

关于基于 LLM 的具身智能代理 在智能游戏中的应用研究

刘帅

2020212267

2024 年 1 月 8 日

摘要

随着以 GPT-4[3] 为代表的大语言模型 (LLMs) 的发展, 提升了人工智能对于世界信息获取、知识泛化和推理的能力, 进一步为发展人工智能代理 (AI Agent) 带来了新的方向和研究思路, 并取得了显著的进展 [17][26][3]。以 LLM 为基础的代理通常包括三个关键部分: **大脑、感知和行动**。针对大脑, 它主要由一个大语言模型组成, 负责存储了关键的记忆、信息和知识, 并承担着信息处理、决策、推理和规划等重要任务, 是决定代理能否展现智能行为的关键因素。对于感知模块, 它类似于人类的感觉器官, 主要功能是将代理的感知空间从仅限文本扩展到包括文本、声音、视觉、触觉、嗅觉的多模态感知空间, 这种扩展使代理能够更好地感知来自外部环境的信息, 对于一个“看不见”的 LLM 来说, 这部分需要通过游戏环境的反馈信号将知识传递给“大脑”。对于行动模块, 用于扩展代理的行动空间, 从而使代理能够具备文本输出、采取具体行动并使用工具, 以便更好地对环境做出响应、提供反馈, 甚至改变和塑造环境。本文将以 Minecraft 的游戏环境作为例子, 结合书中 [29] 第六章 Game AI Panorama 和第七章 Frontiers of Game AI Research 的内容, 对 LLM-based Agent 的主要构成、技术特点、目前局限及发展趋势等方面进行分析。

游戏代理的组成结构

在这个章节, 我们将针对构成 AI Agent 的结构进行简要介绍。如图所示 3, AI agent 主要分为信息、记忆、规划和执行四部分内容。

其中, 信息的部分是 AI Agent 执行的先验, 针对 LLM 的情况, 由于 LLM 经由大量数据训练, 因此囊括绝大多数泛化的世界信息, 从而能够作为合适的提供信息的大脑。

针对记忆部分, 通常包含两方面的因素: 1. 关于游戏环境目前状态的短期存储: 通常将一些重要信息作为 prompt 输入, 从而进行下一轮的任务规划、执行。2. 关于一些较大规模信息的长期存储: 由于 LLM 已经具备了关于泛化世界信息的存储, 我们这一步的目的是存储更多针对游戏环境的特定信息, 例如在 Minecraft 游戏中, STEVE-EYE[5] 为了训练一个多模态大模型, 除了利用 GPT 生成指令-回答对以外, 特定的针对 Minecraft 信息, 例如关于游戏内物品的文本介绍、当前动作的简要描述等, 利用 Minedojo[9] 对 minecraft 的数据进行了存储和训练。Voyager[24] 为了完成挖钻石的任务, 利用 GPT4 构建了执行任务的技能数据库。

针对任务规划方面, 在一篇综述 [25] 中提到, 任务规划的结构如图 2 所示。对于单路径的规划而言, 为 LLM 输入对应的 prompt 和最终的任务要求, 则会在一次推理以内将任务分解成子任务。对于 re-prompting 的范式, 则会在每一步子任务的生成和执行后重新进行一次 query, 也就是说, 它可以及时的改变来自环境的反馈并确保任务能够正确执行, 目前大部分基于具身智能的智能代理都采用了该范式进行任务的规划和执行。针对多种路径的规划范式, 则是通过多条路径的树状结构, 一次性给出多种解决方案, 从而并行的进行任务规划。由于这种范式很容易造成空间和时间的开销, 以及, 以游戏为例的环境存在的占用显存或者进程冲突的情况, 在游戏的代理应用方面并不十分常见。同时, 为了将游戏环境和任务情况进行更为形式化的描述, PDDL[12][23] 中提到

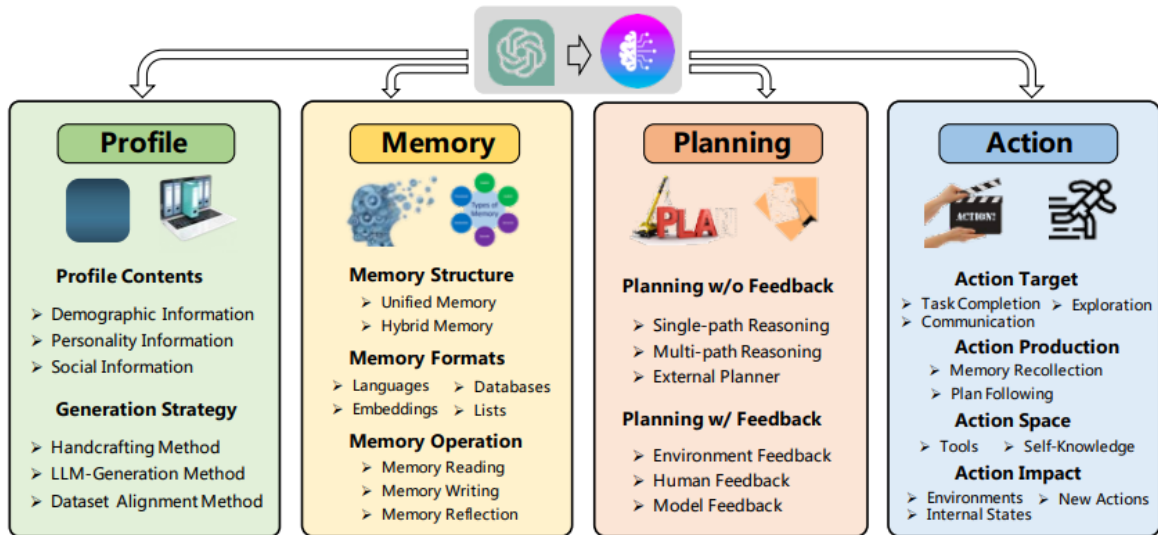


图 1: 基于 LLM 的代理的示意结构

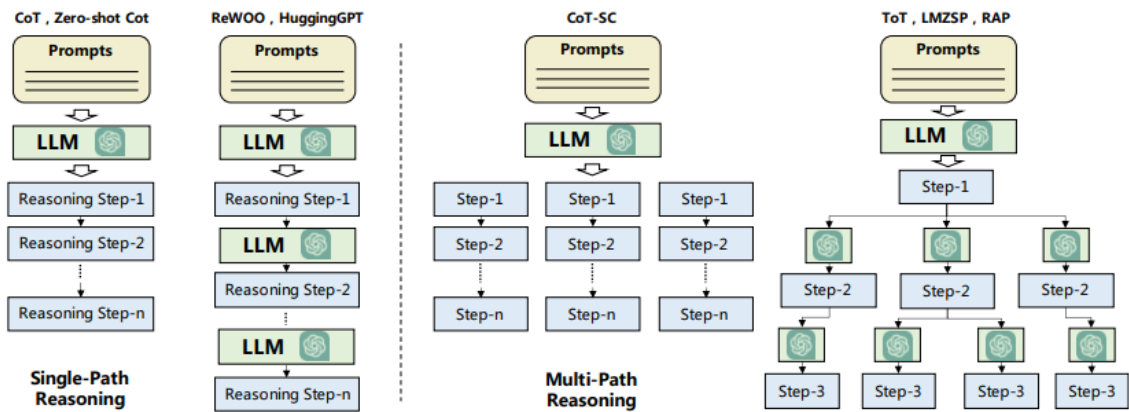


图 2: 基于 LLM 进行任务规划的结构示意

了一种更为形式化的描述，从而为 LLM 的任务规划提供了更结构化的先验，并为场景图生成 [30] (Scene Graph Generation) 等任务提供了充分的数据来源。

关于 Minecraft 的一些数据库示例

本章节将针对 minecraft 当中数据收集流程进行介绍。关于 Minecraft 数据库的存储和训练在 Minedojo[9] 有所提出，他的数据由 740k 条 YouTube 视频，7k 个维基页面及 350k 条 Reddit 帖子所构成的图像-文本对构成。在细节层面，Minedojo 利用类似于 CLIP[19] 的方法训练了 MINECLIP 模型，同时，MINECLIP 的 logits 由于直接输出了图像文本的匹配相似度，可以直接的作为 reward-model，从而参与 RLHF 和一些传统 RL 方法的 PPO 训练过程。具体来讲，MINECLIP 的构成如下。

Butcher Economic Trade					Butcher
Level	Item wanted	Default quantity	Item given	Quantity	
Novice	Raw Chicken	14	Emerald	1	
	Raw Porkchop	7	Emerald	1	
	Raw Rabbit	4	Emerald	1	
	Emerald	1	Rabbit Stew	1	
Apprentice	Coal	15	Emerald	1	
	Emerald	1	Cooked Porkchop	5	
	Emerald	1	Cooked Chicken	8	
Journeyman	Raw Mutton	7	Emerald	1	
	Raw Beef	10	Emerald	1	
Expert	Dried Kelp Block	10	Emerald	1	
Master	Sweet Berries	10	Emerald	1	

Product	Ingredient	Exp	Description
Copper Ingot	Copper Ore	0.7	Used to craft various items, including spyglasses, lightning rods, and copper blocks.
Iron Ingot	Iron Ore	0.7	Used to craft various items, including blast furnaces, anvils, iron blocks, iron nuggets, rails, buckets, cauldrons, chains, compasses, crossbows, flint-and-steels, heavy-weighted pressure plates, hoppers, iron trapdoors, minecarts, pistons, shears, shields, iron armor, iron tools, stonecutters and tripwire hooks.
Gold Ingot	Gold Ore	1	Used to craft various items, including netherite ingots, gold blocks, golden apples, gold nuggets, clocks, golden armor, golden tools, powered rails and light-weighted pressure plates. Also used as a currency for bartering.
Diamond	Diamond Ore	1	Used to craft various items, enchanting tables, jukeboxes and diamond blocks. When normally mined drops 1 diamond and 3-7.

Hostile mobs									
Blaze	Chicken Jockey	Creeper	Drowned	Elder Guardian	Endermite	Evoker	Ghast	Guardian	Hoglin
Husk	Magma Cube	Phantom	Piglin Brute	Pillager	Ravager	Shulker	Silverfish	Skeleton	Skeleton Horseman
Slime	Spider Jockey	Stray	Warden	Witch	Wither Skeleton	Zoglin	Zombie	Zombie Villager	

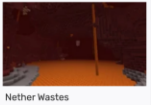
Biome name	Features	Description	Screenshot [hide]
Nether Wastes	Netherrack, Glowstone, Soul Sand, Nether Quartz Ore, Ghasts, Blazes, Zombified Piglins, Nether Fortresses, Wither Skeletons, Lava, Magma cubes, Gravel, Magma Blocks, Bastion Remnants, Ruined Portals, Piglins, Nether Gold Ore	Temperature: 2.0. Rainfall: 0.0. This is one of the biomes used to generate the Nether. Within this biome mobs such as ghasts, packs of piglins, zombified piglins and the occasional magma cubes and endermen spawn. Certain structures, such as Nether quartz ore and glowstone blobs, and Nether fortresses generate only in the Nether.	

图 3: Mindoyo 数据库信息汇总

分帧的图像编码器 ϕ_I : MINECLIP 利用 ViT-B/16 用来将 rgb 帧编码成 512 维的向量, 仅仅对于最后两层进行微调。

时序融合模块 ϕ_a : MINECLIP 采用了 average pooling 和自注意力两种方法来实现相邻帧之间的融合, 在融合之后, 利用了 CLIP Adapter[11] 进行 clip 的训练。

Text encoder ϕ_G : 利用 GPT model 进行文本信息编码, 同样只对最后两层的参数进行微调。

同样地, 在 STEVE-EYE[5] 中, 如图所示 4, LLM 的训练数据来源于多种数据类型, 例如指令对、QA 对、关于图片的简要概述、以及基于上下文的任务规划。Steve-Eye 在获取世界的基础知识、理解周围环境的细微差别以及生成可执行计划以完成各种开放性任务方面表现出色。此外, Steve-Eye 通过视觉或基于文本的提示来响应用户的指令, 提高了人机交互的便捷性和灵活性。

关于 LLM Agent 模型及训练架构分析

本章节将针对 LLM 以及视觉语言模型 (Vision Language Model, VLMs) 的主流模型及训练架构进行分析。关于以 LLM 作为大脑所驱动的智能代理最早出现在 Ghost[31] 的工作中, 具体而言, 图 6 包括 LLM 分解器、LLM 规划器和 LLM 与 minecraft 的仿真环境交互的接口, 它们分别负责将子目标、结构化操作和键盘操作进行分解。在 Minecraft 中给定一个目标, LLM 分解器首先根据从互联网收集的基于文本的知识将其分解成一系列明确定义的 subtasks。然后, LLM 规划器为每个子目标规划一系列结构化操作。这些结构化操作具有清晰的语义和相应的反馈, 使 LLM 能够理解周围环境并在认知层面做出决策。LLM 规划器还将成功的行动列表记录和总结到基于文本的存储器中, 以增强未来的规划能力。最后, LLM 界面执行这些结构化操作, 通过处理原始键盘输入并接收原始观察结果与环境互动。

在 Voyager[24] 的工作所构建的 agent 中同样包含了任务规划、子任务分解、memory 构建部分, 他们的研究重点集中在技能库 (skill library) 的构建上, 利用该工作的先前工作 CLIP 作为外部的知识模型。可以在 Minecraft 中进行探索、掌握各种技能, 并不断进行新的发现, 无需人类干预。该工作, 如图 5 通过三个关键模块实现: 1)

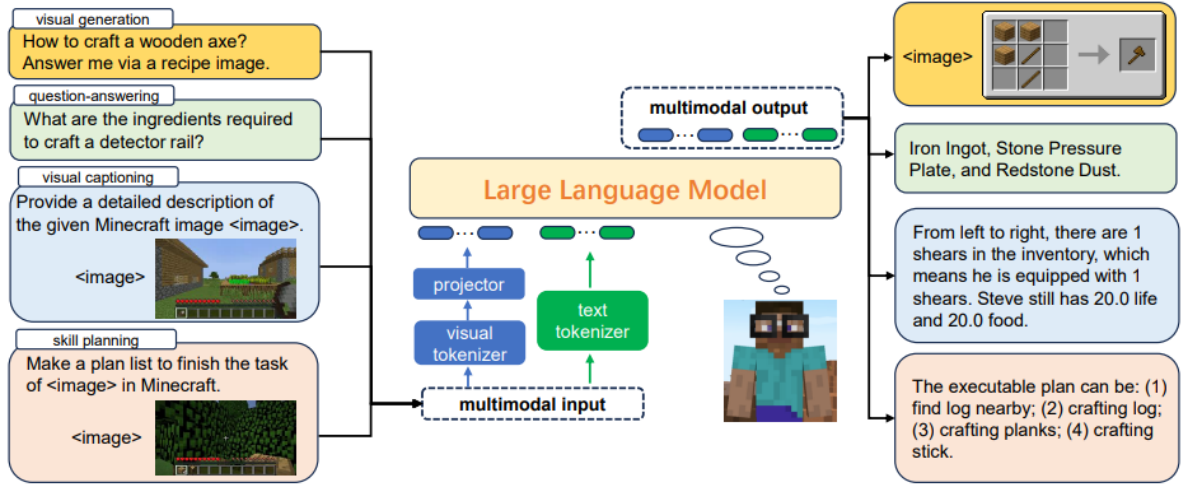


图 4: STEVE 的数据库构成

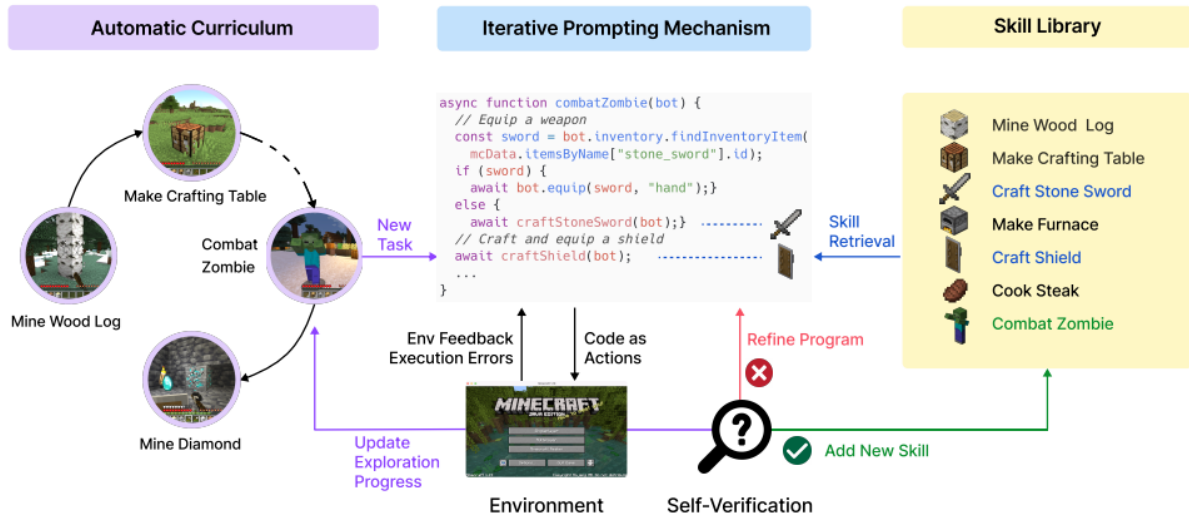


图 5: Voyager 的模型架构

最大化探索的自动课程；2) 用于存储和检索复杂行为的技能库；以及 3) 生成具体控制代码的新的迭代提示机制。我们选择使用代码作为行动空间，而不是低级运动命令，因为程序可以自然地表示时间上延伸和组合的动作 [15][22]，这对于 Minecraft 中的许多长期任务至关重要。VOYAGER 通过 prompt 和上下文学习 [26][20] 与一个 GPT-4 进行交互。该方法绕过了需要访问模型参数和显式基于梯度的训练或微调的需求。

VLMs 相关模型架构 由于 LLM 所驱动的智能代理并不能真正的“看到”环境，比如如何利用他眼前的石块去做一个石镐，这个过程通常需要利用游戏的相关结构将环境信息转化为形式化的文本信息输入给 LLM，而由于将图像信息转化为文本的过程由于语义稠密性的差异，可能造成一定的语义缺失。因此，一个自然而常见的思路则是，如何直接利于利用视觉、语言模型来进行具身智能任务的理解、规划和执行。本章节将介绍主流的 VLMs 及连接层架构，如图 7，分别以 LLAVA[16] 为代表的 projection 架构，InstructBLIP[8] 为代表的 q-former 架构以及 Flamingo[2] 为代表的 Perceiver 架构。

以 LLAVA[16] 为代表的 projection 架构 图 7b 展示了 llava 的训练架构，对于一个输入图片 X_v ，我们采用了一个预训练的 CLIP 模型的 encoder，ViT-L/14[19]，提取出了视觉特征 $Z_v = g(X_v)$ 。llava 在训练的过程中只考虑了最后一个 Transformer 层之前和之后的网格特征，使用一个简单的线性层将图像特征连接到词嵌入空间。具体

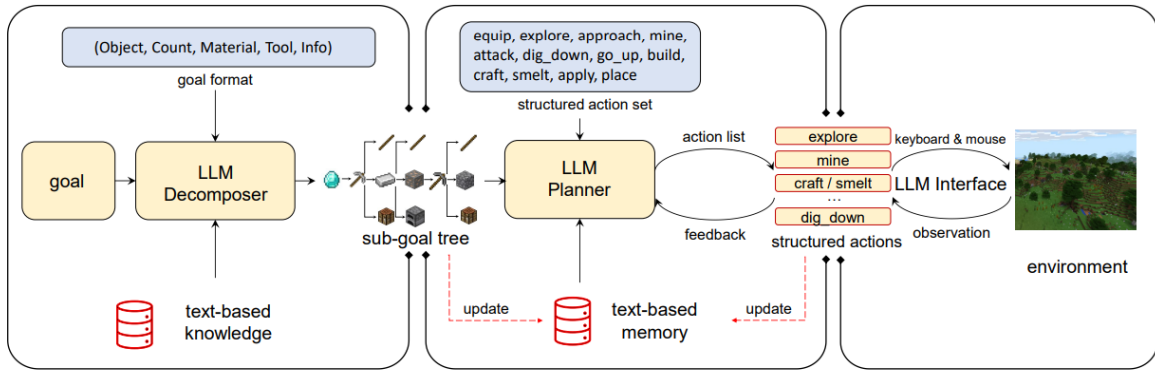
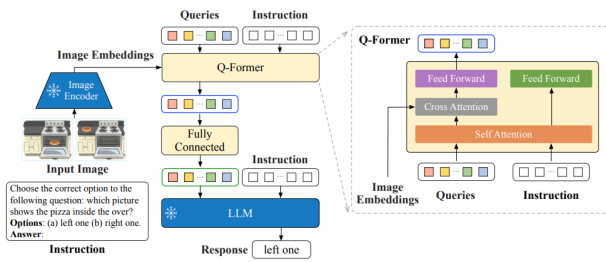
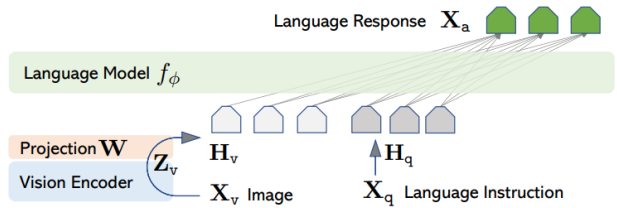


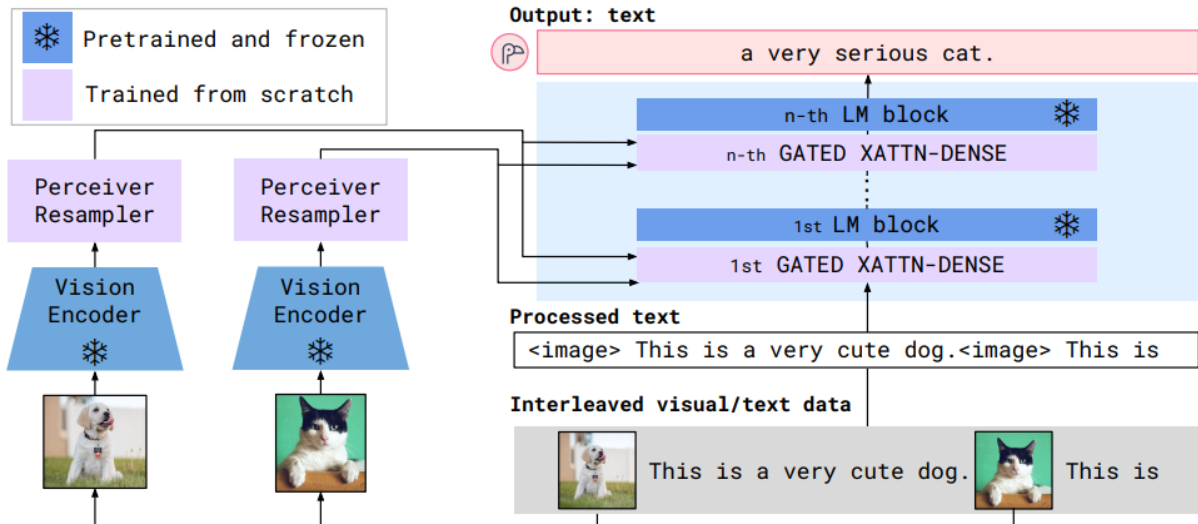
图 6: Ghost 的架构



(a) InstructBlip 采用 Q-Former 作为连接层



(b) LLaVA 采用 Projection 作为连接层



(c) Flamingo 采用 Perceiver 作为连接层

图 7: VLMs 的三种主流架构

来说, llava 使用一个可训练的投影矩阵 W 将 Zv 转换为 embeddings Hv , 它们的维度与语言模型中的 embeddings 空间相同。

InstructBLIP[8] 为代表的 q-former 架构 InstructBLIP 利用了 Q-Former7a, 从一个冻结的图像编码器中提取视觉特征。Q-Former 的输入包括一组 K 个可学习的 embedding, 这些 embeddings 通过交叉注意力与图像编码器的输出进行交互。Q-Former 的输出包括 K 个编码的视觉向量, 每个对应一个查询嵌入, 然后经过线性投影并馈送到冻结的 LLM。Q-Former 在进行指令微调之前分两个阶段使用图像-标题数据进行预训练。第一阶段使用冻结的图像编码器对 Q-Former 进行预训练, 用于视觉-语言表示学习。第二阶段将 Q-Former 的输出调整为用于文本生成的指令信息, 与冻结的 LLM 一起使用。在预训练之后, 使用指令微调对 Q-Former 进行微调, 其中 LLM 的输入包括来自 Q-Former 的视觉编码和任务指令。

Flamingo[2] 为代表的 Perceiver 架构 为了更有效地整合视觉信号, Flamingo 采用了基于 Perceiver 的架构, 从大量的视觉输入特征中生成了数百个标记, 然后使用交叉注意力层与 LM 层交替, 将视觉信息融合到语言解码过程中。训练目标是自回归的, 采用 NLL (Negative Log-Likelihood) 作为 loss。

- Perceiver 重采样器从图像/视频输入的视觉编码器中接收时空特征, 生成固定大小的视觉标记。
- 冻结的 LM 配备了新初始化的交叉注意力层, 这些层夹在预训练的 LM 层之间。因此, LM 可以在上述视觉标记的条件下生成文本。

关于目前学术界常见的仿真环境的调研

仿真环境 在游戏及其相关的具身智能领域, 仿真环境的性质往往决定了哪些环境状态可以作为 AI Agent 的输入, 目前学术界常见的一些仿真环境如下:

表 1: Overview of Embodied AI simulator 数据集特征与用于构建这些数据集的仿真环境密切相关, 在该摘要中, 概述了在创建数据集的过程中常用的 Embodied AI Simulator。

Simulation Environment	Kinematics	Continuous Extended States	Flexible Materials	Deformable Bodies	Realistic Fluid	Realistic Action Execution	TaskPlanning and/or Control	Game-Based or World-Based	Well-Formulated Tasks	Code Execution
OpenAIGym [6]	✓	✗	✗	✗	✗	✓	C	G	✗	✓
Matterport3D [7]	✗	✗	✗	✗	✗	✗	C	W	✗	✗
AI2THOR [13]	✓	✗	✗	✗	✗	✗	TP	G	✗	✓
VirtualHome [18]	✗	✗	✗	✗	✗	✗	TP	G	✗	✗
House3D [27]	✗	✗	✗	✗	✗	✗	TP	W	✗	✗
Habitat 1.0 [21]	✓	✗	✗	✗	✗	✓	C	W	✗	✓
Robosuite [32]	✓	✗	✗	✗	✗	✓	C	W	✗	✓
RFAI [10]	✓	✗	✓	✓	✓	✓	TP+C	W	✗	✓
Minecraft [4]	✓	✗	✓	✗	✗	✓	TP+C	G	✓	✓
GTA [1]	✓	✓	✓	✓	✓	✓	TP+C	G	✗	✗
Omnigibson [14]	✓	✓	✓	✓	✓	✓	TP+C	W	✗	✗
OctoGTA[28]	✓	✓	✓	✓	✓	✓	TP+C	G	✓	✓
Octogibson[28]	✓	✓	✓	✓	✓	✓	TP+C	W	✓	✓

未来发展趋势分析

在和 mincraft 相关的 AI Agent 的目前发展当中, 只能生成简洁的代码。当面临复杂任务时, 它经常会出现错误尝试, 并且在校正方面严重依赖环境反馈, 通常无法最终成功。未来的努力可以通过使 AI Agent 能够在更具挑战性的环境和任务中导航, 或将其与擅长创建复杂、结构良好的程序的最新 LLM 相结合, 以解决这些缺点。

此外, 现有的 Game Agent 仅在模拟环境中运行。转向现实世界可能会引入许多复杂性。例如, 现实世界的情境可能不提供像 MinEcraft 中那样容易获得的 groundtruth 信息, 这会使分辨环境细微差别变得更加复杂。目前对静态图像输入的依赖也引发了关于视频输入在提高任务性能方面效果的问题。

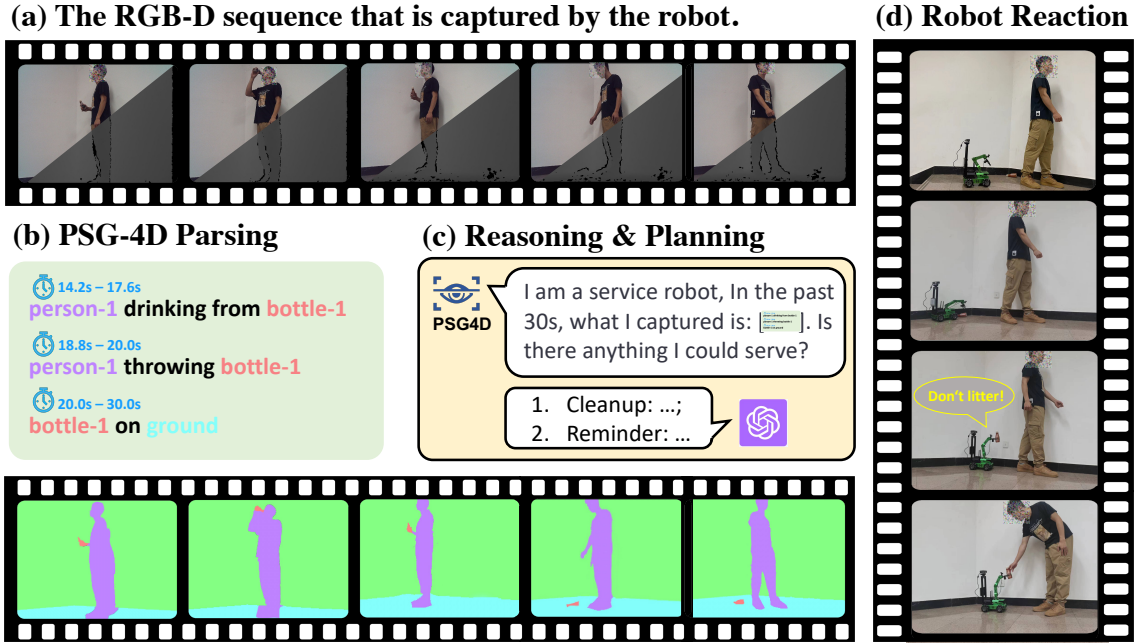


图 8: PSG-4D 模型部署的机器人演示。服务机器人解释了 (a) 中显示的 RGB-D 序列, 其中一个人在喝咖啡后将空瓶子扔在地上。机器人处理这个序列, 将其转化为 (b) 中所示的 4D 场景图。该图包括一组具有时间戳的三元组, 每个对象与一个全景掩模相关联, 准确地定位在 3D 空间中。机器人定期更新其 PSG4D 到 GPT-4, 等待反馈和指令。在这种情景下, GPT-4 建议机器人清理弃置的瓶子并提醒那个人关于他的行为。这一指令被转化为机器人的行动, 如 (d) 中所示。

场景图生成 对于 LLM 驱动的智能代理来说, 如何以更高级别的形式来表示场景信息成为一个共同的问题, 其中一个解决方案是利用场景图生成 [30]。如图 8 所示提供了一种利用场景图信息从而使现实机器人进行任务规划和决策的示例。

参考文献

- [1] Grand theft auto v, 2014.
- [2] Flamingo: a visual language model for few-shot learning, 2022.
- [3] Gpt-4 technical report, 2023.
- [4] Minecraft, 2023.
- [5] Anonymous. Steve-eye: Equipping LLM-based embodied agents with visual perception in open worlds. In *Submitted to The Twelfth International Conference on Learning Representations*, 2023. under review.
- [6] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.
- [7] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.
- [8] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023.
- [9] L. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D.-A. Huang, Y. Zhu, and A. Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge, 2022.
- [10] H. Fu, W. Xu, R. Ye, H. Xue, Z. Yu, T. Tang, Y. Li, W. Du, J. Zhang, and C. Lu. Rfuniverse: A multiphysics simulation platform for embodied ai, 2023.
- [11] P. Gao, S. Geng, R. Zhang, T. Ma, R. Fang, Y. Zhang, H. Li, and Y. Qiao. Clip-adapter: Better vision-language models with feature adapters, 2021.

- [12] W. Hill, I. Liu, A. D. M. Koch, D. Harvey, G. Konidaris, and S. James. Mineplanner: A benchmark for long-horizon planning in large minecraft worlds, 2023.
- [13] E. Kolve, R. Mottaghi, W. Han, E. Vanderbilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. [arXiv preprint arXiv:1712.05474](#), 2017.
- [14] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In [Conference on Robot Learning](#), pages 80–93. PMLR, 2023.
- [15] 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, 2023.
- [16] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning, 2023.
- [17] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback, 2022.
- [18] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba. Virtualhome: Simulating household activities via programs. In [Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition](#), pages 8494–8502, 2018.
- [19] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021.
- [20] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.
- [21] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, et al. Habitat: A platform for embodied ai research. In [Proceedings of the IEEE/CVF international conference on computer vision](#), pages 9339–9347, 2019.
- [22] 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, 2022.
- [23] M. Tatsubori, A. Munawar, and T. Moriyama. Design and implementation of linked planning domain definition language, 2019.
- [24] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023.
- [25] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J.-R. Wen. A survey on large language model based autonomous agents, 2023.
- [26] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus. Emergent abilities of large language models, 2022.
- [27] Y. Wu, Y. Wu, G. Gkioxari, and Y. Tian. Building generalizable agents with a realistic and rich 3d environment. [arXiv preprint arXiv:1801.02209](#), 2018.
- [28] J. Yang, Y. Dong, S. Liu, B. Li, Z. Wang, C. Jiang, H. Tan, J. Kang, Y. Zhang, K. Zhou, and Z. Liu. Octopus: Embodied vision-language programmer from environmental feedback, 2023.
- [29] G. N. Yannakakis and J. Togelius. [Artificial Intelligence and Games](#). Springer, 2018. <https://gameaibook.org>.
- [30] G. Zhu, L. Zhang, Y. Jiang, Y. Dang, H. Hou, P. Shen, M. Feng, X. Zhao, Q. Miao, S. A. A. Shah, and M. Bennamoun. Scene graph generation: A comprehensive survey, 2022.
- [31] X. Zhu, Y. Chen, H. Tian, C. Tao, W. Su, C. Yang, G. Huang, B. Li, L. Lu, X. Wang, Y. Qiao, Z. Zhang, and J. Dai. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory, 2023.
- [32] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu. robosuite: A modular simulation framework and benchmark for robot learning. [arXiv preprint arXiv:2009.12293](#), 2020.