

CoPAL: 共纠正 P 机器人 A 行动 L 使用大型语言模型

Frank Joublin[✉], Antonello Ceravola[✉], Pavel Smirnov[✉], Felix Ocker[✉], Joerg Deigmoeller[✉],
Anna Belardinelli[✉], Chao Wang[✉], Stephan Hasler[✉], Daniel Tanneberg[✉], and Michael Gienger[✉]

Abstract—在追求能够接管传统上由人类执行的任务的完全自主机器人系统的进程中，开放世界的环境复杂性提出了相当大的挑战。为了解决这一紧迫问题，本研究对应用于机器人任务和运动规划的大语言模型 (LLMs) 领域做出了贡献。我们提出了一种系统架构，协调多个认知层次之间的无缝交互，包括推理、规划和运动生成。其核心是一种新的重规划策略，能够处理生成计划中的物理基础的、逻辑的和语义的错误。我们通过模拟环境和两个复杂的现实世界场景中进行实证评估，展示了所提出的反馈架构的有效性及其对可执行性、正确性和时间复杂度的影响：积木世界、调酒师和比萨制作。

索引术语 – 大型语言模型，机器人技术，规划

I. 介绍

经典规划技术 [1]–[3] 专注于搜索最优任务计划，将自然语言理解和低级运动规划问题排除在外。综合任务和运动规划 (TAMP) 方法 [4] 力求将高级推理与真实机器人的运动规划相结合，但仍可能面临不确定性和处理失败反馈的挑战。然而，智能实体代理需要能够适应并从不同类型的错误中恢复。因此，大型语言模型在机器人领域 [5]–[14] 中被越来越多地使用，因为它们提供了非常丰富的常识知识和隐式推理能力 [15]–[17]。进展和日益复杂的应用场景令人印象深刻地展示了这一点 [18]–[22]。特别是在处理信息不完整的情况下，LLMs 是实现纠正行为的有前途工具，因为它们可以利用其内在的世界知识来提供替代解决方案。及时重新评估计划并响应可能未预期的环境变化是对智能机器人应对各种情况的关键要求。利用大语言模型生成不同行动方案建议的能力，并考虑到之前的经历（以提示更新的形式），对于这样的反馈生成和使用似乎是一种合理的方法 [23]：诀窍在于认识到大语言模型正在生成可能的答案，这些答案需要由外部求解器来检查/完善，.. 本文通过提出一个分层架构来解决涉及成功/失败检查的机器人操作任务，从而在这



Fig. 1. 比萨饼领域任务在真实机器人上的执行快照以及所提出的框架的说明性描述。

方面做出了两项贡献：

- 一种具有多级反馈环的新型闭环任务规划机制 (CoPAL)。
- 对规划机制的评估，展示了不同类型的低级反馈如何在三种不同的场景中（包括仿真和真实机器人）提高规划和执行的质量。

II. 相关工作

使用 LLMs 将高层次用户目标转化为中层次动作序列（机器人可用的技能或动作基元）的想法最近在 [24]–[27] 中进行了研究。LLMs 的纯规划能力 [28], [29]、回溯提示 [30], [31] 和纠正性重提示策略 [32] 也得到了研究。虽然这些研究表明如何有效地利用 LLMs 进行规划，但它们并没有提供一个考虑到环境几何配置和任务执行反馈的全层次闭环规划架构。在这里，我们旨在解决这个差距。下面我们将回顾一些关于闭环规划机制的研究，特别是在处理反馈方面的进展。

Ichter 等人提出了 SayCan [33]——一种通过捕获可能技能对于实现高层次目标和从当前状态成功执行的概率来将大语言模型接地到能力函数中的方法。该方法依赖于通过强化学习训练技能及其策略，这在像 GPT-4 这样的大型语言模型时代可以被视为一个昂贵的步骤。作者指出 SayCan 只在当前决策步接收环境反馈，但如果某个技能失败或环境意外变化，则无法

[✉] Honda Research Institute Europe, Offenbach, Germany
{firstname.lastname}@honda-ri.de

Supplementary material: <https://hri-eu.github.io/Loom/>.

获得反馈。从这个意义上说，在规划循环中从我们的仿真系统获取的反馈能够及时触发纠正措施来处理规划过程中失败的技能。

黄等人提出了 InnerMonologue [34]——一项研究，其中作者探讨了 LLM 规划器在多大程度上能够利用环境反馈，如成功/失败、物体检测、场景描述、视觉问答和人类反馈。他们的方法与我们在将各种类型的反馈纳入规划循环方面相当，然而，该框架没有纳入低级几何约束和来自运动规划系统的反馈，这两者对于真实机器人的规划都是关键方面。

林等人指出 SayCan 和 InnerMonologue 在生成最优长期计划方面的局限性 [35]。在那里，提出了 Text2Motion 方法，该方法利用大语言模型预测符号目标状态，然后应用两种规划策略中的一种。第一种是提示大语言模型生成一个被认为是根据某个特定目标而最优的计划。返回计划中的每个技能（中级任务）都应通过几何可行性检查（从当前状态执行的概率），否则将被排除在计划之外。如果没有剩余技能且计划为空，则采用贪婪迭代策略（类似于 SayCan 中的一种）作为备选解决方案。与 Text2Motion 不同，我们的方法处理不可行的技能而不是将其排除在计划外。我们使用一个提供关于计划中失败步骤反馈的模拟器作为几何可行性检查器。更多细节见 Section III。

与我们的方法接近的是 Rana 等人提出的 SayPlan [36]。它依赖于对 3D 场景图的搜索，为 LLMs 提供找到的子图，并基于由场景图模拟器生成的反馈开始迭代重规划。相比之下，作者侧重于语义搜索和推理，这些是在 LLM 之外进行的，忽略了关于反馈的细节。

总结来说，大多数相关工作集中在解决符号规划的挑战上，而只有少数工作致力于将计划传播到物理世界。据我们所知，没有深入分析如何利用由此产生的低级反馈来解决问题。在这里，我们通过提出一个分层重规划架构来解决这一缺口，该架构针对长时间范围的机器人操作任务，收集从所有层级直至运动规划层面的反馈。

III. 校正规划概念

所提出的系统称为 CoPAL，是专门为机器人规划中可用的信息类型设计的，特别是自然语言指令以及语义、符号和感官反馈。我们发现将规划问题分解为四个专门层（见 Figure 2）具有显著优势。这种分解

使得能够定义各模块清晰的角色，这对基于 LLM 的代理特别有益。这些代理可以使用明确的系统消息实例化，从而实现良好的封装。封装的好处不仅是软件工程范式，而且也有相关工作 [37] 和我们自己的实验的支持。系统的前两层包括三个基于 LLM 的会话代理，利用世界知识将不完整和未指定的人类指令转化为长远的操作计划。第三和第四层负责调度、预测、决策以及将操作计划与机器人的物理能力和 3D 环境联系起来。最低层更新环境状态并使其对所有上层可用。它包含场景的几何表示，包括机器人坐标、可用对象以及其他相关信息如父子层次结构。各层将在以下部分进行更详细的描述。

A. 用户交互层

用户交互层作为人类用户与系统其余部分之间的中介。第一个大型语言模型代理，我们称之为亚历克斯，被提示充当人类请求的自然语言解释器，并为不需要任何物理响应的请求提供直接答案。否则，将请求转发给下一个大型语言模型代理。亚历克斯是唯一使用对话历史记录代理，这使它能够根据原始的人类请求传达机器人反馈。它还维护执行动作的历史记录以丰富其对人类的回应。

B. 高级规划器

第二个大型语言模型代理，称为特拉维，负责将亚历克斯的请求翻译成包含详细规范的高阶计划供下一个大型语言模型代理使用。它被提示以链式思维 (CoT) 方式生成以下信息：人类请求的目标，达成该目标所需对象的列表（从环境状态中提取），当前环境状态的总结，以及需要执行的剩余步骤。这些连续步骤构成了用描述性语言编写的更广泛的计划。迫使大型语言模型输出所有这些信息有助于其生成有意义的高阶计划。另一个功能是将特拉维整合来自较低层的反馈到规划阶段，并根据必要调整或重新制定计划。向大型语言模型关于环境变化提供自然语言反馈称为回传提示，正如在 Section II 中所讨论的。回传提示的实施已在诸如积木世界问题等规划挑战方面展示了显著改进，我们将在 Section IV-B 中进一步探讨。

第三个大型语言模型代理，称为衣物，接收特拉维的高层次计划规范并从中生成一个高层次计划。这包括将总体目标转化为较低层级可以处理的动作。对于被调查的场景，我们使用了 Table I 中指定的操作

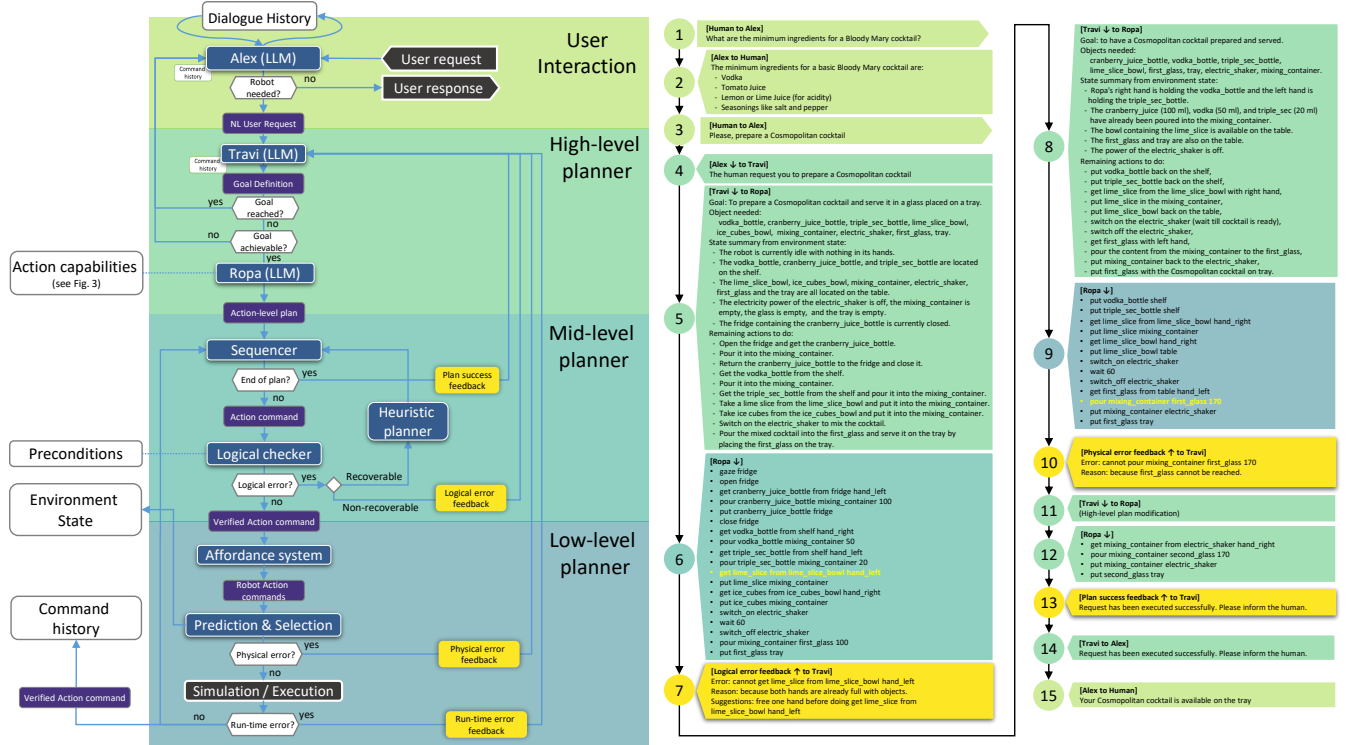


Fig. 2. 左：系统架构。计算模块用蓝色表示，自上而下的指令用绿色表示，自下而上的反馈用橙色表示，与现实世界交互的模块用深绿色表示。右：两个示例人类请求的执行流程（步骤 1 和 3）。

TABLE I
机器人动作能力*

命令	描述
get <object1> from <object2> (<hand>)	take object1 from object2 using hand
put <object1> <object2>	put object1 (in hand) on/in object2
pour <object1> <object2> <x [ml]>	pour x milliliter (liquid/solid) from object1 (in hand) into object2
open_door <object1>	open door of object1 (microwave)
close_door <object1>	close door of object1 (microwave)
screw <object1>	close object1 (bottle) by screwing tap
unscrew <object1>	open object1 (bottle) by unscrewing tap
finger_push <object1>	push object1 with fingertip (to power on/off devices)
wait <x [s]>	wait for x seconds

* 动作命令：动作级别的计划（图 2 高级和中级规划者之间）由这些动作的序列组成。单个动作通过低级规划器由机器人执行。

命令。如果无法生成计划（即目标状态已经实现或机器人无法达到），则会向亚历克斯发送反馈消息以通知人类。重新规划周期的数量限制为固定数量（在我们的实验中是 5 次）尝试，之后将中断规划过程并通知人类。

C. 中层规划者

中级规划器需要补偿来自高层的潜在不完整、错误或冗余的命令序列。它依次处理计划中的每个动作（因此被命名为排序器）。一个逻辑检查器根据语法、对象名称和前提条件验证每个动作指令。考虑了两种类型的错误：可恢复的（见 Table II）和不可恢复的错误（见 Table III）。可恢复的错误通过修改中级计划来处理，方法是插入或替换命令，由启发式规划器完成。目前使用基于规则的算法进行此步骤。不可恢复的错误导致在高级规划层中重新规划。

D. 低级规划器

该规划层次结构的最低层的作用是弥合如 Table I 所示的文字动作指令与其在考虑三维场景、物体物理属性以及机器人操作能力的动作生成中的基础之间的差距。它包含一个提供系统，将命令实例化为一组实现动作目标的潜在原语。例如，“获取瓶子”这一动作可以被双臂机器人通过四种基本动作作用来实例化：用左手或右手以顶部抓取或强力抓取的方式拿住瓶子。每个原语在模拟中根据其可行性进行评估，并根据碰撞和关节限制指标 [38] 进行排序。选择获胜的原语在

模拟中执行。如果所有原语都失败，则会传达最佳原语失败的原因。此层级上的反馈与物理错误和运行时错误有关。物理错误包括如不可达物体、机器人与环境之间的碰撞、违反机器人限制等问题。运行时错误涉及例如硬件故障或其他未建模的问题，并且仅在使用真实机器人操作系统时发挥作用。假设在此层面上产生的错误不易恢复。因此，这些错误会被传达给高层规划器，在那里它们通常会导致计划变更。

E. 反馈循环

当用户提出一个知识性问题（例如 Figure 2 的第 1 步）时，亚历克斯以类似于聊天机器人的自然语言形式提供答案。我们认为这是一个知识反馈，我们不再进一步评估它。另一方面，一旦系统被要求执行一项物理任务（例如，调制鸡尾酒，第 3 步），一系列规划代理（特拉维，衣物）将依次触发，直到产生第一个错误，这可能在不同的层级发生，如前几小节所述。当中级可恢复错误由启发式规划器内部处理（产生中级计划修订）时，不可恢复的中级错误和物理错误则需要高级重新规划，因此会向特拉维发送反馈消息以进行重新提示。逻辑错误反馈，物理误差反馈，和运行时错误反馈被公式化为三元组 $\langle E, R, S \rangle$ ，在一个字符串中：“错误：〈哪个命令失败〉，原因：〈失败的原因是什么〉，建议：〈如何解决问题〉。”

IV. 实验

为了评估所提出的架构，我们通过 API 使用了 OpenAI 的 GPT-4 进行了若干实验。我们扩展了 barman 规划问题 [29]，进行了准备比萨饼的实验，并对积木世界问题进行了实验。

TABLE II
可恢复错误的示例

命令	失败	中层计划修改
put object1 object2	object1 is not in hand	free one hand if needed, then take object1, put object1 object2 and possibly get previous object
put object1 object2	object2 is in closed container	free one hand if needed, then open container, put object1 object2, close container and possibly get previous object
open_door object1	all hands are full	free one hand, open_door object1 and possibly get previous object

TABLE III
不可恢复错误的示例

类型	错误示例
Syntactic/ Semantic Errors	Wrong command syntax Wrong object names
Logical Errors	Getting object with all hands full Pouring into an object placed in a container Pouring into a full object
Physical Errors	Joint limits errors Obstacle object on the way Object out of reach

TABLE IV
系统配置

符号	描述
BL	Baseline, neither mid-level nor high-level replanning
M	Only mid-level planner for replanning
H ₀	Only high-level planner for replanning Feedback: what (cf. Section III-E)
H ₁	Only high-level planner for replanning Feedback: what + why (cf. Section III-E)
H ₂	Only high-level planner for replanning Feedback: what + why + how (cf. Section III-E)
MH _i	Combination of M and H _i

A. 巴曼实验

在此场景中，基于大语言模型的机器人识别必要的食材，制定计划，相应执行，并在必要时重新规划。为了评估目的，我们选择了食谱作为调酒师和比萨实验的真实基准计划。我们确定了十个感兴趣的鸡尾酒：Bloody Mary、Caipirinha、Cosmopolitan、Daiquiri、Gin and Tonic、Long Island Iced Tea、Manhattan、Margarita、Martini 和 Mojito。对于这些，我们指定了食材，并区分液体、固体和可选的食材。完整的列表包括：苦精、可乐、蔓越莓汁、琴酒、冰块、柠檬片、青柠片、薄荷、橙片、朗姆酒、盐、糖浆、切片柠檬、苏打水、方糖、龙舌兰酒、番茄汁、汤力水、三重 sec、苦艾酒、伏特加和威士忌。例如，Listing 1 显示了 Cosmopolitan 预期的食材。

```
"02.03 Cosmopolitan": {
  "ingredient": ["lime_slice"],
  "liquid": ["cranberry_juice", "triple_sec", "vodka"],
  "optional": ["ice_cubes"]
},
```

Listing 1. Ground truth ingredients for a Cosmopolitan.

为了评估反馈类型和不同重新规划水平的影响，我们考虑了几种系统配置，定义见 Table IV。

为了在厨房布局中允许创造性规划，我们使用了

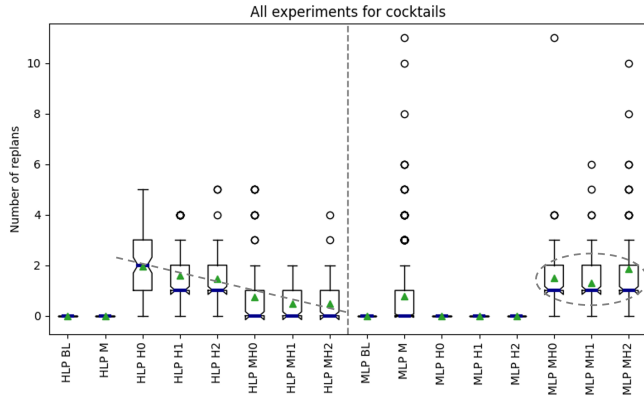


Fig. 3. 高级别（HLP，左侧）和中级别（MLP，右侧）重新规划的数量按设置划分。

OpenAI 的默认温度为 0.8。对于每种鸡尾酒和每次通过高级规划器进行重新规划的配置，我们都进行了 10 次运行，总共得到 600 个试验。我们根据可用数据计算基线（BL）和仅使用中级规划器（M）进行重新规划的结果，将所有需要高级重新规划的计划视为不可执行。

Figure 3 显示了各种设置的重新规划次数，区分高级和中级重新规划。根据定义，在基线中没有任何重新规划，并且对于 H_1 变体没有使用中级规划器进行重新规划。很明显，改进反馈质量减少了必要的高级重新规划次数，请参见左图中虚线标出的趋势 Figure 3。此外，对可恢复错误使用中级重新规划（MLP 对于 MH_1 ）减少了必要的高级重新规划次数（HLP 对于 MH_1 ）。

我们通过考虑计划的可执行性和生成这些计划所需的运行时间来评估架构变化的表现，参见 Figure 4。在这里，如果机器人没有陷入超过五个重规划步骤（由高级规划器完成）的循环，则认为该计划是可执行的。在没有任何重规划（BL）的情况下，大语言模型仅实现 5.7 % 的可执行性。而使用中级规划器进行重规划（M）将可执行性提高到 34.8 %，使用高级规划器，例如 MH_1 ，总是产生可执行的计划。请注意，更详细的反馈也积极影响可执行性（ MH_1 与 MH_0 相比），并且通过中级规划器恢复错误即使反馈质量有限也能提高可执行性（ MH_0 与 H_0 相比）。

仅考虑运行时间指标，分析受到所需规划量的强烈影响（GPT3.5 的亚历克斯持续时间为 $0.6 \pm 0.1s$ ，GPT4 的特拉维持续时间为 $16.1 \pm 5.1s$ ，衣物持续时间为 $14.9 \pm 3.3s$ ）：生成计划的时间复杂度比启发式中级规划器高出大约 1 到 2 个数量级。如 Figure 4 所

示，基线和中级规划器特别快，因为它们只需要提示大语言模型一次即可生成计划。至关重要的是，利用高效的中级规划器与高级规划器结合（ MH_1 ）比仅使用高级规划器（ H_1 ）要快得多。请注意，反馈的质量也对运行时间有正面影响（ MH_2 对比 MH_0 ）。

我们将计划正确性视为另一个性能指标。在调酒师场景中，正确性定义为遵循给定的计划，即在托盘上放置恰好一个杯子，并包括所有必需的成分但不包含任何非可选成分。可以执行的过程的平均正确率为 87%，涵盖了所有架构类型，表明在没有最终来自人类的满意度反馈的情况下，正确性独立于可执行性。总体而言，所提出的架构表现非常出色，考虑到用户并未指定食谱，而是 LLM 从其世界知识中识别出相关成分。

我们使用基于错误成分数量的距离度量进行了深入分析，参见 Equation (1)。缺少的成分，例如没有三秒橙酒的柯斯莫波尔，被认为比多余的成分更糟糕，例如含有罗勒的血腥玛丽，并且分别进行加权（1.0 对比 .2）。该分析显示，5% 的错误鸡尾酒是由缺少的成分引起的（ $d = 1$ ），而 5% 是由于不允许的多余成分引起的（ $d = .2$ ）。多重错误的数量显著较低，大约 1% 是两个缺失成分，以及一个缺失和一个多余成分的情况。

$$d = \sum_{e \in E} w_s \cdot s_e + w_m \cdot m_e \quad (1)$$

其中：

E = experiments

w_s = weighting factor for superfluous ingredients (.2)

s = number of superfluous ingredients

w_m = weighting factor for missing ingredients (1.0)

m = number of missing ingredients

B. 积木世界实验

在第二个实验中，我们使用了来自 [29] 的数据集进行了一个高级积木堆叠实验，该实验的特点是给定的积木目标配置。这个众所周知的积木世界问题由于目标配置中所需的积木顺序而具有挑战性。

本实验展示了所提出方法的泛化能力。我们没有采用通常由使用 PDDL [39], [40] 的经典问题求解器所采用的任何预条件模型。相反，我们的方法依赖于系统的底层部分，这使得它能够转移到其他问题领域。安置约束由所使用的可用性模型来考虑。像在已被占用

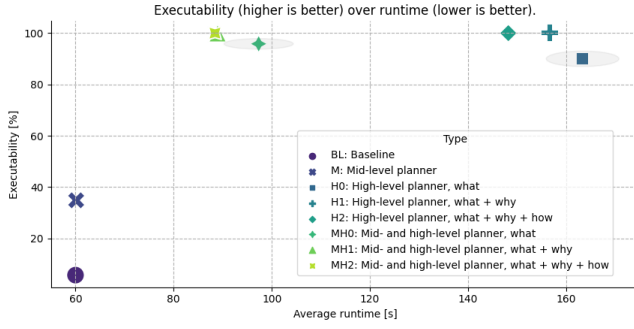


Fig. 4. 关于可执行性和运行时间的架构变体比较：在中高级别上重新规划的架构实现了运行时间和可执行性之间的最佳平衡。

TABLE V

成功率在积木世界实验中

Success rate		Description of the method
3 块	5 块	
<i>CoPAL (Ours)</i>		Backprompt after fully executed plan and evaluate goal using GPT-4
83%	73%	
<i>LLM+P</i>		Classical planner combined with GPT-3 [29]
33%	46%	
<i>LLM-As-P</i>		Plain GPT-3 as planner [29]
0%	0%	

的位置堆叠这样的无效配置通过碰撞检测识别出来，并作为物理错误反馈给高层规划器。块配置的状态从环境状态翻译成自然语言以评估是否实现了期望的目标配置。这一步骤在高层规划层中执行。

Table V 总结了我们在三块积木（左列）和最多 5 块积木（3、4 和 5 块积木）的实验结果。我们在不超过 4 块积木的情况下达到了与 82% 类似的性能，正如在 [30] 中报道的一样。这标志着相较于 [29] 有显著提升，可能是因为我们使用了 CoT、回溯提示和更详细的任务描述。另一方面，在 5 块积木的情况下表现大幅下降。这表明对于短期逻辑问题，将大语言模型与详细且紧密的分步反馈相结合能带来非常好的结果。而对于复杂的规划问题，如 5 块及以上积木的情况，我们实证观察到大语言模型并不擅长解决此类任务。我们认为大语言模型规划适用于不完全是算法性任务，其中常识知识可以被使用，正如在接下来的实验中所展示的一样。在这种情况下，规划范围可以很长。

C. 闭环机器人实验

依照之前的积木世界实验，我们为比萨场景创建了真实基准，包括相应的配料，如面团、番茄酱和蘑菇等，类似于调酒师场景，并运行了几项实验。该场景配置为较大的物体阻挡了食物项目的可达性，需要重



Fig. 5. 带有碰撞和低级反馈的实验摘录。

新规划以移除障碍物。Figure 5 显示了一个实验的部分内容，在此部分中低级规划器检测到两次碰撞，并由大语言模型提供了相应的重新规划。

图 5 描述了低级反馈及其与高级规划层的相互作用，最终通过将遮挡的盐放在架子上（步骤 4 和 5），并将番茄酱瓶收起来（步骤 8）来利用左手拿起黑橄榄。我们发现这是一种非常显著的自发问题解决行为。

实验还使用了一台真实的机器人（见 Figure 1）。我们为真实机器人使用的材料选择了软性材料。整个架构可以轻松地在笔记本电脑上运行。一些实验运行的视频可在补充材料中找到。

V. 总结与展望

本文的核心是一种新颖的基于大语言模型的重规划策略，该策略利用多层次反馈与物理机器人在闭环中进行操作。我们在三种不同场景下展示了规划机制的表现，既包括模拟环境也包括真实机器人。首先，使用大语言模型进行高层次重规划显著提高了可执行性。其次，特别是来自感知层面的低层次反馈的质量改善了生成计划的可执行性和执行时间。第三，引入一种快速、基于启发式的中间层规划来解决传统动作前提条件错误对运行时间产生了积极影响。最后，本文展示了基于大语言模型的机器人在开放世界中行动的潜力，并能够适当应对不可预见的问题。

扩展这些发现，未来的工作应包括关于可解释性的研究，例如通过结合基于 LLM 的方法来获得直观反应和 GOFAI 进行最优规划。

当我们深入研究大语言模型时，提示设计显得至关重要。我们的方法减少了领域中心的偏见，并清晰定义了每个大语言模型代理的角色（请参阅 <https://hri-eu.github.io/Loom/> 以获取所有提示的描述）。反向提示发挥了重要作用，尽管反馈中的建议部分提供的性能改进很小，但我们注意到来自低级规划者的反馈具有质的敏感性，在这种情况下，大语言模型需要根据情况变得更加有创意（障碍、远距离物体、工具需求等）。

未来的研究方向可能包括增强我们的系统，以包含能够提取示例提示的机制，正如 [41] 所暗示的研究。

尽管取得了成就，我们注意到与 GPT-4 API 相关的延迟仍然是一个关注点。虽然 OpenAI API 的推理时间正在稳步减少，在机器人应用中处理这一问题的一种方法是引入语音和行为填充物，使机器人在 API 查询期间表现出行为或进行交流。然而，在正在进行的有效步骤中启动重新规划，如中级计划器所观察到的那样，可以减少由延迟引起的延迟。为了进一步缓解与大语言模型相关的延迟，一个有希望的方向是增加内存使用并从过去的规划场景中学习。

最后，系统应扩展以实现与人类的安全合作，即 在合作智能意义上的安全合作 [42]。

REFERENCES

- [1] E. Karpas and D. Magazzeni, “Automated planning for robotics,” *Annual Review of Control, Robotics, and Autonomous Systems*, 2020.
- [2] R. I. Brafman, D. Tolpin, and O. Wertheim, “Probabilistic programs as an action description language,” *AAAI Conference on Artificial Intelligence*, 2023.
- [3] D. Tanneberg and M. Gienger, “Learning type-generalized actions for symbolic planning,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.
- [4] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, “Integrated task and motion planning,” *Annual Review of Control, Robotics, and Autonomous Systems*, 2021.
- [5] S. Vemprala, R. Bonatti, A. Buckner, and A. Kapoor, “ChatGPT for robotics: Design principles and model abilities,” Microsoft, Tech. Rep., 2023.
- [6] N. Wake, A. Kanehira, K. Sasabuchi, J. Takamatsu, and K. Ikeuchi, “ChatGPT empowered long-step robot control in various environments: A case application,” *IEEE Access*, 2023.
- [7] G. DeepMind, “Demonstrating Large Language Models on robots,” *Robotics: Science and Systems (RSS) Demo Track*, 2023.
- [8] T. Yoneda, J. Fang, P. Li, H. Zhang, T. Jiang, S. Lin, B. Picker, D. Yunis, H. Mei, and M. R. Walter, “Statler: State-maintaining language models for embodied reasoning,” *arXiv preprint arXiv:2306.17840*, 2023.
- [9] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, “Tidybot: Personalized robot assistance with large language models,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.
- [10] M. Gramopadhye and D. Szafr, “Generating executable action plans with environmentally-aware language models,” *arXiv preprint arXiv:2210.04964*, 2023.
- [11] Y. Ding, X. Zhang, S. Amir, N. Cao, H. Yang, A. Kaminski, C. Esselink, and S. Zhang, “Integrating action knowledge and LLMs for task planning and situation handling in open worlds,” *Autonomous Robots (2023)*, 2023.
- [12] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mor-datch, and P. Florence, “Palm-e: An embodied multimodal language model,” *International Conference on Machine Learning*, 2023.
- [13] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, “Code as policies: Language model programs for embodied control,” *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
- [14] Y. Ding, X. Zhang, C. Paxton, and S. Zhang, “Task and motion planning with large language models for object rearrangement,” *arXiv preprint arXiv:2303.06247*, 2023.
- [15] Z. Zhao, W. S. Lee, and D. Hsu, “Large Language Models as commonsense knowledge for large-scale task planning,” *arXiv preprint arXiv:2305.14078*, 2023.
- [16] F. Ocker, J. Deigmöller, and J. Eggert, “Exploring Large Language Models as a source of common-sense knowledge for robots,” in *International Semantic Web Conference*, 2023.
- [17] Y. Ding, X. Zhang, C. Paxton, and S. Zhang, “Leveraging commonsense knowledge from large language models for task and motion planning,” *RSS 2023 Workshop on Learning for Task and Motion Planning*, 2023.
- [18] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, 2020.
- [19] S. Li, X. Puig, C. Paxton, Y. Du, C. Wang, L. Fan, T. Chen, D.-A. Huang, E. Akyürek, A. Anandkumar *et al.*, “Pre-trained language models for interactive decision-making,” *Advances in Neural Information Processing Systems*, 2022.
- [20] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, B. Yin, and X. Hu, “Harnessing the power of LLMs in practice: A survey on ChatGPT and beyond,” *arXiv preprint arXiv:2304.13712*, 2023.
- [21] F. Joubin, A. Ceravola, J. Deigoeller, M. Gienger, M. Franzius, and J. Eggert, “A glimpse in ChatGPT capabilities and its impact for AI research,” *arXiv preprint arXiv:2305.06087*, 2023.
- [22] S. Yang, O. Nachum, Y. Du, J. Wei, P. Abbeel, and D. Schuurmans, “Foundation models for decision making: Problems, methods, and opportunities,” *arXiv preprint arXiv:2303.04129*, 2023.

- [23] S. Kambhampati, “Can LLMs really reason and plan?” *Blog at Communications of the ACM*, 2023. [Online]. Available: <https://cacm.acm.org/blogs/blog-cacm/276268-can-llms-really-reason-and-plan/fulltext>
- [24] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, “Language models as zero-shot planners: Extracting actionable knowledge for embodied agents,” *International Conference on Machine Learning*, 2022.
- [25] Y. Xie, C. Yu, T. Zhu, J. Bai, Z. Gong, and H. Soh, “Translating natural language to planning goals with Large-language models,” *arXiv preprint arXiv:2302.05128*, 2023.
- [26] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, “Progprompt: Generating situated robot task plans using Large Language Models,” *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
- [27] Z. Wang, S. Cai, A. Liu, X. Ma, and Y. Liang, “Describe, explain, plan and select: Interactive planning with Large Language Models enables open-world multi-task agents,” *arXiv preprint arXiv:2302.01560*, 2023.
- [28] K. Valmeekam, A. Olmo, S. Sreedharan, and S. Kambhampati, “Large Language Models still can’t plan (A Benchmark for LLMs on Planning and Reasoning about Change),” *arXiv preprint arXiv:2206.10498*, 2022.
- [29] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, and P. Stone, “LLM+P: Empowering Large Language Models with optimal planning proficiency,” *arXiv preprint arXiv:2304.11477*, 2023.
- [30] K. Valmeekam, M. Marquez, S. Sreedharan, and S. Kambhampati, “On the planning abilities of Large Language Models – A critical investigation,” *arXiv preprint arXiv:2305.15771*, 2023.
- [31] L. Guan, K. Valmeekam, S. Sreedharan, and S. Kambhampati, “Leveraging pre-trained Large Language Models to construct and utilize world models for model-based task planning,” *arXiv preprint arXiv:2305.14909*, 2023.
- [32] S. S. Raman, V. Cohen, E. Rosen, I. Idrees, D. Paulius, and S. Tellex, “Planning with Large Language Models via corrective re-prompting,” *arXiv preprint arXiv:2211.09935*, 2022.
- [33] B. Ichter, A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho *et al.*, “Do as I can, not as I say: Grounding language in robotic affordances,” *Conference on Robot Learning*, 2023.
- [34] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar *et al.*, “Inner Monologue: Embodied reasoning through planning with language models,” *arXiv preprint arXiv:2207.05608*, 2022.
- [35] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg, “Text2motion: From natural language instructions to feasible plans,” *arXiv preprint arXiv:2303.12153*, 2023.
- [36] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, “SayPlan: Grounding Large Language Models using 3D scene graphs for scalable task planning,” *arXiv preprint arXiv:2307.06135*, 2023.
- [37] S. Patra, J. Mason, M. Ghallab, D. Nau, and P. Traverso, “Deliberative acting, planning and learning with hierarchical operational models,” *Artificial Intelligence*, vol. 299, p. 103523, 2021.
- [38] M. Gienger, M. Toussaint, and C. Goerick, “Whole-body motion planning—building blocks for intelligent systems,” *Motion Planning for Humanoid Robots*, p. 67, 2010.
- [39] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins, “PDDL - The Planning Domain Definition Language,” *The AIPS-98 Planning Competition Committee*, 1998.
- [40] P. Haslum, N. Lipovetzky, D. Magazzeni, C. Muise, R. Brachman, F. Rossi, and P. Stone, *An introduction to the Planning Domain Definition Language*. Springer, 2019, vol. 13.
- [41] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar, “Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking,” *arXiv preprint arXiv:2309.01918*, 2023.
- [42] B. Sendhoff and H. Wersing, “Cooperative intelligence-a humane perspective,” in *2020 IEEE International Conference on Human-Machine Systems (ICHMS)*. IEEE, 2020, pp. 1–6.