

基于大语言模型的复杂任务自主规划处理框架

秦龙¹ 武万森¹ 刘丹¹ 胡越¹ 尹全军¹ 阳东升² 王飞跃^{3,4}

摘要 随着深度学习和自然语言处理技术的进步,大语言模型 (Large language models, LLMs) 展现出巨大潜力。尽管如此,它们在处理复杂任务时仍存在局限性,特别是在任务需要结合规划及外部工具调用的场合。面向这一挑战,提出国内首个以军事游戏为背景的中文的复杂任务规划与执行数据集 (Complex task planning and execution dataset, CTPaE), 以及一个基于 LLMs 的自主复杂任务规划 (Complex task planning, CTP) 处理框架 AutoPlan。该框架可以对复杂任务进行自主规划得到元任务序列,并使用递进式 ReAct 提示 (Progressive ReAct prompting, PRP) 方法对已规划的元任务逐步执行。该框架的有效性通过在 CTPaE 上的实验及与其他经典算法的比较分析得到了验证。项目地址: <https://github.com/LDLINGLINGLING/AutoPlan>。

关键词 大语言模型, 工具调用, 多步推理, 深度学习

引用格式 秦龙, 武万森, 刘丹, 胡越, 尹全军, 阳东升, 王飞跃. 基于大语言模型的复杂任务自主规划处理框架. 自动化学报, 2024, 50(4): 862–872

DOI 10.16383/j.aas.e240088

Autonomous Planning and Processing Framework for Complex Tasks Based on Large Language Models

QIN Long¹ WU Wan-Sen¹ LIU Dan¹ HU Yue¹ YIN Quan-Jun¹ YANG Dong-Sheng² WANG Fei-Yue^{3,4}

Abstract With the advancement of deep learning and natural language processing technologies, large language models (LLMs) have shown significant potential. Despite their power, they still face limitations when dealing with complex tasks, especially when the tasks require integrative planning and the invocation of external tools. In response to this challenge, this paper proposes the first domestic dataset for complex task planning and execution with a military game context, the Chinese complex task planning and execution dataset (CTPaE), and a new framework for autonomous complex task planning (CTP) using LLMs named AutoPlan. The framework is capable of autonomously planning complex tasks to generate a sequence of meta-tasks, and employs a progressive ReAct prompting (PRP) method to gradually execute the planned meta-tasks. The effectiveness of the framework has been validated through experiments on the CTPaE and comparative analysis with other classic algorithms. The link of project: <https://github.com/LDLINGLINGLING/AutoPlan>.

Key words Large language models (LLMs), tool-use, multi-hop reasoning, deep learning

Citation Qin Long, Wu Wan-Sen, Liu Dan, Hu Yue, Yin Quan-Jun, Yang Dong-Sheng, Wang Fei-Yue. Autonomous planning and processing framework for complex tasks based on large language models. *Acta Automatica Sinica*, 2024, 50(4): 862–872

收稿日期 2024-02-21 录用日期 2024-03-15

Manuscript received February 21, 2024; accepted March 15, 2024

国家自然科学基金 (62103420, 62103425, 62103428, 62306329), 湖南省自然科学基金 (2023JJ40676, 2021JJ40697, 2021JJ40702), 国防科技大学青年自主创新基金 (ZK-2023-31) 资助

Supported by National Natural Science Foundation of China (62103420, 62103425, 62103428, 62306329), Natural Science Foundation of Hunan Province (2023JJ40676, 2021JJ40697, 2021JJ40702), and Youth Independent Innovation Foundation of National University of Defense Technology (ZK-2023-31)

本文责任编辑 穆朝絮

Recommended by Associate Editor MU Chao-Xu

1. 国防科技大学系统工程学院 长沙 410073 2. 暨南大学公共/应急管理 广州 510632 3. 中国科学院自动化研究所复杂系统管理与控制国家重点实验室 北京 100190 4. 青岛智能产业技术研究院 青岛 266000

1. College of Systems Engineering, National University of Defense Technology, Changsha 410073 2. School of Public Management/Emergency Management, Jinan University, Guangzhou 510632 3. The State Key Laboratory for Management and

大语言模型 (Large language models, LLMs) 凭借其丰富的知识储备和强大的推理能力, 在自然语言理解和交互式知识查询等任务展现出令人瞩目的效果^[1]。然而, 大模型常面临幻觉输出、知识更新滞后以及领域知识理解不足等问题, 这些挑战影响了其在信息真实性、时效性和逻辑一致性等方面的可靠性^[2]。随着上下文学习 (In-context learning)^[3]、思维链 (Chain-of-thoughts)^[4] 以及外部资源注入^[5] 等方法的应用, 大模型在逻辑推理和复杂任务分析方面取得了巨大进步^[2]。工具接口的调用是大模型的典型推理应用之一, 典型的应用包括网络搜索^[6]、

Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190 4. Qingdao Academy of Intelligent Industries, Qingdao 266000

计算器调用^[7]、数据库查询^[8]以及数学问题求解^[9]等。这些应用可以弥补大模型在特定领域任务上的不足, 释放其在解决复杂任务上的潜力, 使系统更精准地理解和执行用户输入, 实现用户与系统之间更加自然、便捷的交互, 在中台调度^[10-11]、具身智能^[12]、军事模拟仿真^[13]和平行智能^[14]等领域具有广阔的应用前景。

为了指导大模型有效利用这些工具接口, 当前的研究通过在上下文中加入工具使用的示例来引导大模型^[15], 或者进行微调来优化大模型在工具使用上的性能^[7, 16]。例如, Toolformer^[7]采用自监督的方式来微调大模型, 使其能够获得调用 API 的能力, 在单步工具接口调用任务上取得了突破。但实际应用往往需要进行连续多步工具调用, 例如在指挥控制场景中, 计算与目标位置的距离这一基础任务涉及到三个步骤, 即: 1) 获取当前自身的位置; 2) 获取目标的位置; 3) 计算两个坐标之间的距离。虽然详细的指令能更好地引导模型实现任务目标, 但人类往往倾向于提供粗粒度、高层次的指令。在实际情况中, 指挥员通常会直接下达计算距离的命令, 而不会给出具体步骤, 这就需要大模型在理解上下文并且对该任务进行规划后, 调用相应的接口来执行。ToolLLM^[17]使用深度优先搜索策略, 边规划边执行指令。TPTU^[8]提出基于任务规划和工具调用的分步处理思路, 提升模型应对复杂任务的能力。然而, 现有的模型在这种高层次指令的任务规划过程中, 经常出现中间步骤缺失、重复、突然中断等不连贯的问题, 主要原因在于: 1) 由于工具类型和应用领域等差异, 现有的大模型仅靠提示学习或者思维链方法直接推理输出结果难以有效地泛化到新的工具、任务以及应用领域; 2) 具备单步工具调用和执行能力的模型在多步调用的过程中, 存在中途模型遗忘或混淆当前应执行的任务以及无法对历史的运算结果进行整合输出最终答案的问题。

为解决上述问题, 本文提出基于大语言模型的复杂任务自主规划处理框架 AutoPlan, 整体框架如图 1 所示。具体来说, AutoPlan 将一个复杂任务分成两个阶段, 先通过一个复杂任务规划 (Complex task planning, CTP) 模型对复杂任务进行规划, 得到一个元任务序列。然后再利用递进式 ReAct 提示 (Progressive ReAct prompting, PRP) 模型执行元任务序列, 并输出最终结果, 从而实现对复杂指令的自主规划处理。为验证上述方法的可行性, 本文构建全新的复杂任务规划与执行数据集 (Complex task planning and execution dataset, CTPaE), 旨在为复杂任务规划与执行研究提供一

个测试基准, 填补该领域的研究空白。CTPaE 的构建经历模板构建、自动拆解和人工评估三个步骤, 以军事战略游戏为背景, 具备多样的任务类型和工具种类。

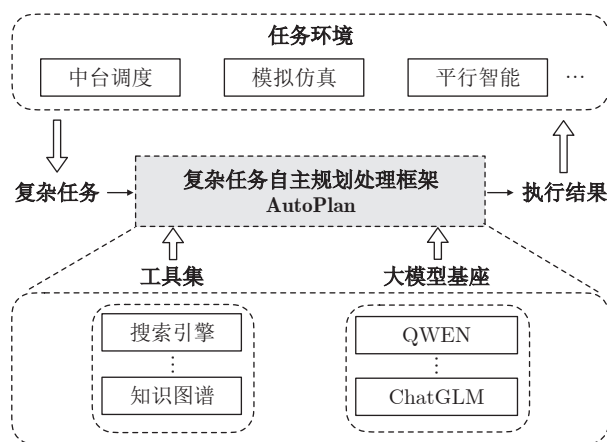


图 1 复杂任务处理框架 AutoPlan 示意图

Fig.1 Diagram of AutoPlan framework for complex task processing

总结来看, 本文的主要贡献在于: 1) 提出全新的复杂任务规划与执行数据集; 2) 提出基于大模型的自动规划和工具调用框架 AutoPlan, 利用先进任务规划后执行的思路, 并且设计 CTP 模型和 PRP 模型来有效解决复杂任务带来的挑战; 3) 与多个经典算法进行对比实验, 结果证明了 CTPaE 的挑战性以及本文提出方法的有效性。此外, AutoPlan 框架还具有广泛的应用前景, 例如在平行智能^[18]场景中, AutoPlan 可以赋能数字人使其具备独立解决问题的能力以及赋能机器人来协助人类完成各类任务, 执行人机交互、任务协调和计算实验等功能, 显著扩展了原始解决方案的能力范围^[19]。此外, 将 AutoPlan 框架与去中心化自治组织 (Decentralized autonomous organizations and decentralized autonomous operations, DAOs)^[20]相结合, 可以实现框架的去中心化、自主化、组织化和有序化, 极大地提高人机协作效率以及任务完成质量。

1 相关研究

使用大模型进行工具调用可以有效拓展大模型的应用范畴, 克服大模型传统对话模式的一些缺陷, 是人工智能领域内的一个热点问题, 并得到了广泛的研究。大模型可以利用软件工具, 如搜索引擎^[21]、移动设备^[22]、办公软件^[23]、计算器^[24]、深度模型^[25]、Python 解释器^[15]和其他通用 API^[26], 通过灵活控制多种工具来提高模型性能或完成复杂的工作流

程. Toolformer^[7] 采用一种自监督的方式来微调大模型, 使其能够获得自动调用 API 的能力. HuggingGPT^[27] 将大模型作为管理 AI 模型的控制, 利用来自人工智能社区 (如 Hugging Face) 的模型来自动解决用户的不同请求. TPTU^[8] 设计两类基于大模型的智能体来分别完成单步以及多步的工具调用任务. ART^[21] 利用思维链技术和上下文学习技术为新任务自动生成多步推理过程, 同时在每一步选择和利用最合适的可用工具. QWEN^[28] 使用 ReAct 方法^[29] 来使大模型调用未见过的工具. 同时, 采用 Self-instruct^[30] 的策略来对大模型进行有监督的微调, 以提高大模型调用工具的能力. WebGPT^[6] 和 WebCPM^[31] 使用搜索引擎来协助实现问答任务. 此外, RCI^[32] 根据提示方案递归地批评和改进模型, 以执行由自然语言引导的计算机任务. 此外, 也有大模型调用工具的相关研究在具身智能领域内开展^[33]. 大模型可以根据用户意图自动设计行动步骤, 引导机器人完成任务, 如 LLM-Planner^[12], 或直接生成可由机器人执行的底层代码, 如 ChatGPT for Robotics^[34]. PaLM-E^[35] 将传感器数据无缝集成到多模态大语言模型框架中, 从而实现机器人行动和任务的高效规划.

综上所述, 目前大语言模型在诸多领域的应用中展现了一定的任务理解和工具调用能力. 然而, 主要限于单一步骤的任务执行, 缺乏针对复杂任务的规划和多步执行功能. 在此背景下, 本文提出一种复杂任务自主规划处理框架. 此框架包含规划和执行两个阶段, 从而有力地解决了当前语言模型所面临的复杂任务处理能力不足的问题. 该框架将前瞻性规划与灵活执行相结合, 使得模型适应性和处理能力得以显著提升, 有效增强了大语言模型的复杂任务处理能力.

2 复杂任务规划与执行数据集 CTPaE

在本节中, 为验证大模型对复杂任务的规划与多步工具调用和执行能力, 本文提出复杂任务规划与执行数据集 CTPaE. CTPaE 以军事战略游戏为背景, 是首个专用于评测大模型的复杂任务规划与执行能力的中文数据集, 具有重要的研究意义.

2.1 任务定义

首先对相关概念进行介绍: 1) 复杂任务, 指无法通过单次调用工具完成的任务; 2) 元任务, 指可以仅靠单次调用工具或通过大模型自身计算就能完成的任务. 复杂任务因其高度复杂性和抽象性, 只有在进行任务规划后才能得到可执行的一系列元任

务, 在分配调用相应的工具资源后将元任务逐步执行. 最终, 需要将所有元任务的结果进行整合, 得到复杂任务的运行结果. 值得注意的是, 对复杂任务的规划需要将其变成具有逻辑关联的元任务集合, 只有这样, 后续模型才能利用规划结果进行工具的调用和执行. 若记复杂任务为 T , 则 T 经过规划后得到的具备逻辑关联的元任务集合记为 $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$, 其中 n 为元任务的数量. \mathcal{S} 中的任意元任务 s_i 的基本属性数据如表 1 所示, 元任务之间可能存在的逻辑关系示意图如图 2 所示. 对复杂任务 T 最终的执行结果定义为 y .

表 1 元任务的属性
Table 1 Properties of meta-tasks

任务属性	符号表示	属性描述
所在位置	s_i	在序列中的逻辑关系
工具需求	a_i	执行该任务的工具需求
参数配置	p_i	调用工具时的参数配置
运行结果	r_i	该任务的运行结果

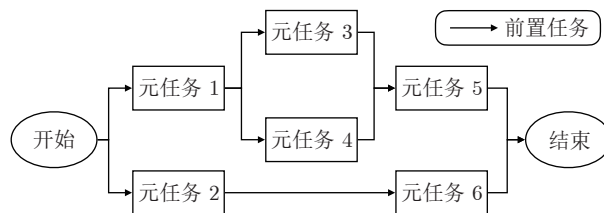


图 2 元任务之间的逻辑关系示意图
Fig. 2 Diagram illustrating the logical relationships between meta-tasks

2.2 工具定义

数据集中涉及到 12 种不同类型的工具, 可以支撑数据集中所有复杂任务指令需要的功能. 模型可以通过调用某个工具并输入相应的参数配置, 就可以得到工具的执行结果. 本文依托于军事战略游戏背景, 自定义多个工具函数且各个工具函数之间是相互独立的. 对各个工具的名称和简要介绍如表 2 所示.

2.3 任务指令

CTPaE 要求模型理解自然语言形式的复杂任务指令, 然后按照指令执行相应的工具, 并输出最终结果, 从而完成指令中包含的任务. 数据集中每一条数据为一个三元组 (T, \mathcal{S}, y) .

具体来说, 在每个任务的开始阶段, 模型会接收到一个自然语言形式的复杂任务指令 $T = \{tp,$

表 2 CTPaE 涉及的工具名称和功能介绍
Table 2 The name and function introduction of the tools involved in the CTPaE

工具名称	功能
google_search	通用搜索引擎, 可访问互联网、查询信息等
military_information_search	军事搜索引擎, 可访问军事内部网络、查询情报等
address_book	获取如电话、邮箱、地址等个人信息
email	发送和接收邮件
image_gen	根据输入的文本生成图像
situation_display	输入目标位置坐标和显示范围、当前敌我双方的战场态势图像, 并生成图片
calendar	获取当前时间和日期
map_search	可以查询地图上所有单位位置信息的工具, 返回所有敌军的位置信息
knowledge_graph	通过武器装备知识图谱获取各类武器装备的信息
math_formulation	可以通过 Python 的 eval(.) 函数计算出输入的字符串表达式结果并返回
weapon_launch	武器发射按钮是可以启动指定武器打击指定目标位置的工具
distance_calculation	可以计算给定目标单位之间的距离

$pp, x_0, x_1, x_2, \dots, x_l\}$, 其中 l 为任务指令的长度, x_i 代表指令的第 i 个字符, tp 是提示模板中的可用工具信息, pp 为提示模板中的可用参数信息. 模型在理解任务并做出规划后, 得到元任务集合 \mathcal{S} . 为了完成元任务, 模型需要从工具库 $\mathcal{A} = \{a_0, a_1, \dots, a_k\}$ 中调用与该元任务相关的工具, 其中 k 为工具库中工具的数量. 在所有的工具调用并执行结束后, 模型需要输出该任务的最终结果 y . y 可能是某个工具的执行结果, 也可能需要从多个工具的执行结果中进行整合得到, 还可能与工具的执行结果无关, 这里需要模型自行判断.

2.4 数据集构建方式

为构建多样、高质量且适用性强的数据集, 在进行构建和标注时主要遵循以下四个原则:

- 1) 逻辑正确且任务可执行. 在构建数据集之前, 应仔细规划任务的逻辑, 确保任务的目标明确、可行, 并且与实际应用场景相符. 这有助于确保数据集能够有效地训练模型, 并产生可靠的结果.
- 2) 多样的数据类型. 数据集的多样性是确保模型具有良好泛化能力的关键. 数据集中应该包含长链条任务、单步调用工具任务以及不使用工具的通用任务等各种类型, 这样能够帮助模型更好地适应不同的应用场景和问题.
- 3) 统一的数据格式和符号. 为避免造成困惑和歧义, 数据集中的数据格式和符号应该保持一致. 例如, 如果使用特定的词汇或术语来描述任务或工具, 应该在整个数据集中始终使用相同的词汇或术语.
- 4) 详尽的辅助提示信息. 数据集中提供的信息应该足够详细以便模型准确理解任务和可使用的工

具, 例如提供对候选的工具和接口进行介绍. 模型在了解工具功能和特性之后, 可以更准确地调用相关工具.

遵循以上原则, 采用人工标注和大模型生成的半自动方法进行数据集构建. 具体来说, 先人工构建若干条数据样本, 每条样本包括复杂任务指令 T 和具备逻辑关联的元任务集合 $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$. 此外, 加入一些额外的信息构建提示模板, 具体包括工具功能介绍、工具参数规范介绍, 通过上下文学习方法将现有的若干条样本和提示模板作为示例输入到 GPT4 中, 要求其按照这个模板生成相同格式的数据. 在对输出结果经过人工挑选和审核后, 得到任务分解结果. 每条任务指令 T 最终的运行结果 y 是通过人工标注得到的, 最终形成 CTPaE.

2.5 特点分析

本文构建的 CTPaE 共包含 2311 条长度不等的任务指令数据, 需要模型对抽象复杂问题进行规划, 得到严密逻辑的元任务, 然后执行每个元任务对应的工具才能得到最终结果. 这里划分 2111 条数据作为训练集, 200 条数据作为测试集. 对数据集中所有样本需要调用工具的次数进行统计与分析, 结果如图 3 所示. 可以看到, 数据集中的样本呈现出多样性, 并且需要调用多次工具的样本占比超过了一半, 这保证了数据集的复杂度. 此外, 图 4 为对任务指令长度的分析结果, 大部分指令的长度位于 0 ~ 100 个字符的区间内, 这有助于保证任务表达的清晰和准确. 另外, 数据长度的差异性也使得数据分布更加多样化.

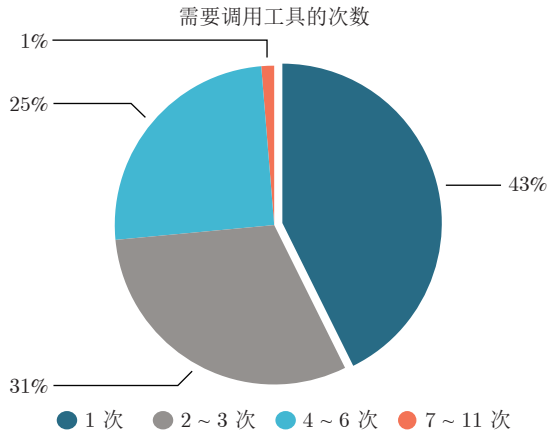


图3 每条样本需要调用工具的次数统计
Fig.3 Statistics on the number of tools used for each sample

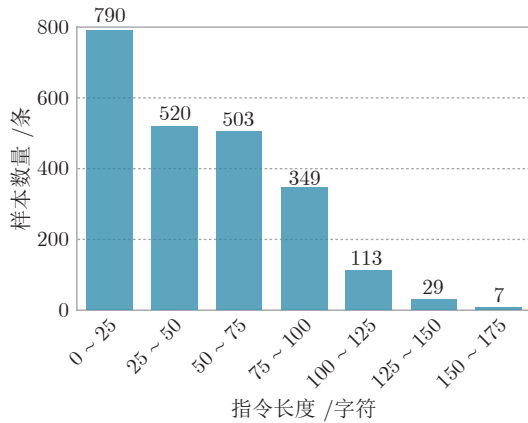


图4 指令长度分析
Fig.4 Analysis of instruction length

2.6 需求分析

由于本数据集包含的复杂任务具有很高的挑战性,普通的深度学习模型通常难以解决.从大模型的角度出发,为能够准确地理解复杂任务,做出合理的规划并且得到正确的执行结果,以下五种能力是模型所必备的:

1) 意图理解能力.能够理解从人类或系统中接收到的任务指令中所包含的意图.当任务指令高度复杂时,模型需要对指令进行深度理解与推理分析.

2) 任务规划能力.能够根据任务指令和自身能力将复杂任务分解成一系列具备逻辑关系的元任务序列,并且能根据任务或环境的变化进行动态调整.

3) 工具调用能力.一方面,能够选择各种现有工具或资源来执行复杂任务.另一方面,能够按照任务要求创建新的工具,拓展其能力范围.

4) 更新迭代能力.能够从日志、输入历史信息、运行结果和异常错误中进行迭代更新.通过从反馈

中不断学习,可以不断提高任务执行的性能和效率.

5) 总结提炼能力.能够在与用户、工具等进行多轮交互之后,总结交互历史信息,准确提炼出最终答案,完成指令中包含的任务.

2.7 评价指标

由于CTPaE主要用于衡量模型对复杂任务规划与多步调用工具进行执行的能力,因此,本文通过任务完成率、任务成功率、调用工具精确率和工具参数相似度四个指标,全面衡量模型的性能,涵盖了任务完成情况、工具使用情况以及参数配置情况.为便于说明,将模型输出的最终结果记为 y .假设数据集有 N 条数据,针对每一条数据 k ,模型输出记为 y_k ,真实标签记为 \hat{y}_k .为了完成每条数据指令中的任务,模型可能调用 M 次工具,那么记调用工具的集合为 $\mathcal{O} = \{a_1, a_2, \dots, a_m\}$.相应地,每一步调用工具也会将相关的参数传入,这里记为 $\mathcal{P} = \{p_1, p_2, \dots, p_m\}$.

1) 任务完成率 (Task completion rate, TCR): 对于给定的复杂任务,模型对任务进行规划并获得最终答案的样本的百分比.这个指标反映了模型生成可执行动作的能力及总结提炼答案的能力,定义为

$$\text{TCR} = \frac{\sum_{k=0}^N \mathbb{I}(y_k \neq \emptyset)}{N} \quad (1)$$

其中, $\mathbb{I}(y_k \neq \emptyset)$ 是判断函数,表示当 y_k 非空时为1,否则为0.

2) 任务成功率 (Task success rate, TSR): 对于给定的问题,模型输出最终结果且答案正确的比例.这里将原始数据 k 、模型输出 y_k 以及真实标签 \hat{y}_k 输入到QWEN-72B的模型中,由模型判断 y_k 与 \hat{y}_k 是否一致

$$\text{TSR} = \frac{\sum_{k=0}^N \mathbb{I}(y_k = \hat{y}_k)}{N} \quad (2)$$

其中, $\mathbb{I}(y_k = \hat{y}_k)$ 是判断函数,表示当 y_k 与 \hat{y}_k 相等时为1,否则为0.

3) 调用工具精确率 (Precision of tool using, PT): 通过模型输出的需调用工具集合与人工标注的需调用工具集合进行对比,计算工具调用的精确率.该指标反映了任务规划以及动作执行过程中的指令遵循能力

$$\text{PT} = \frac{\text{card}(\mathcal{O} \cap \hat{\mathcal{O}})}{M} \quad (3)$$

其中, $\hat{\mathcal{O}}$ 表示人工标注工具集合, $\text{card}(\cdot)$ 表示计算集合的元素个数.

4) 工具参数相似度 (Similarity of tool parameters, ST): 通过将模型生成的工具参数配置的字符串进行拼接后, 与人工标注的字符串拼接计算 Rouge-L. 由于 Rouge-L 包含对召回率和最大公共序列的计算, 该指标表明了对指定动作选取可执行的动作参数的准确性以及动作执行的逻辑合理性

$$ST = \text{Rouge-L}([\mathcal{P}], [\hat{\mathcal{P}}]) \quad (4)$$

其中, $\hat{\mathcal{p}}$ 表示人工标注参数集合, $[\cdot]$ 表示将集合中的所有元素进行拼接操作。

3 AutoPlan 模型

为有效解决复杂任务带来的挑战, 提出基于大模型的自动规划和工具调用框架 AutoPlan. AutoPlan 主要包括两个部分, 复杂任务规划模型 CTP 和递进式 ReAct 提示模型 PRP. 其中 CTP 模型用于自动任务规划, 将复杂任务变成最小元任务序列; 而 PRP 模型是按照逻辑顺序调用相应的工具递进式执行, 最终输出该任务的执行结果. 下面将从基线模型、AutoPlan 总体框架、CTP 模型、PRP 模型四个部分进行详细阐述。

3.1 基线模型

本文的基线模型采用 ReAct (Reason+Action) 方法^[29], 使大语言模型具备基本的任务执行能力. ReAct 是一种基于思维链的技术, 根据人类提供的工具函数, 对人类提出的问题逐步的思考、观察和执行, 最终得出答案. 具体来说, ReAct 方法要求大语言模型在每一个时刻 t 获取对当前环境的观察 obs_t , 并且根据观察来执行当前步的动作, 记为 act_t . 需要注意的是, 这里的 act_t 可以是调用工具, 也可以是模型对当前任务输出的分析和推理. 如果将大语言模型记为 $f_{\pi}(\cdot)$, 那么 $act_t = f_{\pi}(act_t|c_t)$, $c_t = (obs_1, act_1, \dots, obs_{t-1}, act_{t-1}, obs_t)$ 是大语言模型的上下文记录. ReAct 方法通过将模型的动作空间进一步拓展到大语言模型的语言空间, 有效提升了模型理解和推理能力。

3.2 模型框架

现有的中文开源模型如 QWEN、ChatGLM^[36] 等, 结合 ReAct 方法也具备一定的工具调用能力. 但是这些模型只能解决简单的单步工具调用和执行任务, 无法处理复杂的需要多步调用工具任务, 主要原因有两个方面: 1) 由于工具类型和应用领域等差异, 现有的大模型仅靠提示学习或思维链方法直接推理输出结果, 难以有效地解决 CTPaE 中的复杂任务; 2) 具备单步工具调用和执行能力的模型在

多步调用的过程中, 存在中途模型遗忘或混淆当前应执行的任务以及无法对历史的运算结果进行整合输出最终答案的问题。

针对以上问题, AutoPlan 提出两阶段的解决方案, 首先对一个小规模的模型进行微调, 得到专用的复杂任务规划模型 CTP. 然后针对遗忘和混淆问题, 提出递进式 ReAct 提示方法 PRP, 可以有效帮助模型追踪当前任务执行进度, 并整合历史信息输出任务最终的执行结果. 模型的框架如图 5 所示, 主要包括四个组成部分:

1) 提示模板. 这是 AutoPlan 框架的输入. 除任务指令之外, 本文还在指令前面加入额外的提示信息, 包括系统说明、工具的介绍、参数接口规范等. 此外, 还可以在提示信息中加入一些历史或者应用实例, 使得模型可以进行上下文学习。

2) 工具集. 工具集扩展了大语言模型的能力, 使其能够访问和处理超出其内部知识范围的信息, 与其他系统交互, 或执行其自身无法胜任的专门任务. 这里的工具集不限于一些工具的 API, 还可以是服务或者子系统的集合。

3) CTP 模型. 针对复杂任务规划的问题, 本文对现有的大语言模型进行微调, 得到 CTP 模型. CTP 模型可以对复杂任务进行规划得到元任务序列, 是后续任务能够顺利执行的重要前提. CTP 模型可以基于各类现有的开源模型进行训练得到, 如 QWEN、ChatGLM 等。

4) PRP 模型. PRP 模型负责执行一个规划完成的元任务序列. 具体来说, PRP 将所有元任务分成已完成、进行中和未完成三大类, 并随着任务的执行过程不断地迭代更新. 这种递进式的方法可以帮助模型有效追踪当前的任务进度, 避免遗忘和混淆等问题。

3.3 CTP 模型与 PRP 模型

3.3.1 CTP 模型

任务规划的目的是将复杂任务分解成多个有逻辑性且能够根据已有工具完成的元任务序列, 其中复杂任务是模型不能仅调用一次工具或者直接生成结果的任务, 元任务是指能够通过一次使用工具就能完成的任务。

CTP 模型输入为复杂任务, 输出为具备逻辑关联的元任务序列, 即

$$\mathcal{S} = f_{\theta}(T) \quad (5)$$

其中, $f_{\theta}(\cdot)$ 表示 CTP 模型, $\mathcal{S} = \{s_0, s_1, \dots, s_n\}$.

本文对 QWEN-1.8B 模型在 CTPaE 上进行全量微调, 实验设备为一张 A100-80G, 训练中

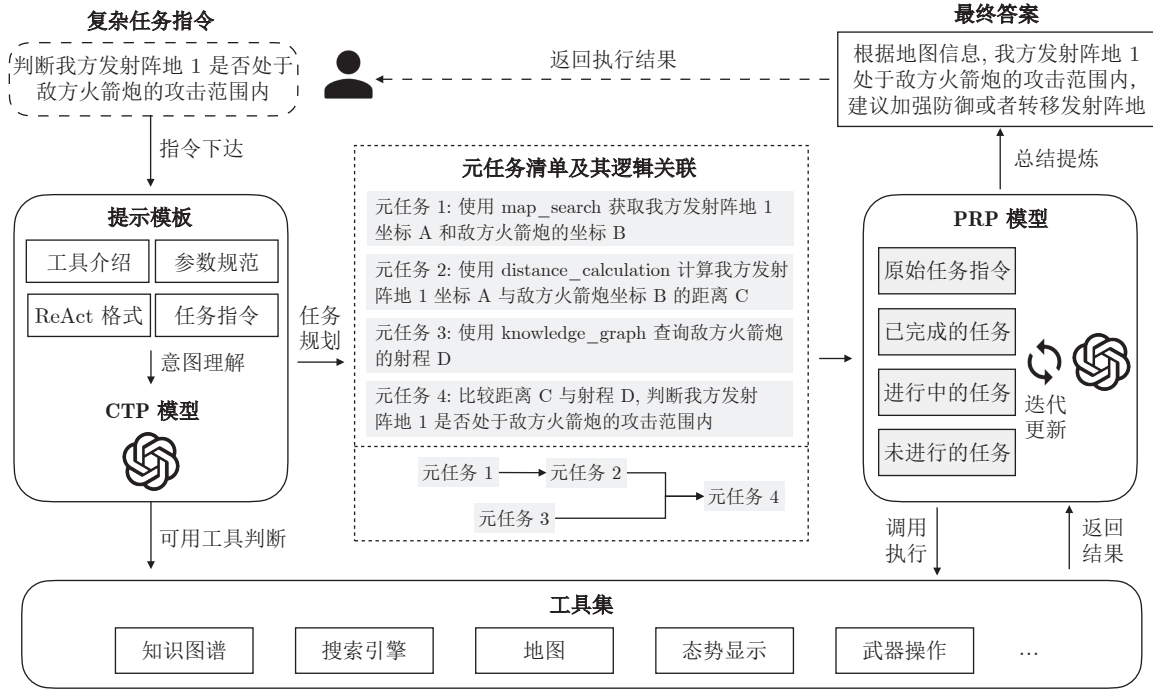


图 5 AutoPlan 总体框架示意图

Fig. 5 The diagram of the overall framework of AutoPlan

batch_size 为 7, 梯度累计次数为 8, 学习率为 2×10^{-5} , 共训练 300 步. 在训练数据中输入数据包括待分解任务和工具介绍、工具接口介绍、ReAct 格式规范, 在推理过程中需要将待分解任务转化为相同格式.

3.3.2 PRP 模型

虽然现有的一些大模型具备一定的工具调用与执行能力, 但是仅限于单步调用工具的简单任务, 对于需要多步调用的复杂任务, 性能会大幅下降. 主要原因在于多步调用工具过程中, 模型由于输入的文本长度过长, 注意力被分散, 导致中途遗忘或混淆当前应该执行的任务. 针对上述问题, 本文提出一种递进式 ReAct 提示模型, 能够帮助模型保持对当前执行任务的追踪. 具体来说, PRP 模型可以将元任务序列中的任务分成三类, 即已完成任务、进行中的任务以及未进行的任务, 在模型执行每一步的时候都作为提示信息加入, 并且随着任务的执行而不断地迭代更新. 这种递进式的方法可以帮助模型更好地关注当前执行的任务, 不会因复杂任务的文本长度过长而导致注意力分散^[37].

PRP 模型输出为前序规划好的元任务序列 \mathcal{S} , 然后逐步调用相应的工具并执行, 得到运行结果, 即

$$r_i = f_\phi(r_0, r_1, \dots, r_{i-1}, s_i) \quad (6)$$

其中, $f_\phi(\cdot)$ 表示 PRP 模型, r_i 为第 i 步的运行结果, 这里模型将前序的所有结果和当前任务一起输

入, 得到当前步的运行结果. 所有的元任务执行完毕后, 模型会整合所有的中间输出和历史信息, 输出最终的执行结果 y , 即

$$y = f_\phi(r_0, r_1, \dots, r_n) \quad (7)$$

4 实验与结果分析

本文的实验基于前文提出的 CTPaE 进行评测, 并与 ReAct^[29] 和 TPTU^[8] 方法进行对比. AutoPlan 的 CTP 模型是基于 QWEN-1.8B 在一张 NVIDIA A100 上进行全量微调. 为更全面地对比和分析实验结果, ReAct、TPTU 以及本文提出的 PRP 方法均在 QWEN-1.8B、QWEN-14B 和 QWEN-72B (int4) 三种不同规模的大语言模型上进行实验.

4.1 与其他方法的比较

首先探究模型的整体性能, 输入为原始的复杂任务指令, 模型需要根据指令来调用并执行相关的工具, 然后给出最终的答案. 这里采用前文中介绍的四个指标来对模型的性能进行综合评价, 即任务完成率 TCR、任务成功率 TSR、调用工具精确率 PT 以及工具参数相似度 ST, 其中任务成功率 TSR 为最主要的评价指标. 表 3 为本文提出的 AutoPlan 和其他方法在 CTPaE 上的性能对比结果. 其中 ReAct 技术是基于大语言模型, 通过 Reason+

表 3 与相关方法在 CTPaE 上的性能比较
Table 3 Performance comparison with related methods on the CTPaE

方法	规模 (B)	评价指标 (%)			
		TSR	TCR	PT	ST
ReAct	1.8	7.99	30.30	39.23	34.50
	14	37.37	90.00	60.57	48.99
	72	39.24	76.40	68.33	60.04
TPTU	1.8	0.60	18.80	33.07	24.92
	14	36.13	87.30	60.19	48.30
	72	39.84	76.80	68.14	59.96
AutoPlan	1.8	18.70	45.70	91.11	48.15
	14	52.30	94.70	90.81	79.24
	72	87.02	99.90	99.34	97.09

Act 的方式来直接进行任务执行. 而 TPTU 方法则是使用提示学习的方法, 利用大语言模型对任务进行规划, 然后再结合 ReAct 方法完成任务的执行. 本文的 AutoPlan 先利用 CTP 模块对复杂任务进行规划, 得到一个元任务序列, 然后再利用 PRP 方法对该元任务序列进行执行, 进而生成最终的执行结果.

从表 3 可以看出, 大语言模型规模对性能的影响显著. 随着模型规模的增加, 大多数评价指标的结果也有所提升. 这是因为更大的模型通常具有更强的理解和推理能力, 可以更好地捕捉输入数据的复杂性, 从而提高任务规划和执行的准确率.

相比于另外两种方法, AutoPlan 在所有模型规模和全部指标上均有显著优势. 例如, 在 72 B 规模下, AutoPlan 的 TCR 和 PT 分别为 99.90% 和 99.34%, 表明模型具备强大的理解能力以及后续的规划执行能力. 而 TSR 作为主要的性能评价指标, AutoPlan 的性能与另外两种方法相比提高了约 47%, 增长幅度均高于其他三个指标. 该结果充分证明了 AutoPlan 的有效性. 从表中还可以看到, AutoPlan 框架在 1.8 B 规模下的调用工具精确率已经达到了 91.11%, 但是任务成功率只有 18.70%, 原因在于模型无法给工具配置正确的参数, 即工具参数相似度只有 48.15%. 而随着模型规模的增加, 工具参数相似度得到了显著的提高, 进而也带来了更好的任务成功率.

4.2 消融实验分析

本文从两个方面开展消融实验分析: 1) 对任务规划模块的消融实验. 对比方法为不使用任务规划、利用 TPTU 方法进行规划和利用人工对复杂任务进行规划三种方法. 执行阶段均采用相同的 Re-

Act 方法. 2) 对执行策略的效果分析. 主要对比方法为 ReAct 方法. 前序的任务规划方法分别采用人工规划和本文提出的 CTP 模块进行规划.

4.2.1 任务规划效果分析

表 4 为不同任务规划方法的结果. 不难发现, TPTU 方法在小规模模型 (如 1.8 B 和 14 B 的结果) 上难以发挥出效果, 可能的原因是模型规模小、泛化能力不足, 导致在任务分解的细节上与人工标注的 label 存在一定误差, 这些误差在没有对 ReAct 的执行逻辑进行改进前被执行模型所放大. 在改进后的 ReAct 执行模型上获得较大提升, 这也反映了没有对任务执行模型进行指定任务微调时泛化能力不足的问题. 而在 72 B 的情况下, 模型具备了一定的推理能力后, 这种基于提示的任务规划方法可以发挥出一定的效果, 但是并不显著. 本文提出的 CTP 方法在三个不同规模的模型上的表现与不进行规划的结果基本一致, 与人工标注的结果仍有一定差距.

表 4 不同任务规划方法性能比较
Table 4 Performance comparison of different task planning methods

方法	规模 (B)	评价指标 (%)			
		TSR	TCR	PT	ST
不进行规划	1.8	7.99	30.30	39.23	34.50
	14	37.37	90.00	60.57	48.99
	72	39.24	76.40	68.33	60.04
TPTU	1.8	0.60	18.80	33.07	24.92
	14	36.13	87.30	60.19	48.30
	72	39.84	76.80	68.14	59.96
CTP	1.8	7.27	30.20	39.23	34.50
	14	37.54	89.90	60.63	49.02
	72	39.48	76.90	68.01	59.82
人工规划	1.8	8.17	43.38	39.24	34.50
	14	47.70	92.05	83.54	72.21
	72	61.69	97.60	86.78	80.75

4.2.2 执行策略效果分析

表 5 展示了不同任务执行策略的结果. 这里首先使用人工规划或 CTP 方法, 将复杂的任务分解为元任务序列, 然后分别采用 ReAct 方法和本文提出的 PRP 方法来执行这些元任务序列. 值得注意的是, 在两种不同的任务规划策略下, ReAct 方法的四个指标值均远低于 PRP 模型的执行方法. 具体而言, PRP 模型可以帮助 1.8 B 规模的模型在调用工具精确率上实现约 52% 的增长. 这表明递进式的策略可以有效帮助模型准确理解当前任务的执

表 5 不同任务执行策略性能比较
Table 5 Performance comparison of different task execution strategies

任务规划方法	任务执行方法	规模 (B)	评价指标 (%)			
			TSR	TCR	PT	ST
人工规划	ReAct	1.8	8.17	43.38	39.24	34.50
		14	47.70	92.05	83.54	72.21
		72	61.69	97.60	86.78	80.75
	PRP	1.8	18.39 (+10.22)	45.60 (+2.22)	91.15 (+51.91)	48.28 (+13.78)
		14	53.29 (+5.59)	94.70 (+2.65)	91.07 (+7.53)	79.44 (+7.23)
		72	86.43 (+24.74)	99.90 (+2.30)	99.47 (+12.69)	97.89 (+17.14)
CTP	ReAct	1.8	7.27	30.20	39.23	34.50
		14	37.54	89.90	60.63	49.02
		72	39.48	76.90	68.01	59.82
	PRP	1.8	18.70 (+11.43)	45.70 (+15.50)	91.11 (+51.88)	48.15 (+13.65)
		14	52.30 (+14.76)	94.70 (+4.80)	90.81 (+30.18)	79.24 (+30.22)
		72	87.02 (+47.54)	99.90 (+23.00)	99.34 (+31.33)	97.09 (+37.27)

行状态,从而调用正确的 API 来执行任务.此外,在任务完成率和任务成功率指标上,使用 CTP 模型结合 PRP 模型,可以达到甚至超越人工规划的结果.

5 结论与展望

本文针对军事游戏背景,构建国内首个中文的复杂任务规划与执行数据集 CTPaE,旨在测试大语言模型对复杂任务的理解、规划及执行能力,有效填补了该领域的空白.此外,本文提出基于大语言模型的复杂任务自主规划处理框架 AutoPlan,将复杂的任务执行分解成两个阶段,即任务规划和任务执行. AutoPlan 中的 CTP 模型通过微调获得复杂任务的规划能力,将复杂任务分解成元任务序列.为提高长序列任务的执行能力,本文提出递进式 ReAct 提示方法 PRP,保证模型能够准确关注到当前的任务进度.最后,将本文提出的方法在 CTPaE 上进行验证和分析,并与领域内的经典算法进行对比分析,验证了本文方法的有效性.通过消融实验分析,验证了两阶段执行方式以及各个模块的有效性.

本文提出的 AutoPlan 框架具有广泛的应用前景,例如实现指挥控制的自动化、工业生产中的任务自动化执行以及作为数字人等角色在平行智能中发挥重要作用.若将 DAOs 的技术引入,可以使框架进一步拓展,与其他任务环境、工具集及基础模型形成自主化、有序化和去中心化的应用结构,借助区块链技术的追溯性还可以保证任务指令内容的正确和规范.另外,当迁移到新的任务场景或者加入新的工具后,还可以借助基于检索生成的技术将

新的任务知识或工具知识加入到 AutoPlan 框架中,从而提高模型对新任务或新工具的泛化能力.

致谢

感谢国防科技大学的李新梦和朱正秋对本研究的深入讨论及宝贵建议.

References

1 Lu Jing-Wei, Guo Chao, Dai Xing-Yuan, Miao Qing-Hai, Wang Xing-Xia, Yang Jing, et al. The ChatGPT after: Opportunities and challenges of very large scale pre-trained models. *Acta Automatica Sinica*, 2023, **49**(4): 705–717 (卢经纬, 郭超, 戴星原, 缪青海, 王兴霞, 杨静, 等. 问答 ChatGPT 之后: 超大预训练模型的机遇和挑战. *自动化学报*, 2023, **49**(4): 705–717)

2 Zhao W X, Zhou K, Li J Y, Tang T Y, Wang X L, Hou Y P, et al. A survey of large language models. *arXiv preprint arXiv: 2303.18223*, 2023.

3 Dong Q X, Li L, Dai D M, Zheng C, Wu Z Y, Chang B B, et al. A survey for in-context learning. *arXiv preprint arXiv: 2301.00234*, 2023.

4 Wei J, Wang X Z, Schuurmans D, Bosma M, Ichter B, Xia F, et al. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv: 2201.11903*, 2023.

5 Cui J X, Li Z J, Yan Y, Chen B H, Yuan L. ChatLaw: Open-source legal large language model with integrated external knowledge bases. *arXiv preprint arXiv: 2306.16092*, 2023.

6 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.

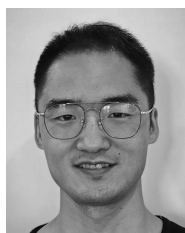
7 Schick T, Dwivedi-Yu J, Dessì R, Raileanu R, Lomeli M, Zettlemoyer L, et al. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv: 2302.04761*, 2023.

8 Ruan J Q, Chen Y H, Zhang B, Xu Z W, Bao T P, Du G Q, et al. TPTU: Task planning and tool usage of large language model-based AI agents. *arXiv preprint arXiv: 2308.03427*, 2023.

9 Patel A, Bhattamishra S, Goyal N. Are NLP models really able to solve simple math word problems? *arXiv preprint arXiv: 2103.07191*, 2021.

10 Wang F Y. New control paradigm for industry 5.0: From big models to foundation control and management. *IEEE/CAA*

- Journal of Automatica Sinica*, 2023, **10**(8): 1643–1646
- 11 Wang X X, Yang J, Wang Y T, Miao Q H, Wang F Y, Zhao A J, et al. Steps toward industry 5.0: Building “6S” parallel industries with cyber-physical-social intelligence. *IEEE/CAA Journal of Automatica Sinica*, 2023, **10**(8): 1692–1703
 - 12 Song C H, Wu J M, Washington C, Sadler B M, Chao W L, Su Y. LLM-Planner: Few-shot grounded planning for embodied agents with large language models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Vancouver, Canada: IEEE, 2023. 2998–3009
 - 13 Yang Dong-Sheng, Lu Jing-Wei, Li Qiang, Wang Fei-Yue. Issues and challenges of ChatGPT-like large scale pre-trained model for command and control. *Journal of Command and Control*, 2023, **9**(2): 146–155
(阳东升, 卢经纬, 李强, 王飞跃. 超大预训练模型在指挥控制领域的应用与挑战. 指挥与控制学报, 2023, **9**(2): 146–155)
 - 14 Wang F Y. Parallel intelligence in metaverses: Welcome to Hanoi! *IEEE Intelligent Systems*, 2022, **37**(1): 16–20
 - 15 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. Hawaii, USA: PMLR, 2023. 10764–10799
 - 16 Parisi A, Zhao Y, Fiedel N. TALM: Tool augmented language models. arXiv preprint arXiv: 2205.12255, 2022.
 - 17 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. arXiv preprint arXiv: 2307.16789, 2023.
 - 18 Yang Jing, Wang Xiao, Wang Yu-Tong, Liu Zhong-Min, Li Xiao-Shuang, Wang Fei-Yue. Parallel intelligence and CPSS in 30 years: An ACP approach. *Acta Automatica Sinica*, 2023, **49**(3): 614–634
(杨静, 王晓, 王雨桐, 刘忠民, 李小双, 王飞跃. 平行智能与 CPSS: 三十年发展的回顾与展望. 自动化学报, 2023, **49**(3): 614–634)
 - 19 Wu W S, Yang W Y, Li J J, Zhao Y, Zhu Z Q, Chen B, et al. Autonomous crowdsensing: Operating and organizing crowdsensing for sensing automation. *IEEE Transactions on Intelligent Vehicles*, DOI: 10.1109/TIV.2024.3355508
 - 20 Wang Y T, Wang X, Wang X X, Yang J, Kwan O, Li L X, et al. The ChatGPT after: Building knowledge factories for knowledge workers with knowledge automation. *IEEE/CAA Journal of Automatica Sinica*, 2023, **10**(11): 2041–2044
 - 21 Paranjape B, Lundberg S, Singh S, Hajishirzi H, Zettlemoyer L, Ribeiro M T. ART: Automatic multi-step reasoning and tool-use for large language models. arXiv preprint arXiv: 2303.09014, 2023.
 - 22 Wang B, Li G, Li Y. Enabling conversational interaction with mobile UI using large language models. In: Proceedings of the CHI Conference on Human Factors in Computing Systems. Hamburg, Germany: ACM, 2023. 1–17
 - 23 Li H X, Su J R, Chen Y T, Li Q, Zhang Z X. SheetCopilot: Bringing software productivity to the next level through large language models. arXiv preprint arXiv: 2305.19308, 2023.
 - 24 Chen Z P, Zhou K, Zhang B C, Gong Z, Zhao W X, Wen J R. ChatCoT: Tool-augmented chain-of-thought reasoning on chat-based large language models. arXiv preprint arXiv: 2305.14323, 2023.
 - 25 Patil S G, Zhang T J, Wang X, Gonzalez J E. Gorilla: Large language model connected with massive APIs. arXiv preprint arXiv: 2305.15334, 2023.
 - 26 Hao S B, Liu T Y, Wang Z, Hu Z T. ToolkenGPT: Augmenting frozen language models with massive tools via tool embeddings. arXiv preprint arXiv: 2305.11554, 2024.
 - 27 Shen Y L, Song K T, Tan X, Li D S, Lu W M, Zhuang Y T, et al. HuggingGPT: Solving AI tasks with ChatGPT and its friends in hugging face. arXiv preprint arXiv: 2303.17580, 2023.
 - 28 Bai J Z, Bai S, Chu Y F, Cui Z Y, Dang K, Deng X D, et al. QWEN technical report. arXiv preprint arXiv: 2309.16609, 2023.
 - 29 Yao S Y, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, et al. ReAct: Synergizing reasoning and acting in language models. arXiv preprint arXiv: 2210.03629, 2023.
 - 30 Wang Y Z, Kordi Y, Mishra S, Liu A, Smith N A, Khashabi D, et al. Self-instruct: Aligning language models with self-generated instructions. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. Toronto, Canada: ACL, 2023. 13484–13508
 - 31 Qin Y J, Cai Z H, Jin D, Yan L, Liang S H, Zhu K L, et al. WebCPM: Interactive web search for Chinese long-form question answering. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. Toronto, Canada: ACL, 2023. 8968–8988
 - 32 Kim G, Baldi P, McAleer S. Language models can solve computer tasks. arXiv preprint arXiv: 2303.17491, 2023.
 - 33 Wu W S, Chang T, Li X M, Yin Q J, Hu Y. Vision-language navigation: A survey and taxonomy. *Neural Computing and Applications*, 2024, **36**(7): 3291–3316
 - 34 Vemprala S, Bonatti R, Bucker A, Kapoor A. ChatGPT for robotics: Design principles and model abilities. arXiv preprint arXiv: 2306.17582, 2023.
 - 35 Driess D, Xia F, Sajjadi M, Lynch C, Chowdhery A, Ichter B, et al. PaLM-E: An embodied multimodal language model. arXiv preprint arXiv: 2303.03378, 2023.
 - 36 Du Z X, Qian Y J, Liu X, Ding M, Qiu J Z, Yang Z L, et al. GLM: General language model pretraining with autoregressive blank infilling. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. Dublin, Ireland: ACL, 2023. 320–335
 - 37 Xiao G X, Tian Y D, Chen B D, Han S, Lewis M. Efficient streaming language models with attention sinks. arXiv preprint arXiv: 2309.17453, 2023.



秦 龙 国防科技大学系统工程学院副研究员。2014 年获得国防科技大学博士学位。主要研究方向为复杂系统建模与仿真。

E-mail: qldb2007@sina.com

(QIN Long Associate researcher at the College of Systems Engineering, National University of Defense Technology. He received his Ph.D. degree from National University of Defense Technology in 2014. His research interest covers modeling and simulation of complex systems.)



武万森 国防科技大学系统工程学院博士研究生。2018 年获得国防科技大学学士学位。主要研究方向为视觉语言多模态。本文通信作者。

E-mail: wuwansen14@nudt.edu.cn

(WU Wan-Sen Ph.D. candidate at the College of Systems Engineering, National University of Defense Technology. He received his bachelor degree from National University of Defense Technology in 2018. His main research interest is vision-and-language multi-modality. Corresponding author of this paper.)



刘 丹 国防科技大学系统工程学院算法工程师. 主要研究方向为大语言模型, 自然语言处理.

E-mail: 15616297890@163.com

(**LIU Dan** Algorithm engineer at the College of Systems Engineering, National University of Defense

Technology. His research interest covers large language models and natural language processing.)



胡 越 国防科技大学系统工程学院讲师. 2021 年获得国防科技大学博士学位. 主要研究方向为智能启发式搜索与系统仿真.

E-mail: huyue11@nudt.edu.cn

(**HU Yue** Lecturer at the College of Systems Engineering, National

University of Defense Technology. He received his Ph.D. degree from National University of Defense Technology in 2021. His research interest covers intelligent heuristic search and system simulation.)



尹全军 国防科技大学系统工程学院研究员. 2005 年获得国防科技大学博士学位. 主要研究方向为行为建模, 云仿真.

E-mail: yin_quanjun@163.com

(**YIN Quan-Jun** Researcher at the College of Systems Engineering, Na-

tional University of Defense Technology. He received his Ph.D. degree from National University of Defense Technology in 2005. His research interest covers behavior modeling and cloud simulation.)



阳东升 暨南大学公共/应急管理学院教授. 主要研究方向为指挥控制理论与方法.

E-mail: ydsh_chsh@163.com

(**YANG Dong-Sheng** Professor at the School of Public Management/Emergency Management, Jinan

University. His research interest covers theories and methods of command and control.)



王飞跃 中国科学院自动化研究所复杂系统管理与控制国家重点实验室研究员. 主要研究方向为智能系统和复杂系统的建模、分析与控制.

E-mail: feiyue.wang@ia.ac.cn

(**WANG Fei-Yue** Professor at the State Key Laboratory for Manage-

ment and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interest covers modeling, analysis, and control of intelligent systems and complex systems.)