
Cyber-Mind: Robotic Control System Based on ChatGLM

庄嘉鑫

2023270012

刘子源

2022310709

田倍闻

2023310488

张凯

2023316026

1 引言

具身智能的研究与发展涉及到多个学科领域的交叉融合，包括计算机科学、认知科学、神经科学和机器人学等。机器人学为具身智能提供了机械的身体和基本的运动控制，而深度学习中的神经网络则成为具身智能中主要的工具。基于试错的强化学习成为了具身智能中机器人技能的一种主要学习手段。计算机视觉、计算机图形学、自然语言处理等领域的进展为具身智能提供了处理视觉信号、语言交流等能力，使得机器能够更好地理解和参与人类的日常活动中。

机器人与环境的物理互动和感知是具身智能领域核心挑战之一。现代大语言模型（如 GPT-4 和 ChatGLM 等）的出现为解决这一问题提供了新的思路。通过将多模态的大语言模型与机器人结合，引入多模态输入，可以增强模型对现实中对象的理解，帮助机器人处理具身推理任务，实现更加智能地执行物理任务。

总之，探索具身智能是实现通用人工智能的必由之路。我们基于已有的大语言模型和视觉语言模型，在仿真模拟环境中实现了能够与人类进行自然语言交互和感知决策一体化的机器人系统。大语言模型赋予了系统的决策推理能力，视觉模型提升了系统的环境感知能力，同时，我们引入多轮对话反馈和思维链机制，进一步提升了系统的鲁棒性和可扩展性。我们的工作在 GitLab 上开源：<https://git.tsinghua.edu.cn/aml23-cyber-mind/cyber-mind>。

2 相关工作

具身智能 (Embodied AI) 意在给机器人赋予 AI 大模型的决策、推理能力，使机器人能够更加智能。现实场景下任务环境的复杂多样性，对具身智能模型的多模态感知和推理决策能力都提出了较高的要求。近年来，机器人多模态感知决策一体化的工作不断提出，Brohan 等提出的 Robotics Transformer[1] 利用 Transformer 网络架构统一

了图像数据、自然语言指令和机器人动作参数，采用短序列的图像和自然语言指令作为输入，并在每个时间步为机器人输出一个动作模型。模型可以吸收大量的数据并有效地推广，增加了机器人控制的泛化能力，在多个不同任务数据集上都取得了不错的结果。[2] 在 Robotics Transformer 的基础上，将机器人动作表示为特殊的语言指令，进一步将 Internet 上的 VQA 任务数据和机器人动作语言指令数据共同融入训练过程，模仿 Chain of Thought 的过程提升了模型的因果推理能力，增强了机器人模型对新场景的泛化能力。视觉模型感知能力和语言模型推理能力的辅助，赋予了机器人更强的感知决策推理能力。随着大语言模型 (LLM) 的不断优化，大语言模型表现出更优越的逻辑推理、代码编写等能力，利用大模型辅助实现机器人决策推理的工作也相继出现。谷歌团队基于提示工程，利用大模型的代码编写能力实现使用语言驱动机械臂的工作。文中提出 Code as Policies 的方案，通过提示输入模版和 API 信息，大模型会按照一定的策略编写代码和调用 API 来驱动机械臂实现一定的动作 [3]。Huang Wenlong 在 Code as Policy 的基础上，使用视觉语言模型 (VLM) 让模型拥有“视觉”感知能力，以增加机械臂对于开放世界的通用性。同时对轨迹进行一定的优化以满足环境的变化和物理限制，增加了机械臂的效率 [4]。

3 方法

已有的工作广泛使用大语言模型根据任务直接生成代码执行，存在以下缺点：(1) 直接生成代码要求大语言模型模型较强的代码生成能力及鲁棒性，在复杂的现实场景中难以达到高成功率 (2) 感知决策缺少环境过程反馈，一次性的代码生成和执行操作不适合真实的环境交互过程 (3) 视觉模型没有和语言模型进行良好的结合交互。如 [4] 中，虽然引入了视觉语言模型 (VLM)，但缺乏视觉模型和大语言模型的交互，视觉模型只适用于生成特定的图片，限制了整个系统可以完成的任务类型。基于以上问题，我们提出一个在线的安全、鲁棒、可扩展的方案，以 LLM 为控制系统的决策中心，通过提供机械臂 API 的接口描述，让大语言模型自主选择合适的 API 以及正确的参数来驱动机械臂完成指定任务，同时加入了大语言模型与视觉语言模型及环境的灵活交互、反馈过程，提升了模型的鲁棒性。系统流程框架图如图1所示。

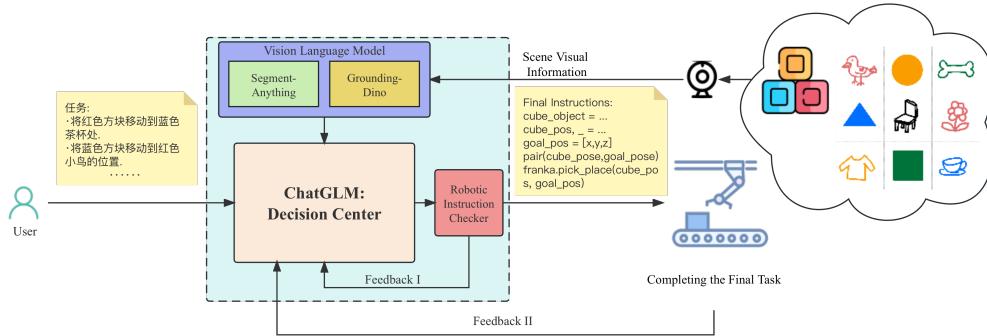


图 1: Cyber-Mind Pipeline

我们提供了精简版的机械臂 API 以及视觉语言模型的 API 供大语言模型来进行调用。用户仅需要与大语言模型进行交互输入基本的自然语言任务描述，大语言模型自动解析、处理任务，自主推理生成正确的参数来调用相应的机械臂 API。采用 ChatGLM 作为整个系统的决策中心——“Cyber-Mind”，Segment Anything 及 Grounding Dino 接口作为视觉语言模型。ChatGLM 处理用户自然语言输入，自动生成对应的问题和提示词传输给视觉语言模型，视觉语言模型根据 ChatGLM 生成的文本要求及摄像头的实时图像标记出目标位置，返回位置信息给 ChatGLM，进一步整合信息按照要求输出对应的参数来调用我们所提供的机械臂的 API，驱动机械臂完成用户任务。

我们的方法主要创新有以下几点：

1. ChatGLM 的 Prompt 中引入了 Chain of Thought 机制，并给出了例子，提升了模型的推理能力，如图2所示。

```
...
LLM_prompt_v4 = """
你现在是一个机械臂的代理，需要将我的指令语言转化成对应的格式以供机械臂API调用。
我将询问你把方块移动到目标地点。
我现在已经告诉你了已有cube的信息
cube1 : 蓝色
cube2 : 红色
cube3 : 绿色
其中方块名字就是 cube1,cube2,cube3...,上面代码中有cube的颜色坐标之类属性的信息
目标地点则需要你去写一段文字去咨询视觉语言模型，由于视觉语言模型只能读英语，所以请你写一段英文去咨询视觉语言模型。
请你先识别我指令中的目的地，按照所绘的模板: find the [object]。将你认为的目的地翻译成英文填入[object]中输出。
可以一步步推理方块名称和目的地，输出推理的过程
因此，你最终的输出将严格按照格式: $$方块名字，一段咨询视觉语言模型的英文字符串$$
中间一定要逗号隔开！
COT prompt:这是一个例子：
比如用户输入：“已知图片中有个红色的小鸟，我想把和小鸟颜色相同的方块放到黑色椅子的位置”
推理过程如下：
1. 方块是和小鸟颜色相同，即红色方块，根据提供的方块信息，得知方块名字为cube2
2. 目的地是黑色椅子，所以按照模版得到的指令是“find the black chair”
3. 按照格式进行输出$$cube2, find the black chair$$
下面，是我的指令语言：请严格按照格式输出内容，不要多余的信息，不准在推理过程中或答案无关的文本中留下$$，以免我处理答案出现错误。
...
"""

```

图 2: 引入 CoT 机制的 Prompt

2. 在 ChatGLM 输出之后加入了格式检查机制，确保模型生成的参数传入到对应的 API 中安全可靠，如图3所示。

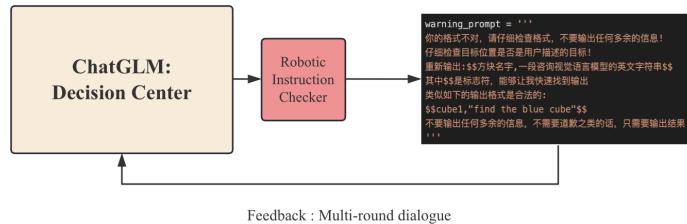


图 3: 多轮对话格式反馈

3. 由 ChatGLM 首先向 VLM 发出请求，再由 VLM 向 ChatGLM 返回结果，最终决策仍然由 ChatGLM 给出，通过两者交互的过程，提升了模型的感知决策一体化能力。

整个任务完成后，系统会暂停并等待用户输入下一段任务描述。用户输入指令到最终任务执行完成的过程不需要人为干涉，由系统自主反馈运行，即可高概率完成任务，实现了高鲁棒性的具身智能控制决策系统。

4 实验结果

4.1 基本性能测试

为进一步探究系统性能及各模块实际作用，我们设计了一些简单的和复杂的任务进行测试。为此，我们使用 Isaac Sim 仿真软件搭建测试场景。其中包含三个颜色的方块以及一个九宫格，九宫格中的每一个格子都有特定的图案。根据我们设计的测试场景，我们设计了不同的任务，包括简单的调用机械臂 API，例如：请把方块放到指定位置。也包括复杂的任务，例如：请将与咖啡杯相同颜色的方块放到绿色骨头位置。

在测试阶段，仿真场景中的摄像机图片会实时传递给系统，供视觉语言模型识别物体，感知环境。实验仅需要用户进行任务的自然语言描述，后续推理过程及反馈均由系统自动完成。实验表明，我们的方案能够比较好的实现这些任务。本项目的实验环境：

- 仿真环境：Nvidia Isaac Sim
- 语言模型：ChatGLM_turbo
- 视觉语言模型：Segment Anything + Grounding Dino

如图4所示：demo1 输入任务为：“将红色方块放置在蓝色咖啡杯处”，demo2 输入任务为：“已知咖啡杯为蓝色，将于咖啡杯颜色相同的方块放置在黑色椅子处”。机械臂顺利完成了任务，通过视频可以看到在用户输入指令后，系统不需要再人为进行干涉，ChatGLM 能够自行处理信息，并生成指定的文本调用视觉语言模型获取对于物体的位置信息。最终感知决策控制机械臂完成任务。

此外，我们还进行重复测试来检验系统的鲁棒性和稳定性，指令采用类似 demo1 和 demo2 结构的指令。每一种任务类型运行 10 次，正确率测试如表1所示。

表 1: 不同任务类型成功率测试

指令格式	成功率
demo1(简单)	100%
demo2(复杂)	90%

由于系统在线运行，当未完成任务时，用户再输入一遍指令后系统能够将错误放置的方块再放回指定位置上。出现错误的大部分原因是由于视觉语言模型对于我们所提供的九宫格的图片识别不够准确，虽然可以依靠添加颜色信息提升视觉语言模型的能力，但仍存在视觉语言模型错判漏判情况。但由于系统在运行的过程中是安全受限的，即使任务执行出错，也不会破坏整个系统的工作，用户可以重新输入指令纠正。

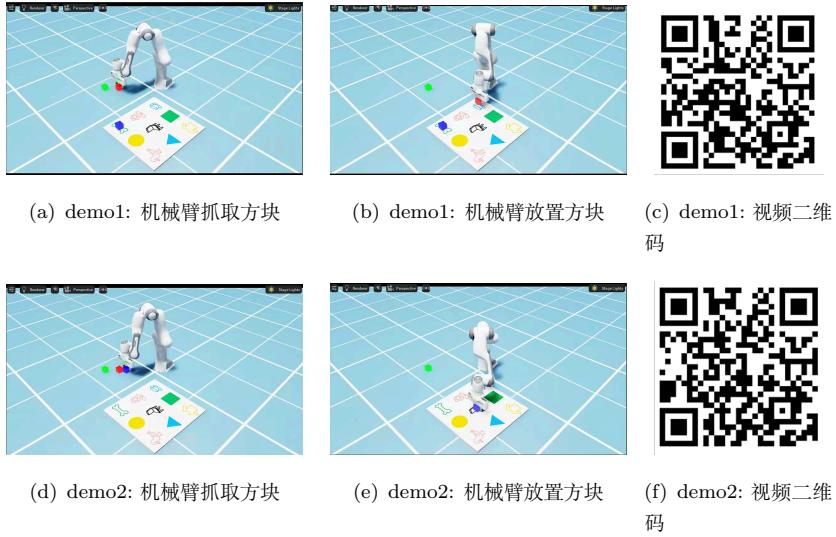


图 4: 测试 demo 效果图. demo1 输入: “将红色方块放置在蓝色咖啡杯处”, demo2 输入: “已知咖啡杯为蓝色, 将于咖啡杯颜色相同的方块放置在黑色椅子处”

4.2 消融实验

为检验 CoT 对系统的提升能力, 我们尝试了加入 CoT 和不加 CoT 的不同实验结果。若加入 CoT, 实验结果如下:

Initial Answer = 1. 方块是和咖啡杯颜色相同, 即蓝色方块, 根据提供的方块信息, 得知方块名字为 cube1。2. 目的地是与咖啡杯颜色相同的三角形处, 所以按照模板得到的指令是“find the triangle with the same color as the coffee cup”。3. 按照格式进行输出: \$\$cube1,find the triangle with the same color as the coffee cup\$\$

如果不加入 CoT, 输入: “已知咖啡杯为蓝色, 请将与咖啡杯颜色相同的方块放置在与咖啡杯颜色相同的三角形处”, 实验结果如下:

Initial Answer = 根据您的指令, 我们需要找到与咖啡杯颜色相同的方块, 并将其放置在与咖啡杯颜色相同的三角形处。根据已知的方块信息, 咖啡杯对应的方块是 cube1, 颜色为蓝色。因此, 我们需要找到与蓝色相同的方块。找到目标方块后, 我们需要将其放置在与咖啡杯颜色相同的三角形处。由于视觉语言模型只能理解英语, 我们需用英语描述三角形的位置。以下是一段咨询视觉语言模型的英文描述: “triangle with the same color as the coffee cup” 所以, 根据您的指令, 我们的输出为: [cube1, “find the triangle with the same color as the coffee cup”] **Filtered Out =** Matching failed, return Matching failed return name = Matching failed Format wrong:Multiple rounds of dialogue feedback... **Initial Answer =** 抱歉刚刚的格式出现了问题, 现在按照您的要求重新输出: \$\$cube1,”find the triangle with the same color as the coffee cup”\$\$ 请查阅。**Filtered Out =** cube1,”find the triangle with the same color as the coffee cup” Format Correct:Starting the VLM model

可见，当不加入 CoT 时，系统最终仍然有大概率成功决策，但是推理过程远远不如加入 CoT 清晰，且第一轮未按照正确格式输出，经过格式检查和反馈之后，第二轮才达到了和加入 CoT 相同的结果。通过简单的实验，同时验证了 CoT 及格式检查机制的重要性。

我们也可以通过大量的实验发现加入 CoT 和格式检查之后，系统的成功率接近 100%，远远超过不加入 CoT 和格式检查的成功率，这是由于如果缺乏 CoT，很多复杂的任务将无法有效推理，最终错误决策。因此，消融实验证明了 CoT 及格式检查机制的有效性和重要性。

5 结论

本文提出了一种在线决策反馈的高鲁棒性的具身智能系统框架，将 ChatGLM 大模型作为系统的决策中心，通过视觉语言模型感知环境，实现了感知决策一体化模型，同时加入了 CoT 和格式检查多轮反馈，提升了系统鲁棒性和可拓展性。实验中，我们使用 Isaac Sim 仿真软件中搭建了模拟场景，并输入不同难度的任务描述进行了大量的实验和测试。我们以其中两个代表性例子作为案例展示，并制作视频展示效果。总体来说，我们所提出的方案能够比较好地完成设计的任务，具有较高的鲁棒能力，可以通过用户自然语言描述完成指定的复杂任务。后续工作中，我们会尝试利用多轮对话的特点来构建 RLHF 来进一步提升系统的鲁棒性，或者引入行为模仿学习来实现系统的在线学习。同时值得一提的是，本方案对于大模型以及视觉语言模型依赖比较强，因此系统的性能很大程度上受限于大语言模型和视觉语言模型的能力以及提示词，未来工作中也将针对此问题进行更进一步的研究和改进，提升系统综合能力。

6 贡献归属

- 庄嘉鑫（组长）：协调组员工作，Isaac Sim 仿真场景搭建，方法设计，系统联调以及报告撰写
- 张凯：文献调研，ChatGLM 及 VLM 模型 Pipeline 构建，推理反馈，系统联调以及报告撰写
- 田倍闻：文献调研，Segment Anything，Grounding Dino 视觉语言模型复现以及方法设计
- 刘子源：文献调研，方法设计，材料汇总以及系统测试

References

- [1] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan C. Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Anand Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Ho Vuong, F. Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. *ArXiv*, abs/2212.06817, 2022.
- [2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Krzysztof Choromanski, Tianli Ding, Danny Driess, Chelsea Finn, Peter R. Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil J. Joshi, Ryan C. Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Sergey Levine, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Pierre Sermanet, Jaspiar Singh, Anika Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Ho Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Ted Xiao, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *ArXiv*, abs/2307.15818, 2023.
- [3] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control, 2023.
- [4] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models, 2023.