics*, 2024, **40**(2): btae075

103 Xiao S T, Liu Z, Zhang P T, Muennighoff N, Lian D, Nie J Y. C-pack: Packed resources for general Chinese embeddings. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. Washington, USA: Association for Computing Machinery, 2024. 641−649

104 Li Z C, Wang J H, Jiang Z S, Mao H Y, Chen Z X, Du J Z, et al. DMQR-RAG: Diverse multi-query rewriting for RAG. arXiv preprint arXiv: 2411.13154, 2024.

105 Guo D Y, Yang D J, Zhang H W, Song J X, Zhang R Y, Xu R X, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. arXiv preprint arXiv: 2501.12948, 2025.

106 Zeng Z Y, Cheng Q Y, Yin Z Y, Wang B, Li S M, Zhou Y H, et al. Scaling of search and learning: A roadmap to reproduce o1 from reinforcement learning perspective. arXiv preprint arXiv: 2412.14135, 2024.

郑逸宁 复旦大学计算与智能创新学院博士研究生. 2019 年获得复旦大学学士学位. 主要研究方向为大语言模型中的代码生成与工具使用.
E-mail: ynzheng19@fudan.edu.cn
(**ZHENG Yi-Ning** Ph.D. candidate at the College of Computer Science and Artificial Intelligence, Fudan University. He received his bachelor degree from Fudan University in 2019. His research interest covers code generation and tool use in large language models.)

余 镇 复旦大学计算与智能创新学院硕士研究生. 2022 年获得江西师范大学学士学位. 主要研究方向为工具学习和个性化智能体.
E-mail: yuz24@m.fudan.edu.cn
(**YU Zhen** Master student at the College of Computer Science and Artificial Intelligence, Fudan University. He received his bachelor degree from Jiangxi Normal University in 2022. His research interest covers tool learning and personalized agent.)

李不凡 复旦大学计算与智能创新学院硕士研究生. 2024 年获得东北大学学士学位. 主要研究方向为大语言模型的后训练. E-mail: 24210240197@m.fudan.edu.cn
(**LI Bu-Fan** Master student at the College of Computer Science and Artificial Intelligence, Fudan University. He received his bachelor degree from Northeastern University in 2024. His main research interest is post-training of large language models.)

杨 捷 复旦大学计算与智能创新学院硕士研究生. 2021 年获得武汉大学学士学位. 主要研究方向为工具调用与智能体.
E-mail: yangj24@m.fudan.edu.cn
(**YANG Jie** Master student at the College of Computer Science and Artificial Intelligence, Fudan University. He received his bachelor degree from Wuhan University in 2021. His research interest covers tool use and agent.)

殷林琪 复旦大学计算与智能创新学院博士研究生. 2025 年获得复旦大学学士学位. 主要研究方向为工具学习.
E-mail: 21307140112@m.fudan.edu.cn
(**YIN Lin-Qi** Ph.D. candidate at the College of Computer Science and Artificial Intelligence, Fudan University. She received her bachelor degree from Fudan University in 2025. Her main research interest is tool learning.)

印张悦 复旦大学计算与智能创新学院博士研究生. 2021 年获得华东师范大学学士学位. 主要研究方向为大语言模型和机器推理.
E-mail: yinzy21@m.fudan.edu.cn
(**YIN Zhang-Yue** Ph.D. candidate at the College of Computer Science and Artificial Intelligence. He received his bachelor degree from East China Normal University in 2021. His research interest covers large language models and machine reasoning.)

袁枫烨 复旦大学计算与智能创新学院硕士研究生. 2024 年获得同济大学学士学位. 主要研究方向为大语言模型的工具调用与智能体.
E-mail: fyyuan24@m.fudan.edu.cn
(**YUAN Feng-Ye** Master student at the College of Computer Science and Artificial Intelligence, Fudan University. He received his bachelor degree from Tongji University in 2024. His research interest covers tool use in large language models and agent.)

魏海洋 复旦大学计算与智能创新学院硕士研究生. 2022 年获得安徽大学学士学位. 主要研究方向为工具学习和复杂意图检测.
E-mail: 23210240325@m.fudan.edu.cn
(**WEI Hai-Yang** Master student at

the College of Computer Science and Artificial Intelligence, Fudan University. He received his bachelor degree from Anhui University in 2022. His research interest covers tool learning and complex intent detection.)

陆嘉昊 复旦大学计算与智能创新学院博士研究生. 2025 年获得复旦大学学士学位. 主要研究方向为工具学习. E-mail: 21307130022@m.fudan.edu.cn

(**LU Jia-Hao** Ph.D. candidate at the College of Computer Science and Artificial Intelligence, Fudan University. He received his bachelor degree from Fudan University in 2025. His main research interest is tool learning.)

方世成 复旦大学计算与智能创新学院博士研究生. 2025 年获得复旦大学学士学位. 主要研究方向为工具学习. E-mail: 21307140067@m.fudan.edu.cn

(**FANG Shi-Cheng** Ph.D. candidate at the College of Computer Science and Artificial Intelligence, Fudan University. He received his bachelor degree from Fudan University in 2025. His main research interest is tool learning.)

陈 爽 复旦大学计算与智能创新学院博士后. 2023 年获得东北大学博士学位. 主要研究方向为自然语言处理和大语言模型中的情感计算. E-mail: chenshuang_fd@fudan.edu.cn

(**CHEN Shuang** Postdoctor at the College of Computer Science and Artificial Intelligence, Fudan University. She received her Ph.D. degree from Northeastern University in 2023. Her research interest covers natural language processing and sentiment computing in large language models.)

邱锡鹏 复旦大学计算与智能创新学院教授. 2006 年获得复旦大学博士学位. 主要研究方向为自然语言处理和大语言模型. 本文通信作者. E-mail: xpqiu@fudan.edu.cn

(**QIU Xi-Peng** Professor at the College of Computer Science and Artificial Intelligence, Fudan University. He received his Ph.D. degree from Fudan University in 2006. His research interest covers natural language processing and large language models. Corresponding author of this paper.)