上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

决策大模型：智能体与机器人

张伟楠

上海交通大学
http://wnzhang.net

2024年5月

决策大模型是新一代人工智能底层技术：

- 赋能智能体(AI Agents)在数字世界中做有效决策；
- 赋能具身智能机器人(Embodied AI Robots)在物理世界中做有效控制。

2024年上海交通大学ACM班强化学习课程大纲

强化学习基础部分

（中文课件）

1. 强化学习、探索与利用
2. MDP和动态规划
3. 值函数估计
4. 无模型控制方法
5. 规划与学习
6. 参数化的值函数和策略
7. 深度强化学习价值方法
8. 深度强化学习策略方法

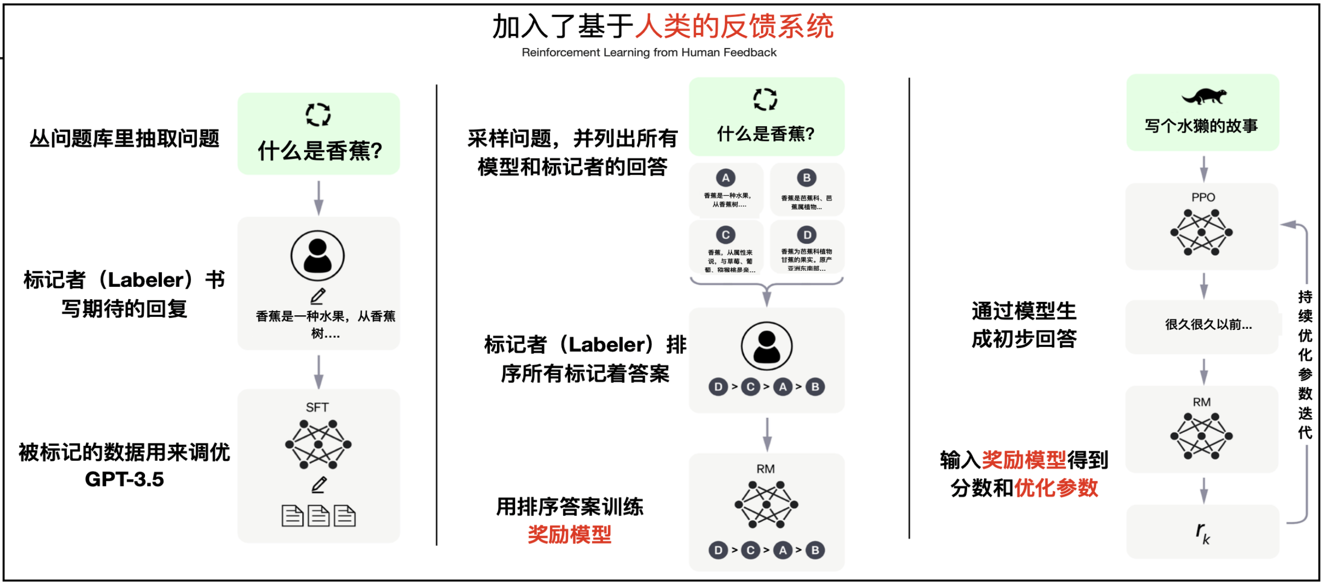
强化学习前沿部分

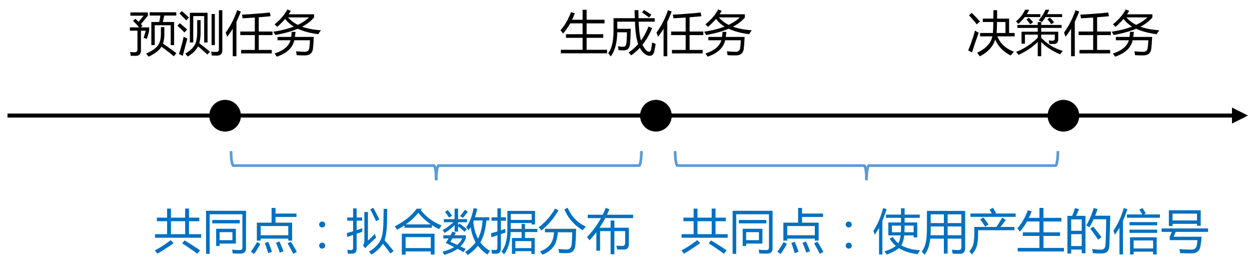
（英文课件）

9. 基于模型的深度强化学习
10. 模仿学习
11. 离线强化学习
12. 多智能体强化学习基础
13. 多智能体强化学习前沿
14. AI Agent与决策大模型
15. 基于扩散模型的强化学习
16. 技术交流与回顾

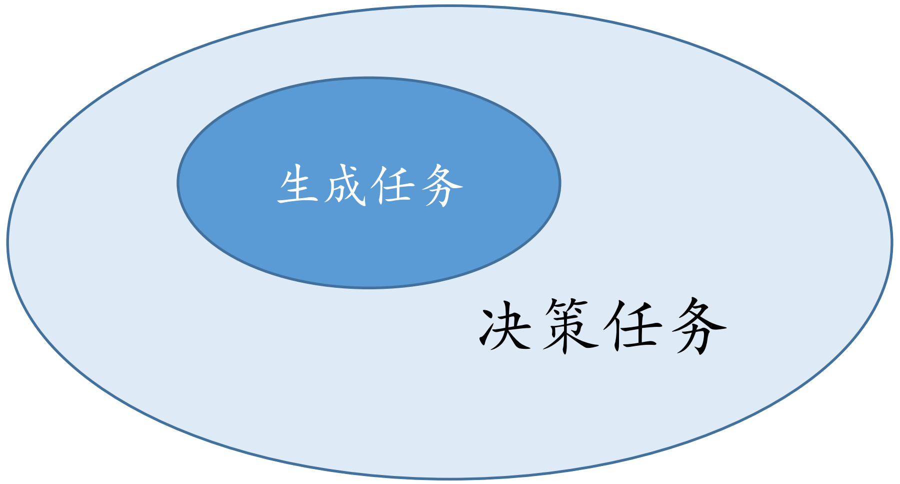
从生成到决策

- 面向**预测**任务的机器学习
  - 给予特征数据，预测目标概率分布（监督学习）
- 面向**生成**任务的机器学习
  - 生成数据样例（无监督学习）
- 面向**决策**任务的机器学习
  - 在动态系统中采取序列行动决策


加入了基于人类的反馈系统
Reinforcement Learning from Human Feedback
从问题库里抽取问题
什么是香蕉?
标记者（Labeler）书写期待的回复
香蕉是一种水果，从香蕉树....
被标记的数据用来调优GPT-3.5
SFT
采样问题，并列出所有模型和标记者的回答
什么是香蕉?
A
香蕉是一种水果，从香蕉树....
B
香蕉是芭蕉科、芭蕉属植物...
C
D
标记者（Labeler）排序所有标记着答案
D > C > A > B
用排序答案训练奖励模型
RM
D > C > A > B
写个水獭的故事
PPO
通过模型生成初步回答
很久很久以前...
RM
输入奖励模型得到分数和优化参数
r_k
持续优化参数迭代



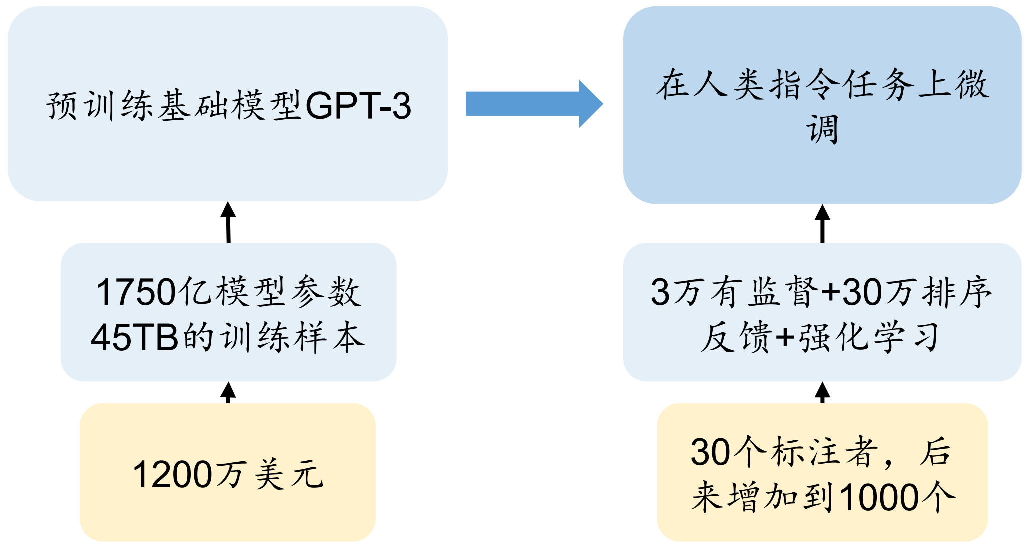
决策大模型：ChatGPT的“下一步”



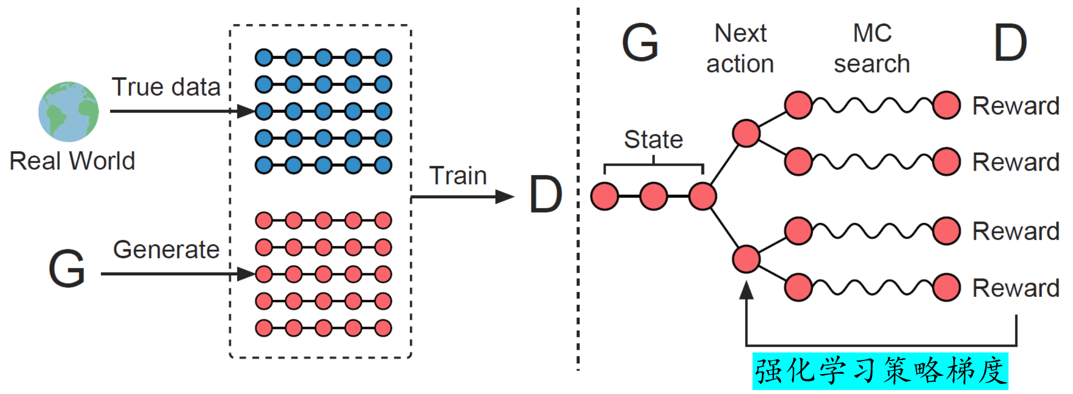
ChatGPT带来的启示

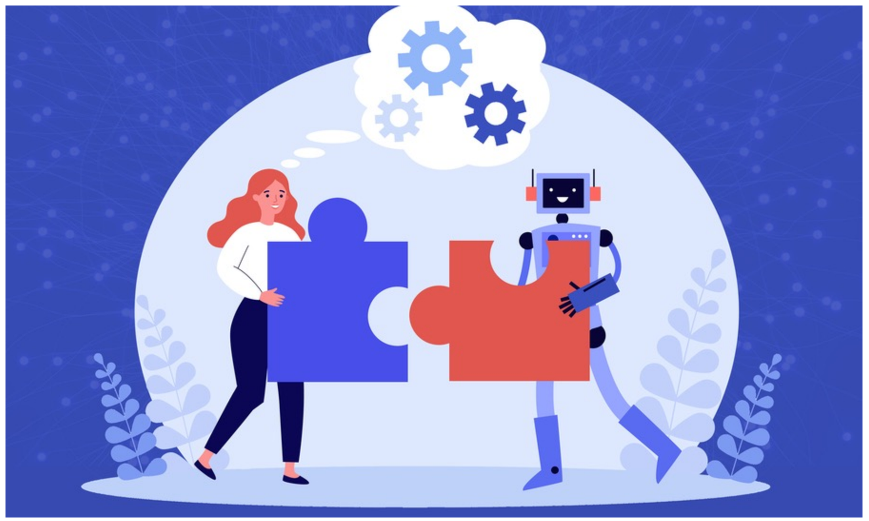
1. 基础模型： ChatGPT的基本能力都在GPT-3中蕴含了，后续的微调只是解锁了它本有的能力；

2. 决策任务： 文字生成任务其实就是决策任务，可以用强化学习训练；



ChatGPT基础模型





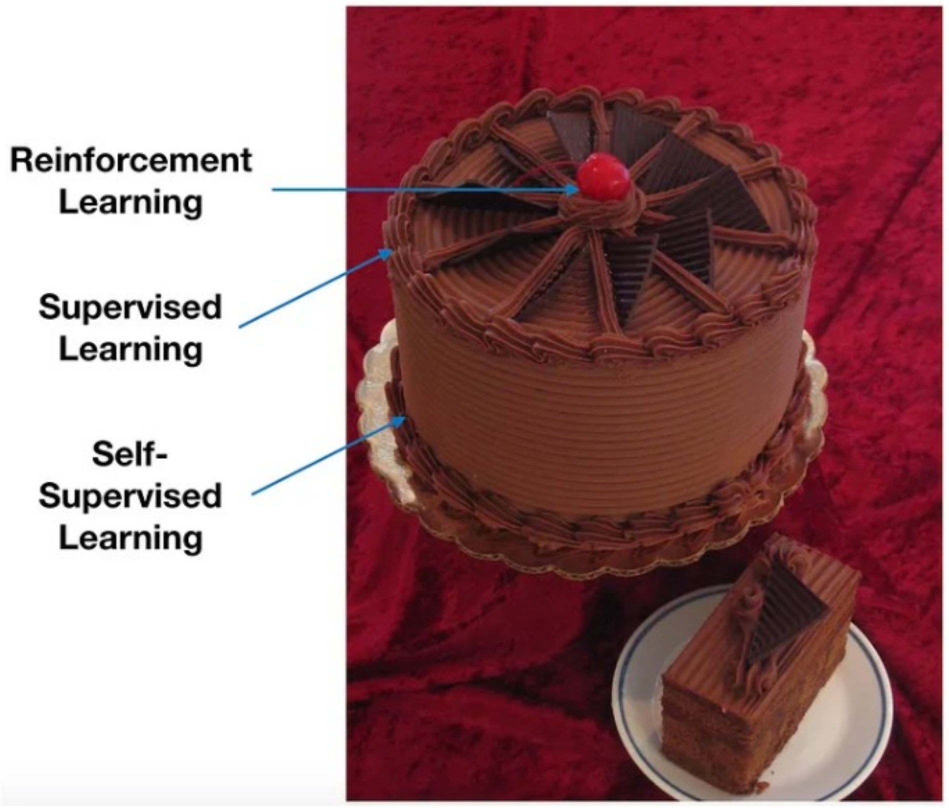
强化学习训练

3. 优化价值： 人在环路的训练方式实现人智协同（对齐人类价值观），实质上是优化了ChatGPT带给用户的价值；

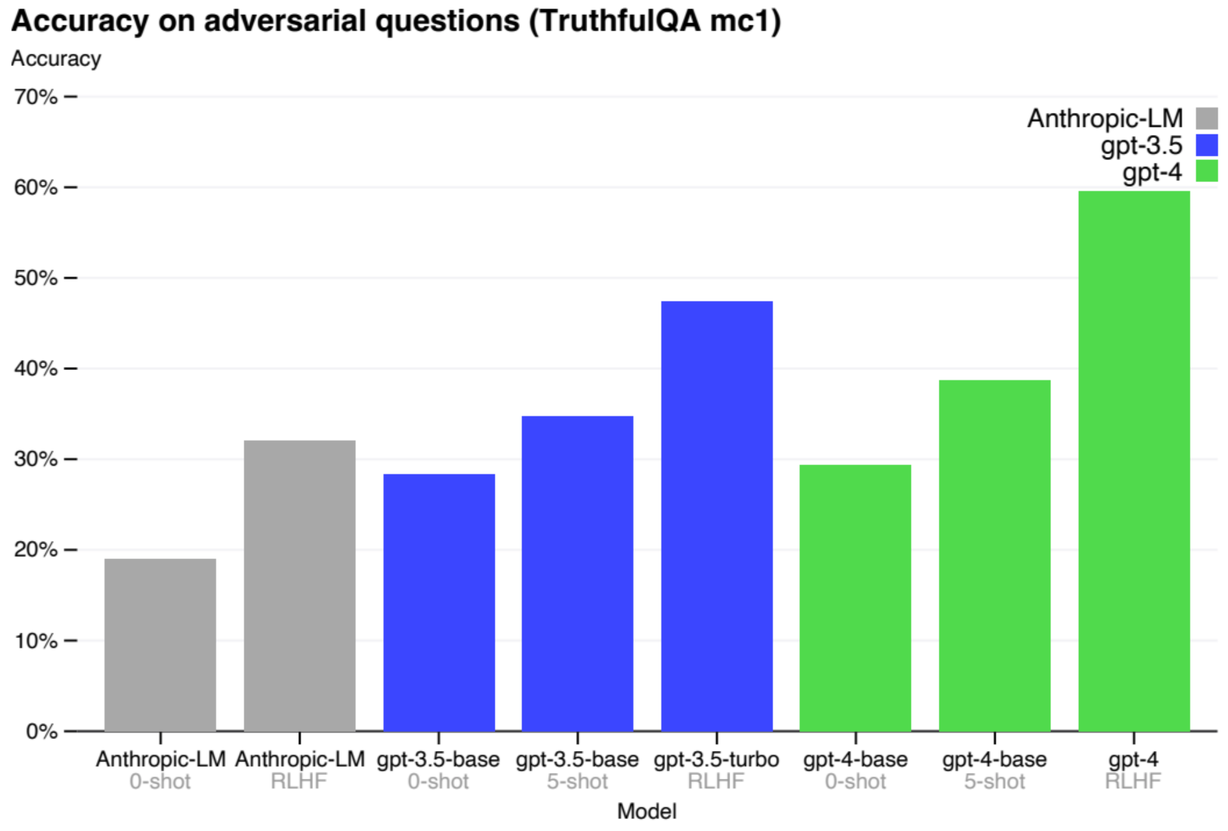
强化学习在大模型中的作用

大家之前以为强化学习的作用

"If intelligence is a cake, the bulk of the cake is self-supervised learning, the icing on the cake is supervised learning, and the cherry on the cake is reinforcement learning (RL)." — Yann LeCun head of Facebook AI



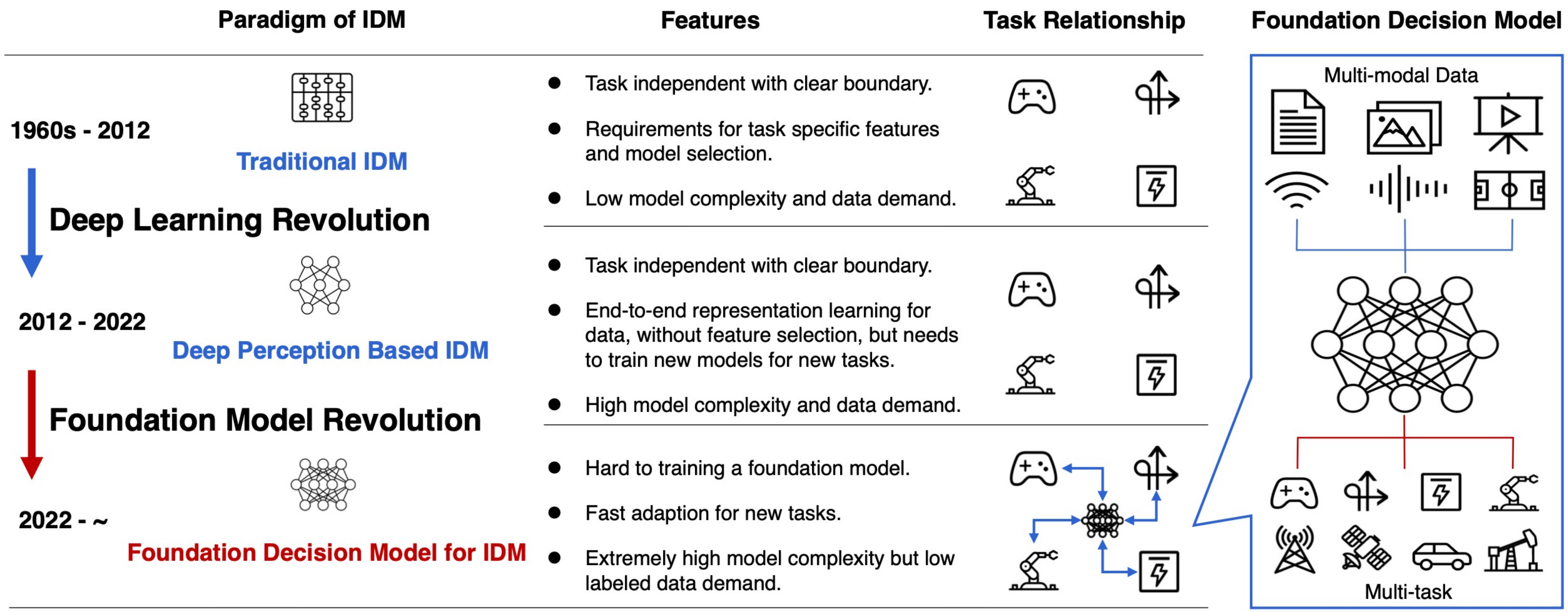
大家现在看到的强化学习的作用



from GPT-4 Technical Report

面向智能决策（Intelligent Decision-Making）的基础模型

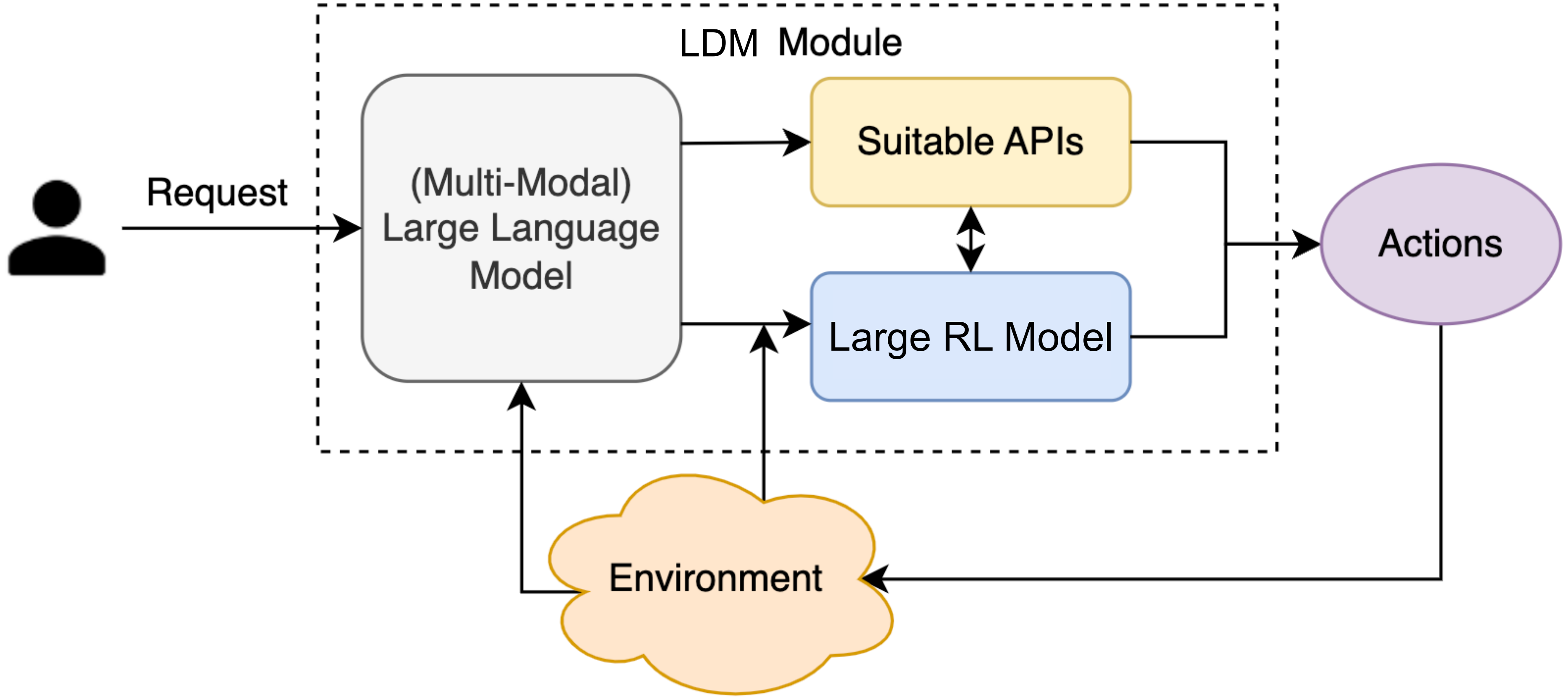
- 使用大模型解决决策任务中的环境变化、开放环境、策略泛化性问题



Paradigm of IDM	Features
1960s - 2012 Traditional IDM	• Task independent with clear boundary. • Requirements for task specific features and model selection. • Low model complexity and data demand.
Deep Learning Revolution	
2012 - 2022 Deep Perception Based IDM	• Task independent with clear boundary. • End-to-end representation learning for data, without feature selection, but needs to train new models for new tasks. • High model complexity and data demand.
Foundation Model Revolution	
2022 - ~ Foundation Decision Model for IDM	• Hard to training a foundation model. • Fast adaption for new tasks. • Extremely high model complexity but low labeled data demand.

决策大模型的两种范式

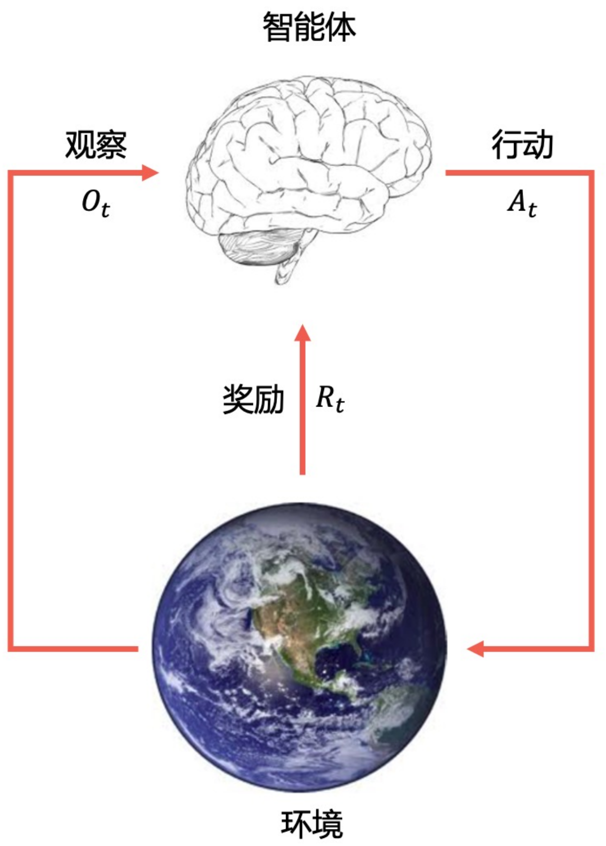
- ☐ 范式A（LLM Agent）：以大语言模型为中心，配合API选择的决策
- ☐ 范式B（Large RL Models）：以强化学习大模型为中心，做底层的决策控制



范式A：LLM Agent – 以大语言模型为中心，配合API选择的决策

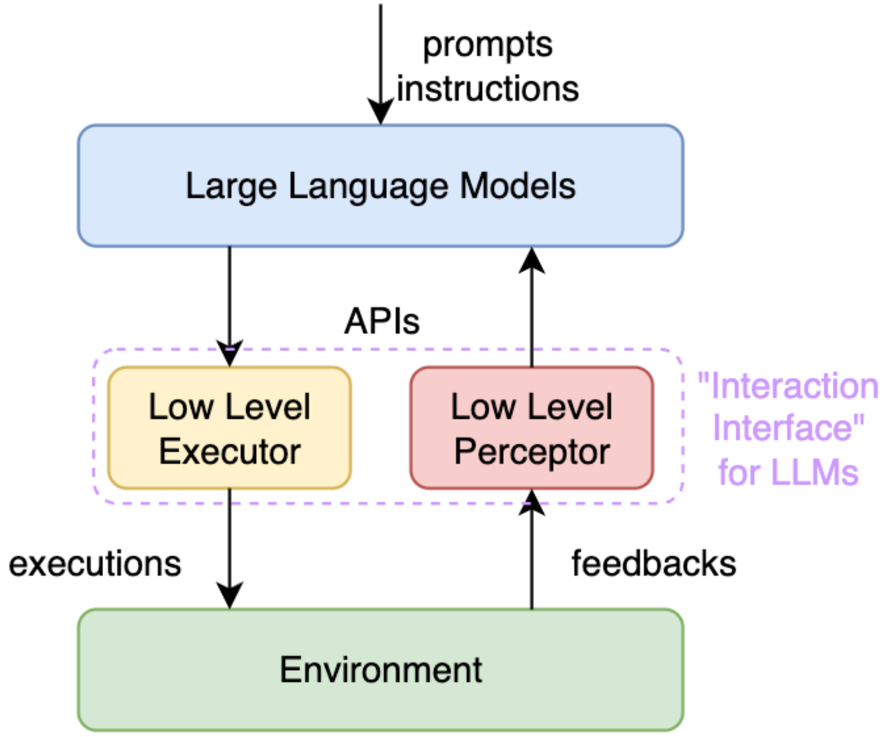
什么是智能体（AI Agent）？

- 智能体是集环境感知和多轮决策为一体的人工智能个体，它通过不断感知当前环境观测，计算得到适合的决策动作，进而达到任务目标。



- 在每一步 t，智能体：
  - 获得观察 O_t
  - 执行动作 A_t
  - 获得奖励 R_t
- 环境：
  - 获得动作 A_t
  - 给出奖励 R_t
  - 给出观察 O_{t+1}
- t 在环境这一步增加

基于LLM的智能体则集成了LLM的世界知识，可以在认知层面上、在数字世界中做出高质量的决策

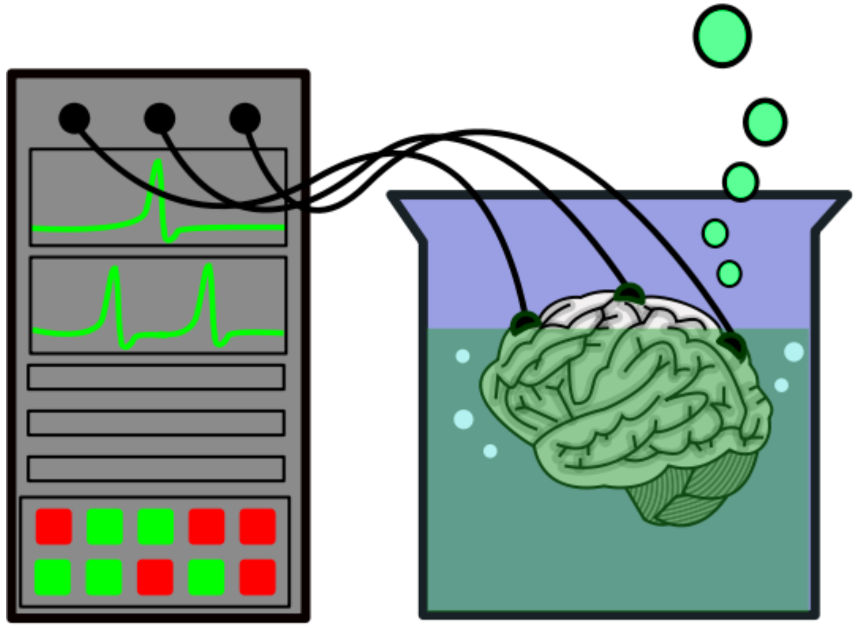


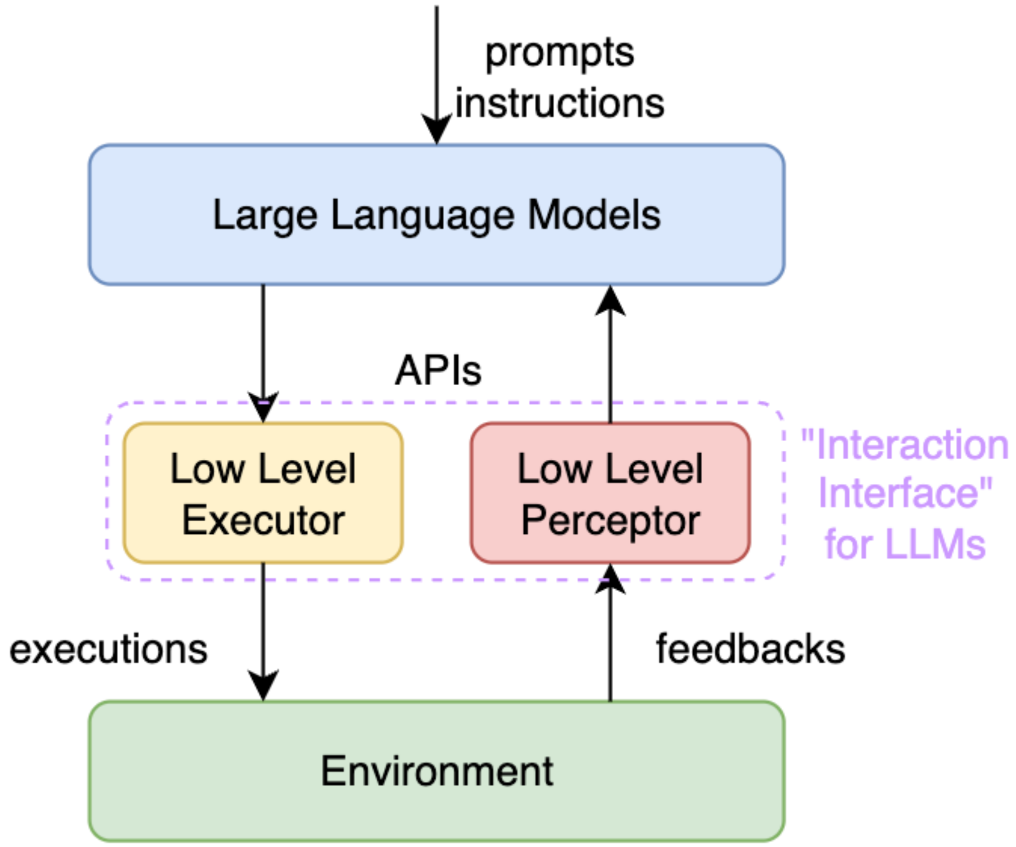
- Prompts: language in a certain pattern
- Low level executor: robots, agents
- Low level perceptor: usually perception models

范式A：LLM Agent

Large Language Models: The Brain in a Vat

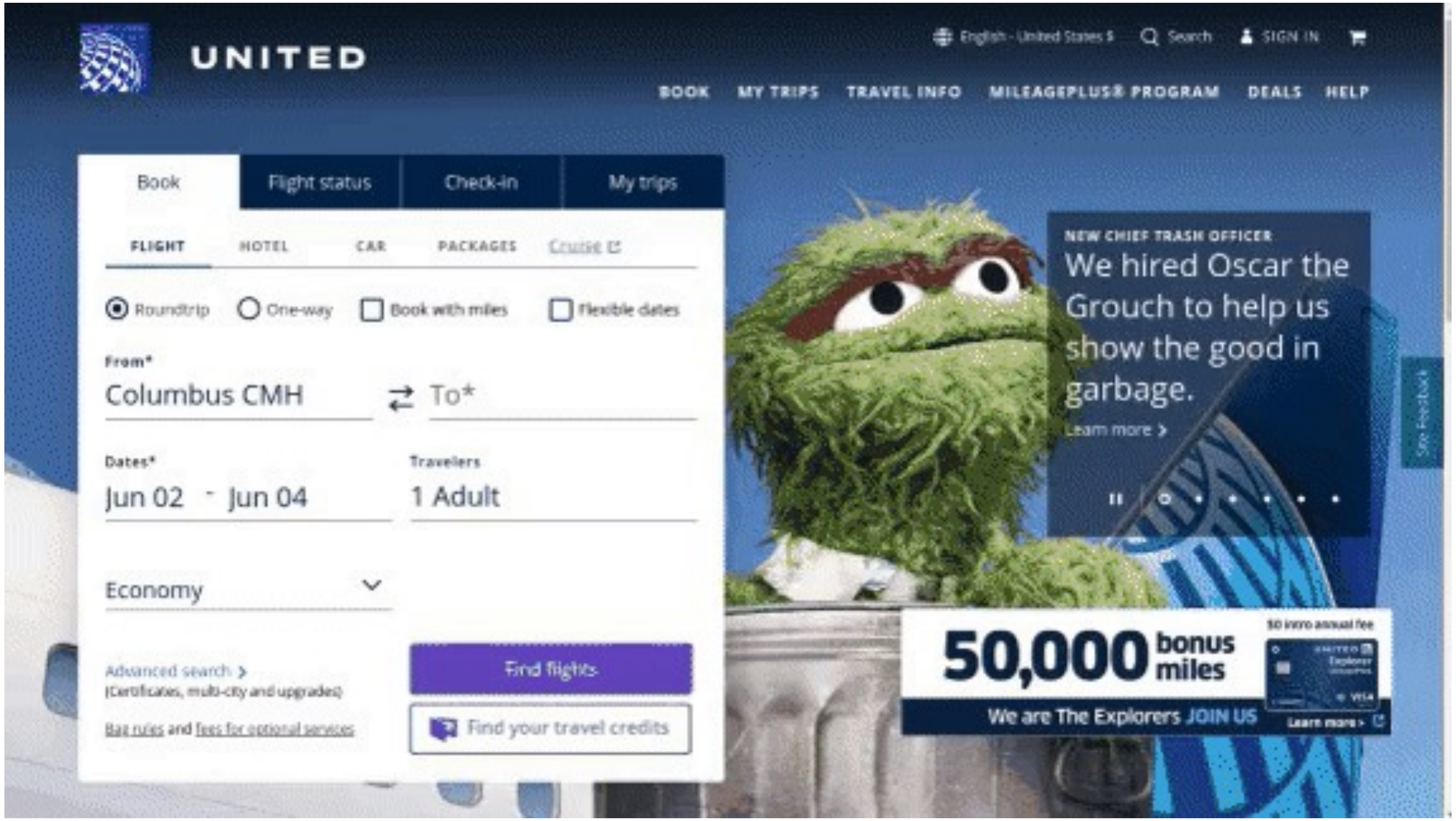
- LLMs have shown remarkable reasoning and planning ability.
- To utilize LLMs in real-life control tasks, additional supports are necessary.
- Current researches over LLMs and external environment interactions share a similar pipeline.





- Prompts: language in a certain pattern
- Low level executor: robots, agents
- Low level perceptor: usually perception models

范式A：LLM Agent - 以大语言模型为中心，配合API选择的决策 示例：购票



[✓ Roundtrip] One-way ☐ Book with miles ☐ Flexible dates

From* Columbus CMH ⇄ To*

Dates* Jun 02 - Jun 04

Travelers 1 Adult

Economy

(Certificates, multi-city and upgrades)

Bag rules and fees for optional services

NEW CHIEF TRASH OFFICER

We hired Oscar the Grouch to help us show the good in garbage.

Learn more ›

50,000 bonus miles

$0 intro annual fee

We are The Explorers JOIN US

Learn more ›

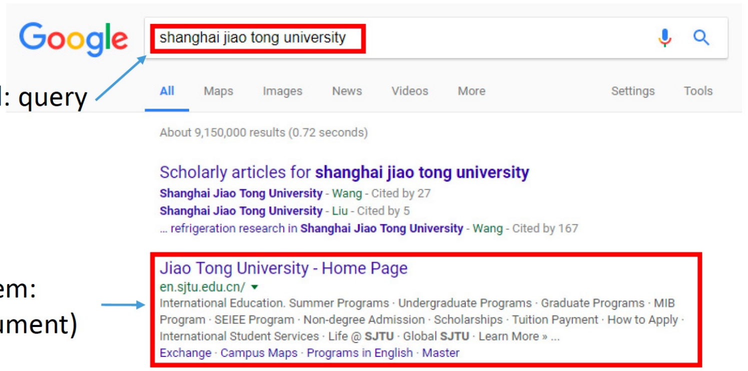
Xiang Deng et al. Mind2Web: Towards a Generalist Agent for the Web. Arxiv 2023.

重新思考信息检索

- Information retrieval (IR) is the activity of obtaining information items relevant to an information need from a collection of information items.

- 传统范式：通过构建文档索引，直接返回匹配搜索query的文档

Information need: query



Google

shanghai jiao tong university

All · Maps · Images · News · Videos · More · Settings · Tools

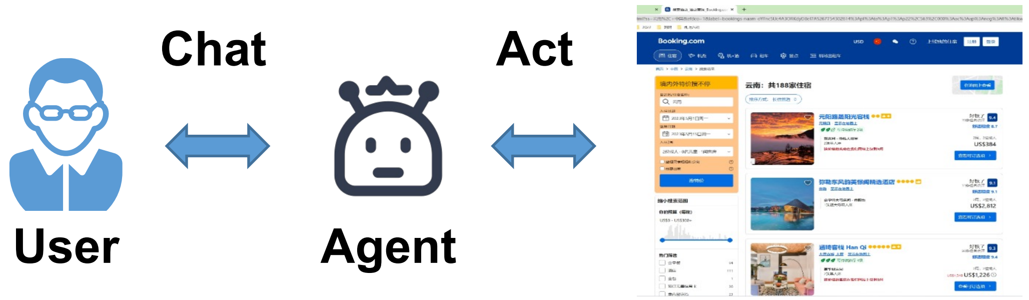
About 9,150,000 results (0.72 seconds)

Scholarly articles for **shanghai jiao tong university**
Shanghai Jiao Tong University - Wang - Cited by 27
Shanghai Jiao Tong University - Liu - Cited by 5
… refrigeration research in **Shanghai Jiao Tong University** - Wang - Cited by 167

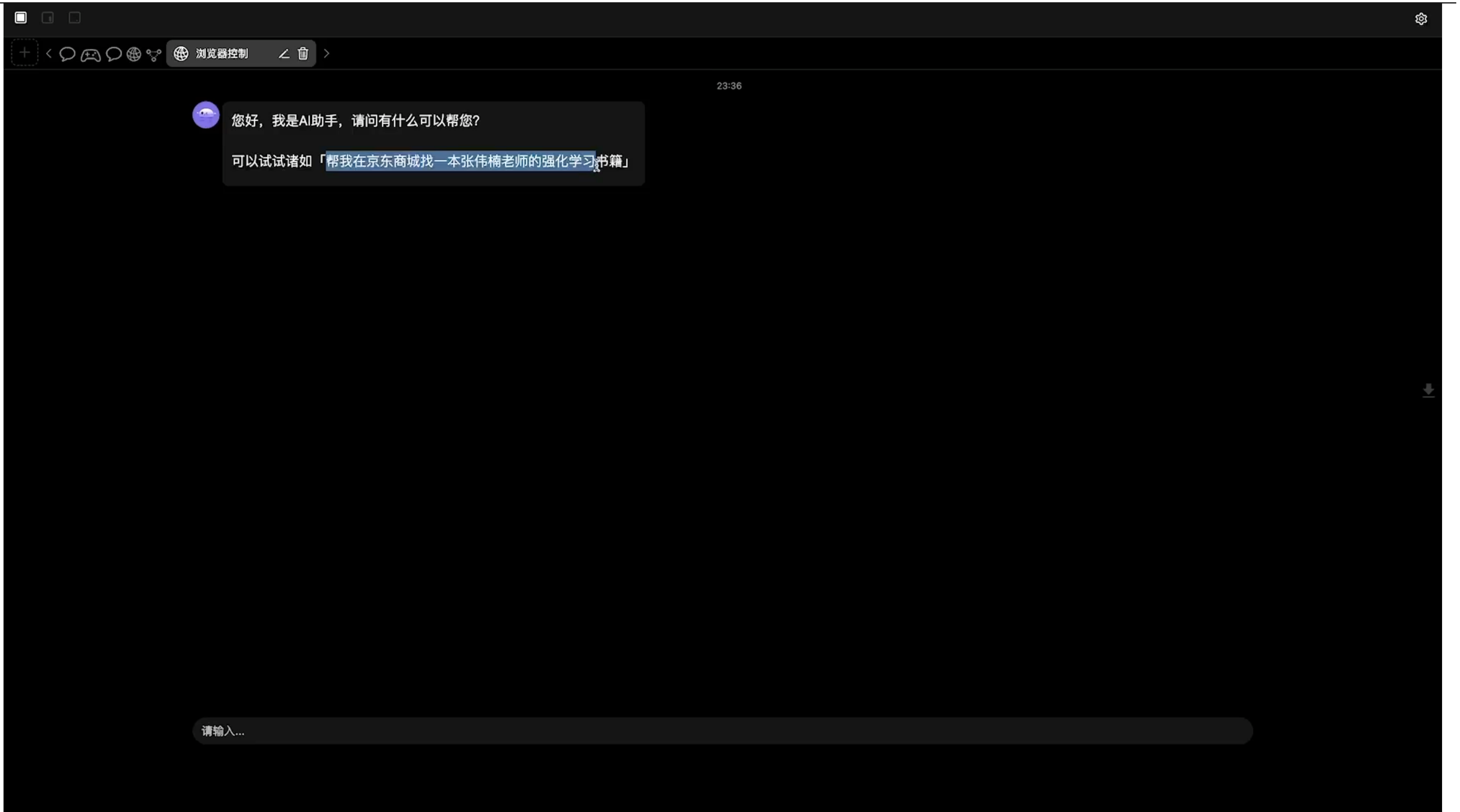
Jiao Tong University - Home Page
en.sjtu.edu.cn/ ▾
International Education. Summer Programs · Undergraduate Programs · Graduate Programs · MIB Program · SEIEE Program · Non-degree Admission · Scholarships · Tuition Payment · How to Apply · International Student Services · Life @ **SJTU** · Global **SJTU** · Learn More » ...
Exchange · Campus Maps · Programs in English · Master

Information item:
Webpage (or document)

- 新范式：通过和互联网做交互动作，最终走到达成information need的information state



AI Agent帮助用户达到目标状态（示例：购买图书）



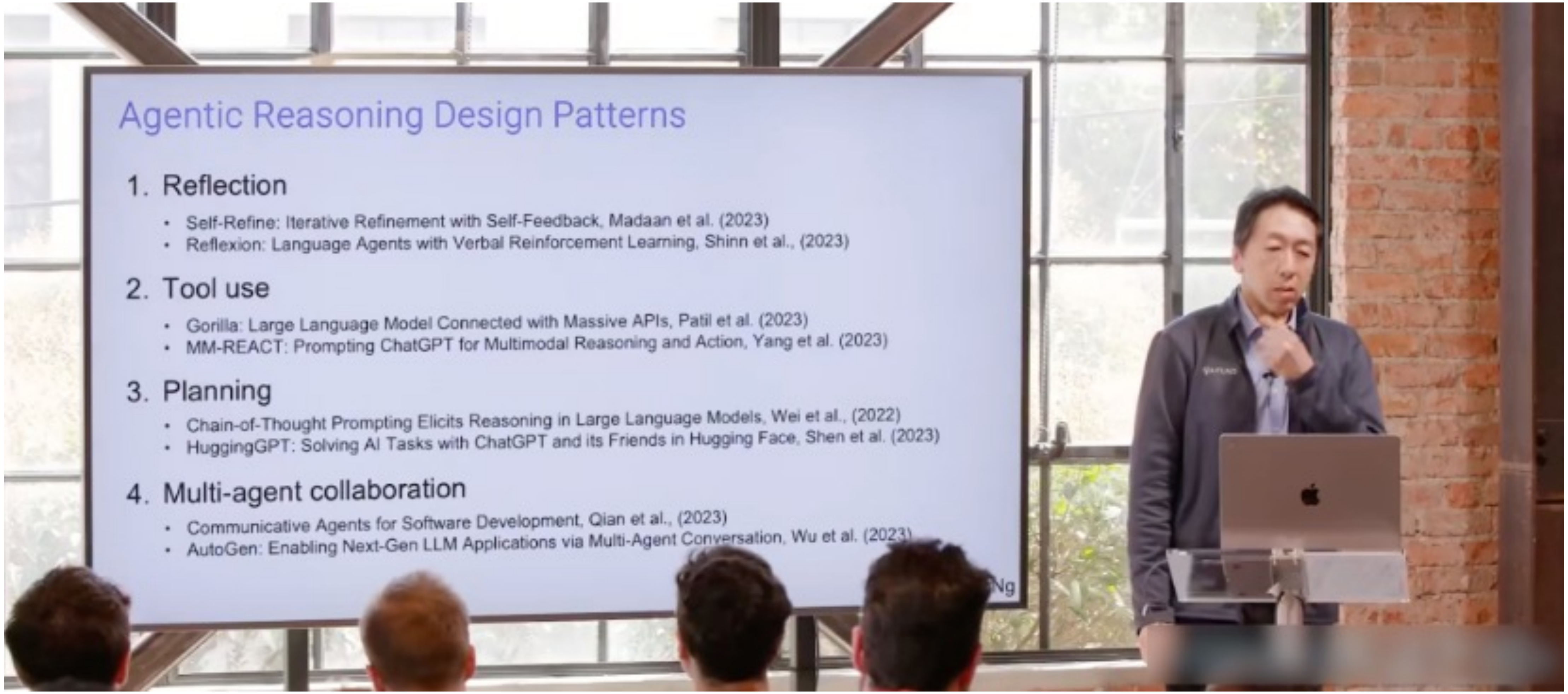
23:36

您好，我是AI助手，请问有什么可以帮您?

可以试试诸如「帮我在京东商城找一本张伟楠老师的强化学习书籍」

范式A：LLM Agent的能力

Agent的能力包括反思、工具使用、任务规划、多智能体协作等

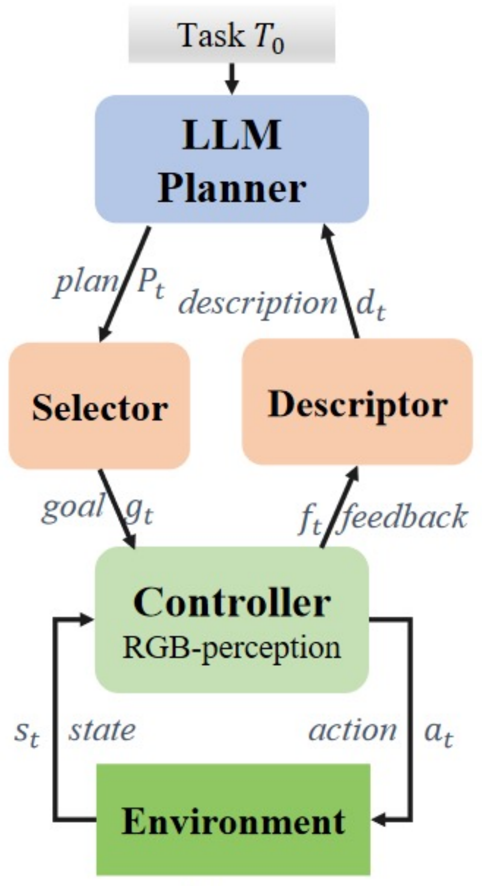


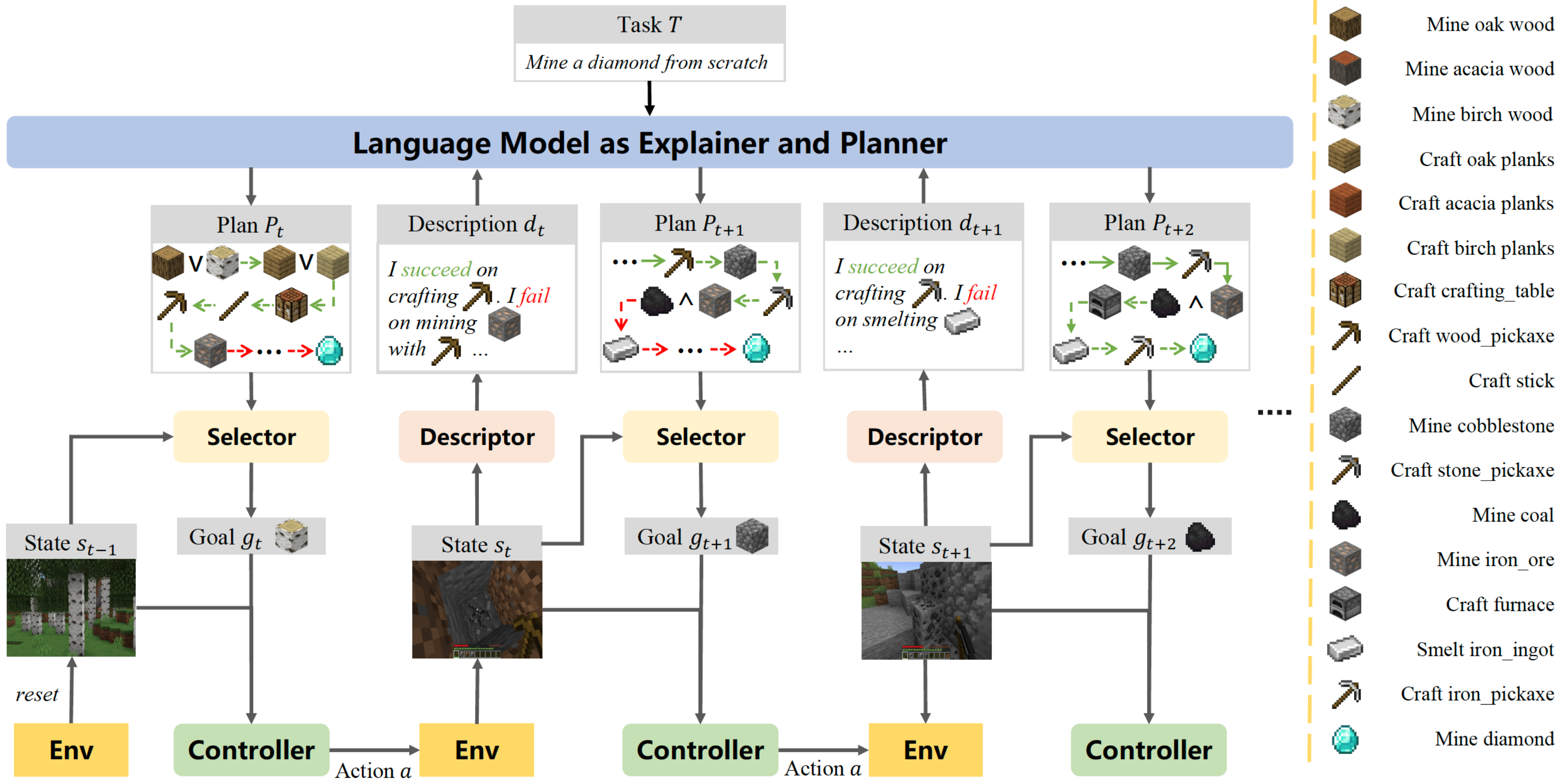
Agentic Reasoning Design Patterns

1. Reflection
   - Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
   - Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)
2. Tool use
   - Gorilla: Large Language Model Connected with Massive APIs, Patil et al. (2023)
   - MM-REACT: Prompting ChatGPT for Multimodal Reasoning and Action, Yang et al. (2023)
3. Planning
   - Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Wei et al., (2022)
   - HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face, Shen et al. (2023)
4. Multi-agent collaboration
   - Communicative Agents for Software Development, Qian et al., (2023)
   - AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation, Wu et al. (2023)

LLM Agent – DEPS

DEPS: describe, explain, plan and select

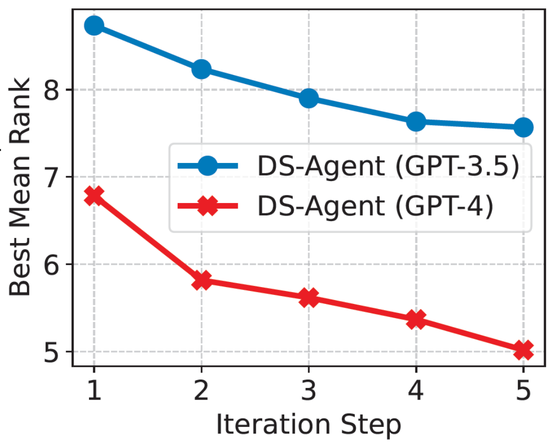


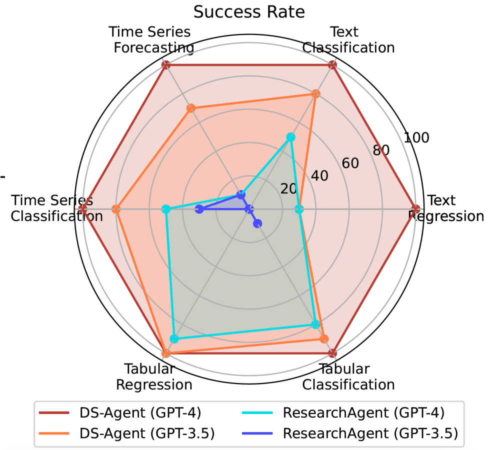


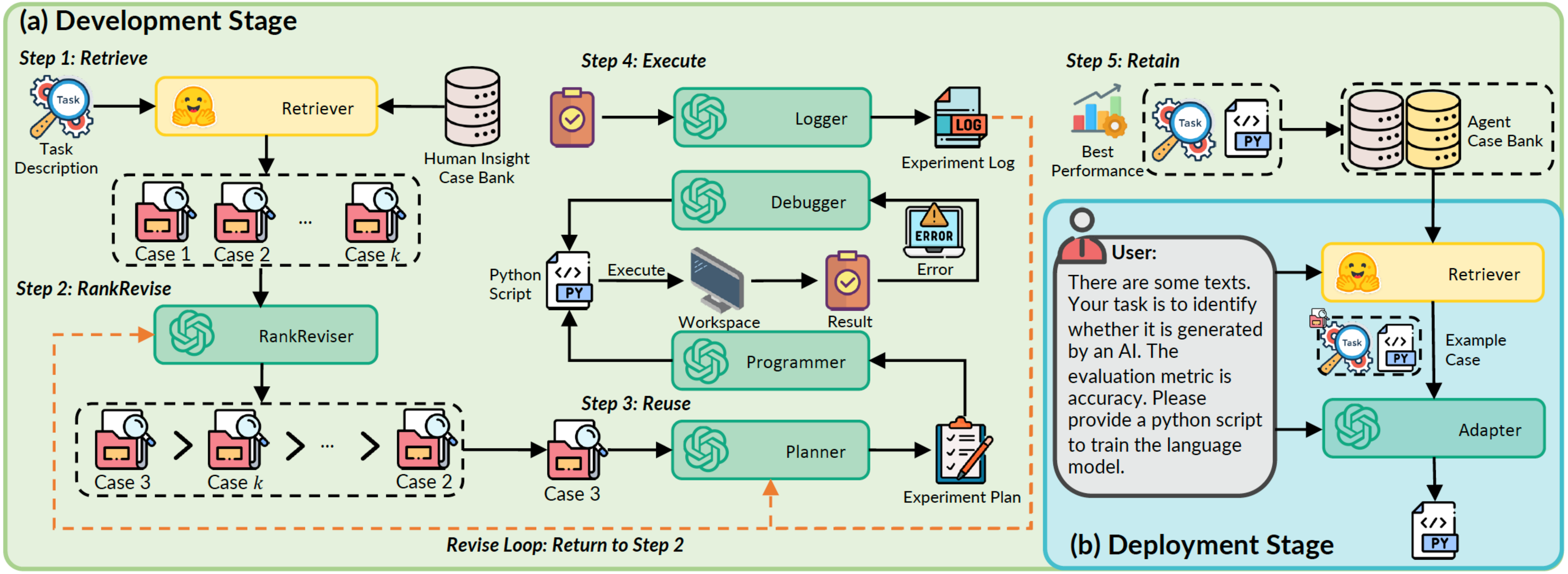
Z. Wang, et al. Describe, Explain, Plan and Select: Interactive Planning with Large Language Models Enables Open-World Multi-Task Agents.” arXiv, 2023.
http://arxiv.org/abs/2302.01560

LLM Agent – DS-Agent

DS-Agent is a novel automatic framework that harnesses LLM agent and case-based reasoning (CBR).



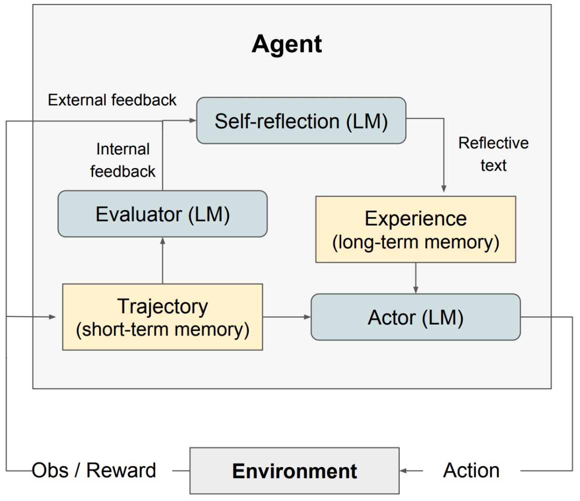


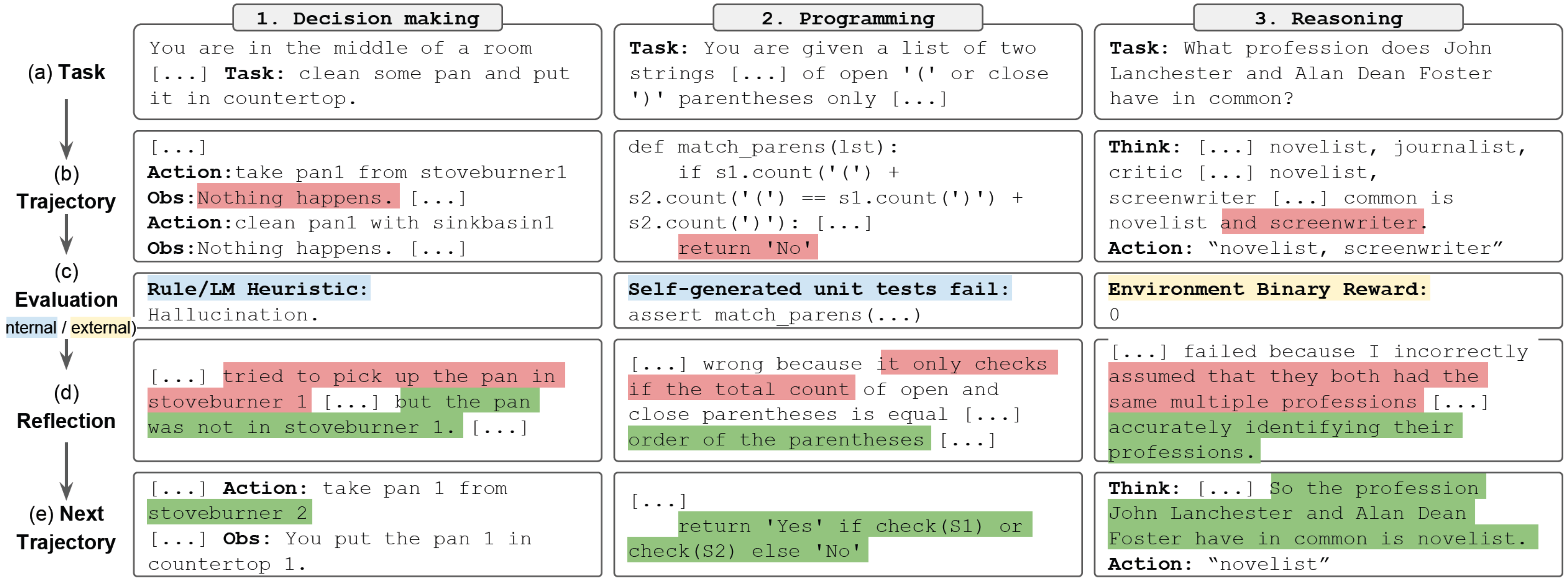


S Guo, et al. DS-Agent: Automated Data Science by Empowering Large Language Models with Case-Based Reasoning. ICML 2024.

LLM Agent - Reflexion

Reflexion agents verbally reflect on task feedback signals, then maintain their own reflective text in an episodic memory buffer to induce better decision-making in subsequent trials.





	1. Decision making	2. Programming	3. Reasoning
(a) **Task**	You are in the middle of a room [...] **Task:** clean some pan and put it in countertop.	**Task:** You are given a list of two strings [...] of open '(' or close ')' parentheses only [...]	**Task:** What profession does John Lanchester and Alan Dean Foster have in common?

(b) **Trajectory**	[...]
	Action:take pan1 from stoveburner1
	Obs: Nothing happens. [...]
	Action:clean pan1 with sinkbasin1
	Obs: Nothing happens. [...]
		`def match_parens(lst):`
		`    if s1.count('(') +`
		`s2.count('(') == s1.count(')') +`
		`s2.count(')'): [...]`
		`    return 'No'`
			Think: [...] novelist, journalist, critic [...] novelist, screenwriter [...] common is novelist and screenwriter.
			Action: "novelist, screenwriter"

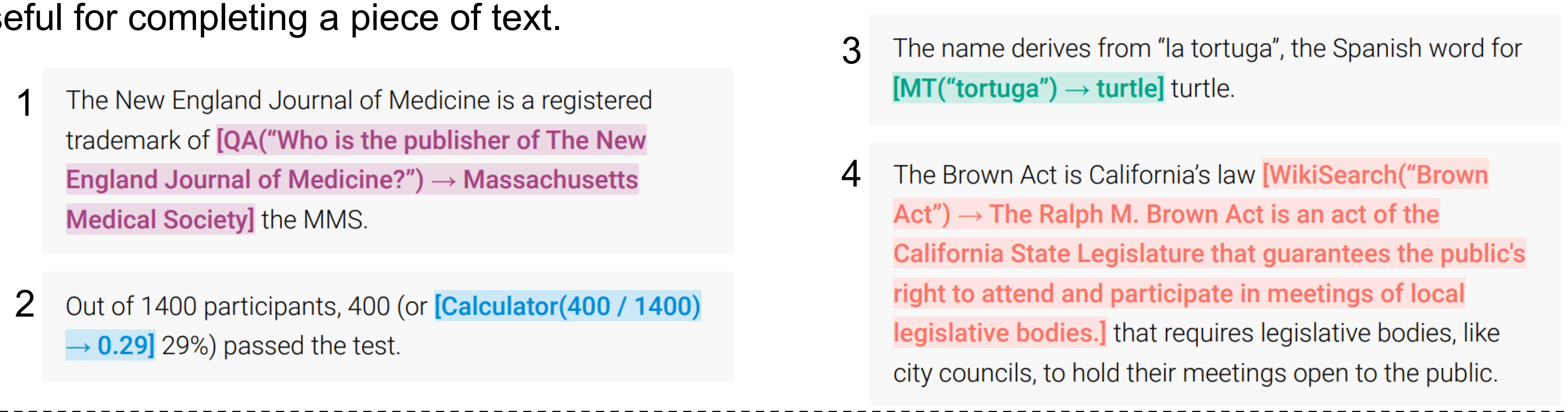
(c) **Evaluation** (internal / external)	**Rule/LM Heuristic:** Hallucination.	**Self-generated unit tests fail:** assert match_parens(...)	**Environment Binary Reward:** 0
(d) **Reflection**	[...] tried to pick up the pan in stoveburner 1 [...] but the pan was not in stoveburner 1. [...]	[...] wrong because it only checks if the total count of open and close parentheses is equal [...] order of the parentheses [...]	[...] failed because I incorrectly assumed that they both had the same multiple professions [...] accurately identifying their professions.

(e) **Next Trajectory**	[...] **Action:** take pan 1 from stoveburner 2
	[...] **Obs:** You put the pan 1 in countertop 1.
		[...]
		`    return 'Yes' if check(S1) or`
		`check(S2) else 'No'`
			Think: [...] So the profession John Lanchester and Alan Dean Foster have in common is novelist.
			Action: "novelist"

N Shinn, et al. Reflexion: Language Agents with Verbal Reinforcement Learning. NeurIPS 2023.

LLM Agent – Toolformer

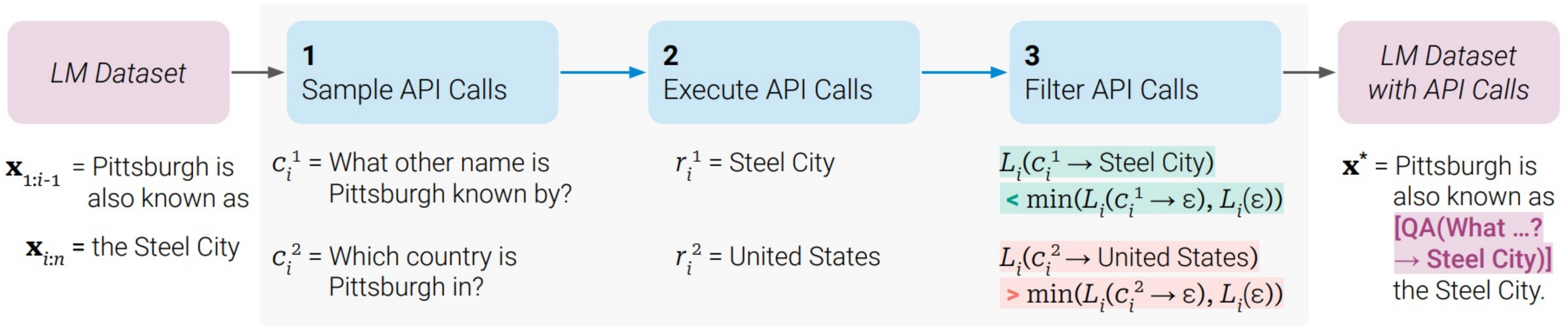
Toolformer autonomously decides to call different APIs (from 1 to 4: a question answering system, a calculator, a machine translation system, and a Wikipedia search engine) to obtain information that is useful for completing a piece of text.


1 The New England Journal of Medicine is a registered trademark of [QA(“Who is the publisher of The New England Journal of Medicine?”) → Massachusetts Medical Society] the MMS.

2 Out of 1400 participants, 400 (or [Calculator(400 / 1400) → 0.29] 29%) passed the test.

3 The name derives from “la tortuga”, the Spanish word for [MT(“tortuga”) → turtle] turtle.

4 The Brown Act is California’s law [WikiSearch(“Brown Act”) → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.] that requires legislative bodies, like city councils, to hold their meetings open to the public.



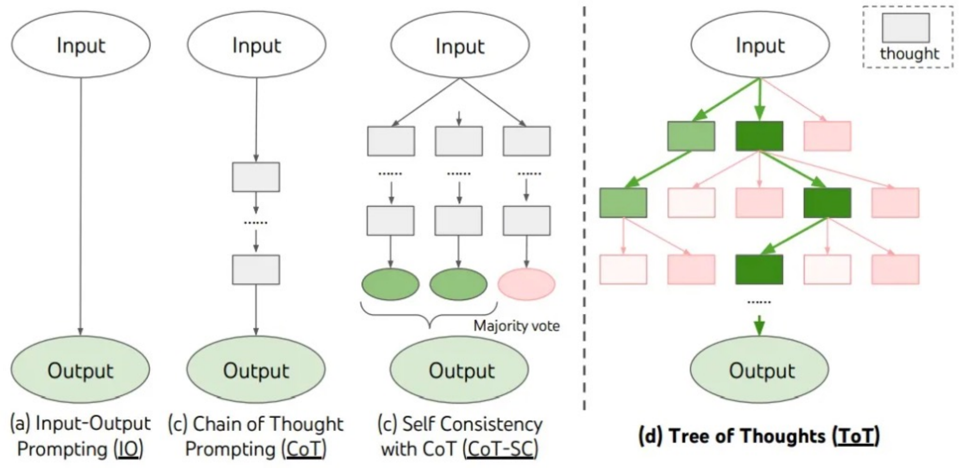
LM Dataset
1 Sample API Calls
2 Execute API Calls
3 Filter API Calls
LM Dataset with API Calls
$\mathbf{x}_{1:i-1}$ = Pittsburgh is also known as
$\mathbf{x}_{i:n}$ = the Steel City
c_i^1 = What other name is Pittsburgh known by?
c_i^2 = Which country is Pittsburgh in?
r_i^1 = Steel City
r_i^2 = United States
$L_i(c_i^1 \rightarrow$ Steel City$) < \min(L_i(c_i^1 \rightarrow \varepsilon), L_i(\varepsilon))$
$L_i(c_i^2 \rightarrow$ United States$) > \min(L_i(c_i^2 \rightarrow \varepsilon), L_i(\varepsilon))$
$\mathbf{x}^*$ = Pittsburgh is also known as [QA(What …? → Steel City)] the Steel City.

Timo Schick et al. Toolformer: Language Models Can Teach Themselves to Use Tools. NeurIPS 2023.

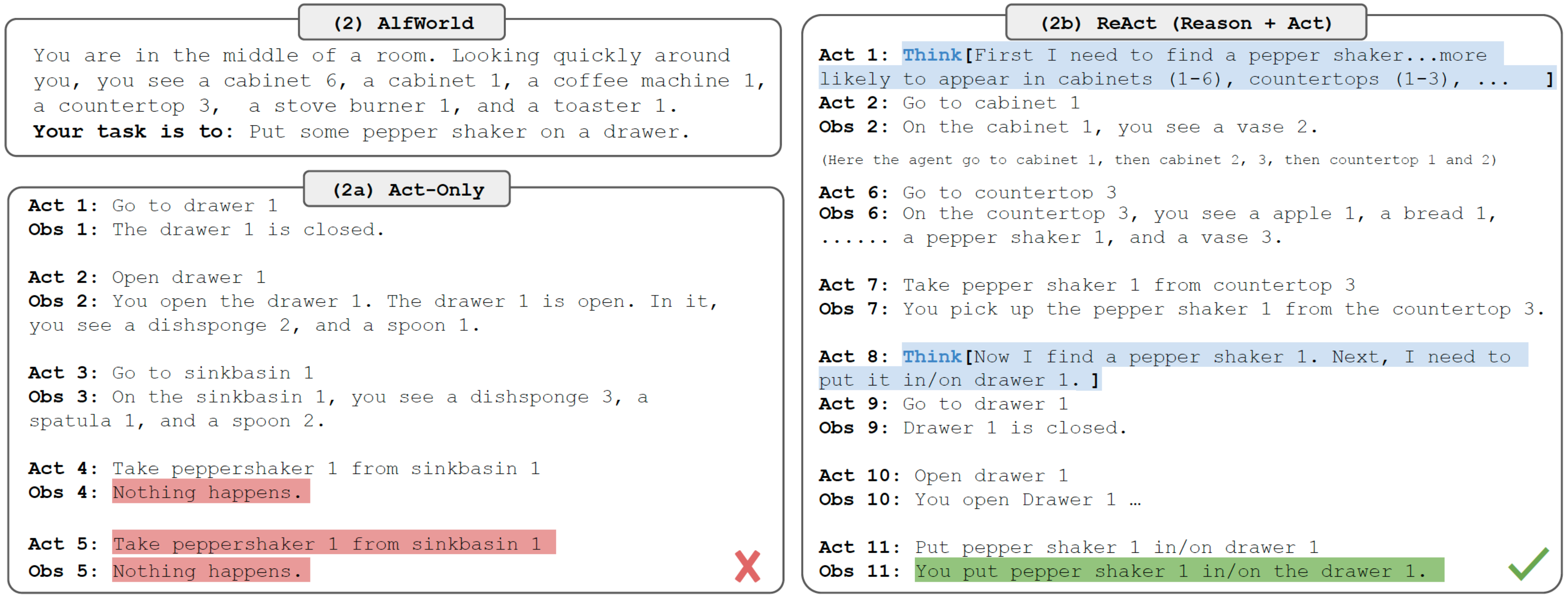
LLM Agent – ReAct

思维是智能体内部生成的信号，不直接影响交互环境；基于思维和当前状态，智能体产生对应的动作，进而更好地完成任务

- 思维链 (Chain-of-Thought)、思维树 (Tree-of-Thought)



(a) Input-Output Prompting (IO) (c) Chain of Thought Prompting (CoT) (c) Self Consistency with CoT (CoT-SC) (d) Tree of Thoughts (ToT)



(2) AlfWorld

You are in the middle of a room. Looking quickly around you, you see a cabinet 6, a cabinet 1, a coffee machine 1, a countertop 3, a stove burner 1, and a toaster 1.
Your task is to: Put some pepper shaker on a drawer.

(2a) Act-Only

Act 1: Go to drawer 1
Obs 1: The drawer 1 is closed.

Act 2: Open drawer 1
Obs 2: You open the drawer 1. The drawer 1 is open. In it, you see a dishsponge 2, and a spoon 1.

Act 3: Go to sinkbasin 1
Obs 3: On the sinkbasin 1, you see a dishsponge 3, a spatula 1, and a spoon 2.

Act 4: Take peppershaker 1 from sinkbasin 1
Obs 4: Nothing happens.

Act 5: Take peppershaker 1 from sinkbasin 1
Obs 5: Nothing happens. ✗

(2b) ReAct (Reason + Act)

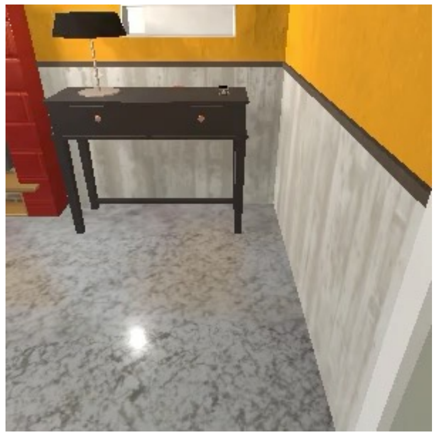
Act 1: Think[First I need to find a pepper shaker...more likely to appear in cabinets (1-6), countertops (1-3), ...]
Act 2: Go to cabinet 1
Obs 2: On the cabinet 1, you see a vase 2.
(Here the agent go to cabinet 1, then cabinet 2, 3, then countertop 1 and 2)
Act 6: Go to countertop 3
Obs 6: On the countertop 3, you see a apple 1, a bread 1, ... a pepper shaker 1, and a vase 3.

Act 7: Take pepper shaker 1 from countertop 3
Obs 7: You pick up the pepper shaker 1 from the countertop 3.

Act 8: Think[Now I find a pepper shaker 1. Next, I need to put it in/on drawer 1.]
Act 9: Go to drawer 1
Obs 9: Drawer 1 is closed.

Act 10: Open drawer 1
Obs 10: You open Drawer 1 …

Act 11: Put pepper shaker 1 in/on drawer 1
Obs 11: You put pepper shaker 1 in/on the drawer 1. ✓



ALFworld

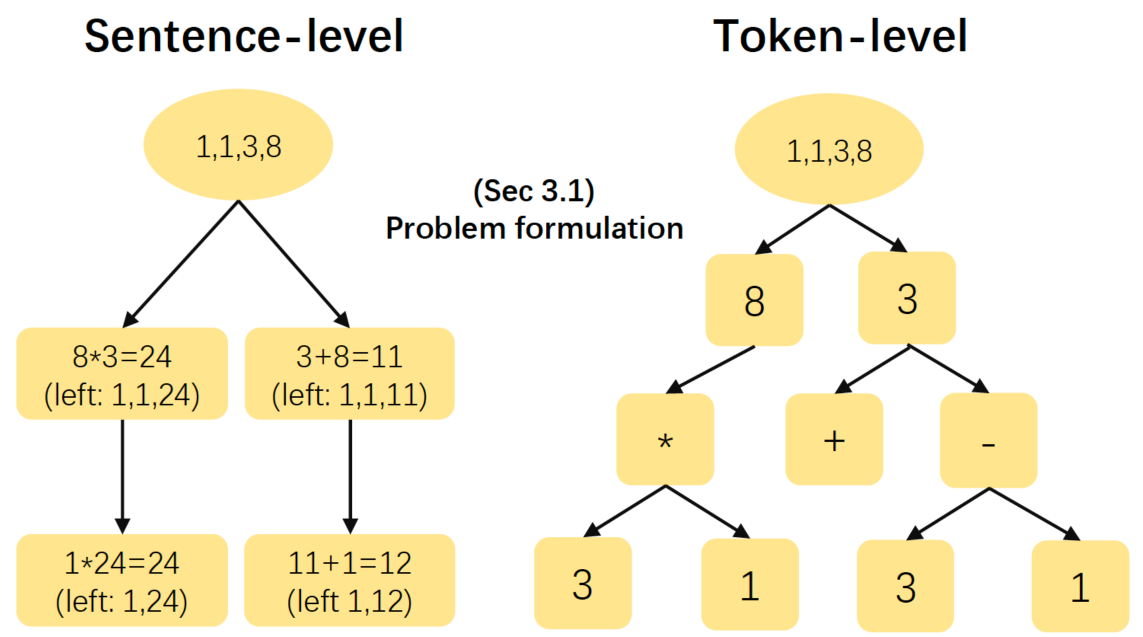
Shunyu Yao et al. ReAct: Synergizing Reasoning and Acting in Language Models. ICLR 2023.

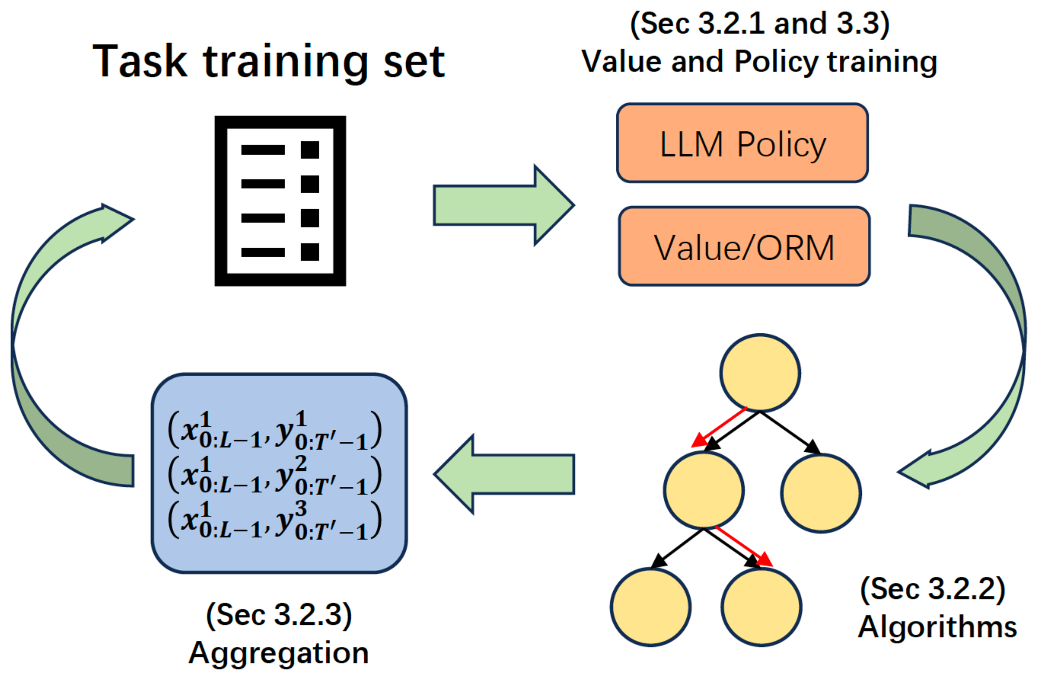
LLM Agent - LLM-TS

将句子或者token看成思维空间中的节点，由此可以在认知空间做蒙特卡洛树搜索

- 训练每个节点的价值评估函数
- 此价值函数可以更好地指导大语言模型策略做训练和动作执行

Task	Policy	Value	Accuracy(%)
GSM8K	GPT-3.5	GPT-3.5 (ToT)	72.7
	GPT-3.5	LLaMA-V (Ours)	**74.0**
	LLaMA-SFT	LLaMA (ToT)	37.4
	LLaMA-SFT	GPT-3.5 (ToT)	45.8
	LLaMA-SFT	LLaMA-V (Ours)	**52.5**
Game24	GPT-3.5	GPT-3.5 (ToT)	15.5
	GPT-3.5	LLaMA-V (Ours)	**19.1**
	LLaMA-SFT	LLaMA (ToT)	9.2
	LLaMA-SFT	GPT-3.5 (ToT)	21.0
	LLaMA-SFT	LLaMA-V (Ours)	**64.8**

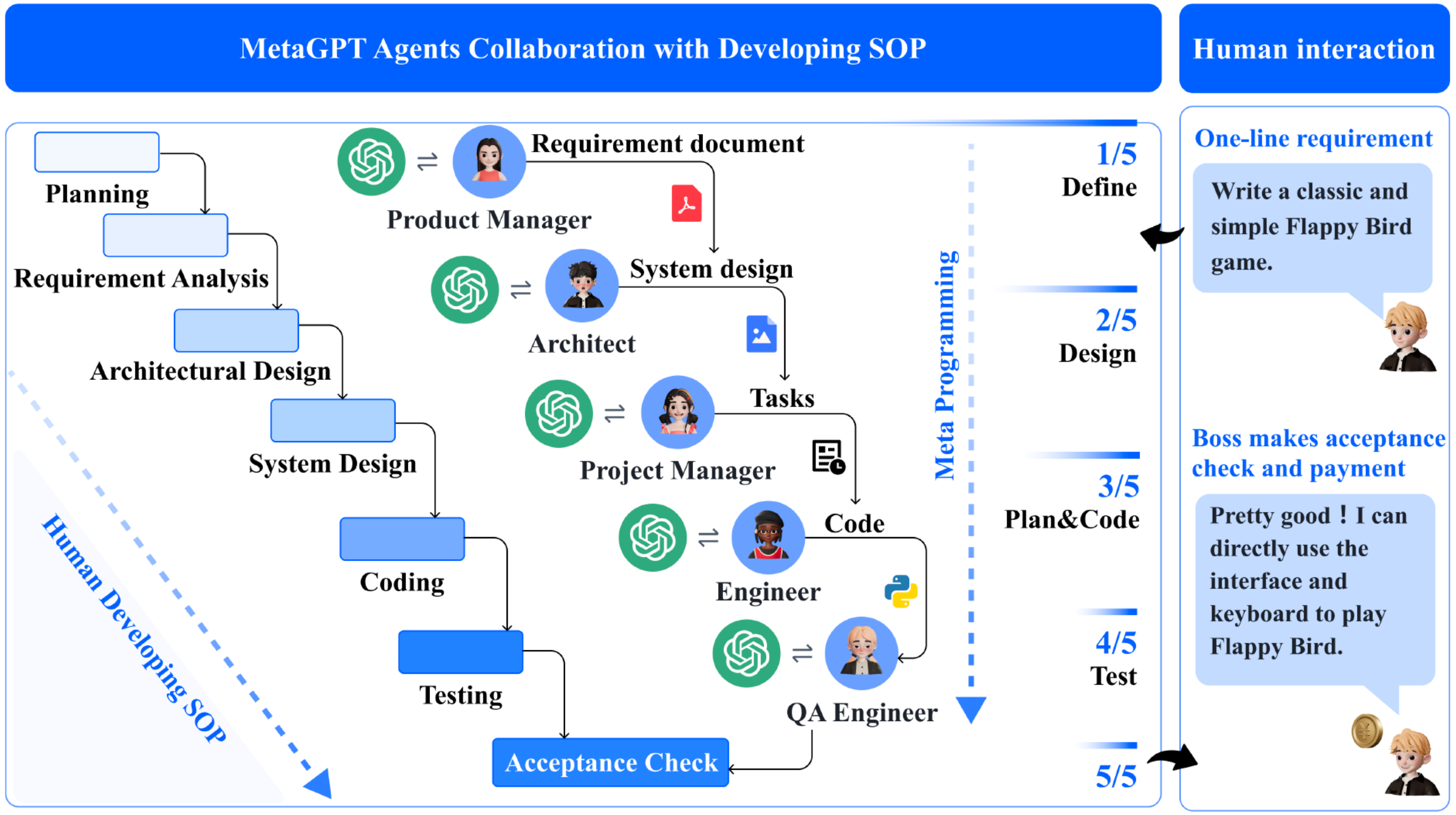


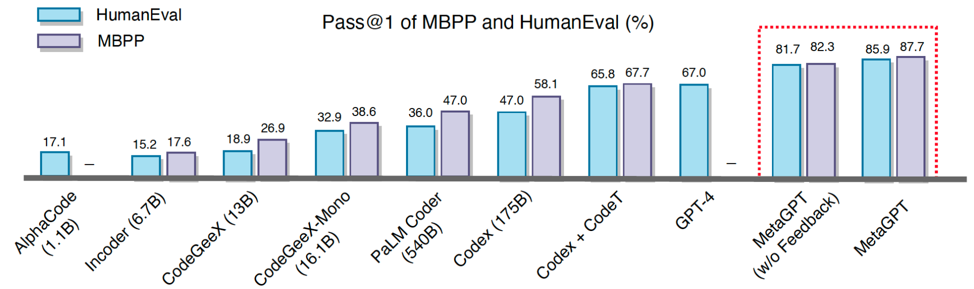

Task training set
(Sec 3.2.1 and 3.3)
Value and Policy training
LLM Policy
Value/ORM
$(x^1_{0:L-1}, y^1_{0:T'-1})$
$(x^1_{0:L-1}, y^2_{0:T'-1})$
$(x^1_{0:L-1}, y^3_{0:T'-1})$
(Sec 3.2.3)
Aggregation
(Sec 3.2.2)
Algorithms

Wan Z, Feng X, Wen M, et al. Alphazero-like tree-search can guide large language model decoding and training. ICML 2024.

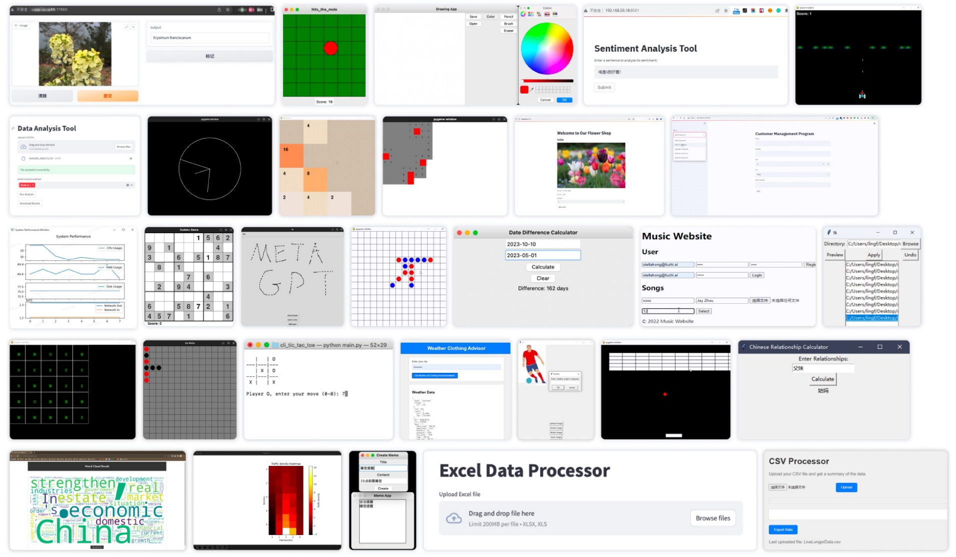
LLM Agent – MetaGPT

Multi-agent framework MetaGPT encodes Standardized Operating Procedures (SOPs) into prompt sequences for more streamlined workflows





MetaGPT generates more coherent solutions than previous chat-based multi-agent systems



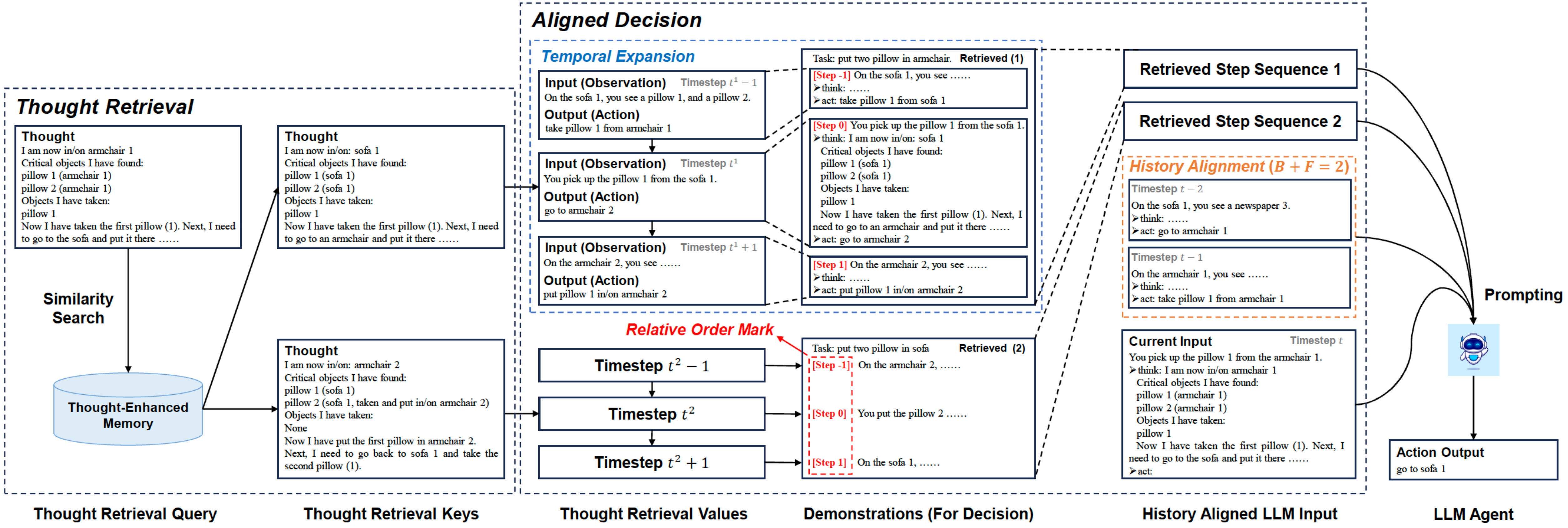
MetaGPT designs different types of games

S. Hong et al. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. Arixv 2023. https://arxiv.org/abs/2308.00352

LLM Agent - TRAD

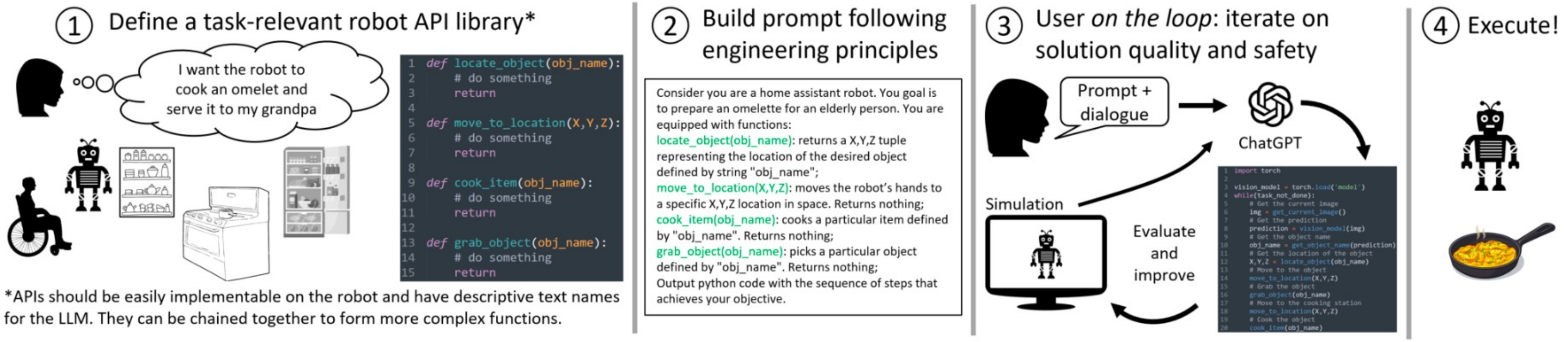
使用历史中类似的成功轨迹数据，在思维链过程中检索类似的思维点，并对齐Agent在轨迹上的动作，以增强LLM Agent在目前步应该采取的动作性能。

Method		ReAct-RD	ReAct-RV	TR	TRAD (Ours)
Website 1 (form filling)	**Step SR**	0.843	0.826	0.941	**0.950**
	SR	0.500	0.450	**0.800**	**0.800**
Website 2 (advanced IR)	**Step SR**	0.941	0.937	0.958	**0.974**
	SR	**0.900**	0.850	0.850	**0.900**
Website 3 (advanced IR)	**Step SR**	0.962	0.987	**1.000**	**1.000**
	SR	0.850	0.800	0.850	**1.000**
Website 4 (form filling)	**Step SR**	0.820	0.860	0.845	**1.000**
	SR	0.350	0.350	0.400	**1.000**
Average	**Step SR**	0.891	0.902	0.936	**0.981**
	SR	0.650	0.613	0.725	**0.925**


Thought Retrieval
Thought
I am now in/on armchair 1
Critical objects I have found:
pillow 1 (armchair 1)
pillow 2 (armchair 1)
Objects I have taken:
pillow 1
Now I have taken the first pillow (1). Next, I need to go to the sofa and put it there ……
Similarity Search
Thought-Enhanced Memory
Thought
I am now in/on: sofa 1
Critical objects I have found:
pillow 1 (sofa 1)
pillow 2 (sofa 1)
Objects I have taken:
pillow 1
Now I have taken the first pillow (1). Next, I need to go to an armchair and put it there ……
Thought
I am now in/on: armchair 2
Critical objects I have found:
pillow 1 (sofa 1)
pillow 2 (sofa 1, taken and put in/on armchair 2)
Objects I have taken:
None
Now I have put the first pillow in armchair 2. Next, I need to go back to sofa 1 and take the second pillow (1).
Aligned Decision
Temporal Expansion
Input (Observation) Timestep $t^1 - 1$
On the sofa 1, you see a pillow 1, and a pillow 2.
Output (Action)
take pillow 1 from armchair 1
Input (Observation) Timestep t^1
You pick up the pillow 1 from the sofa 1.
Output (Action)
go to armchair 2
Input (Observation) Timestep $t^1 + 1$
On the armchair 2, you see ……
Output (Action)
put pillow 1 in/on armchair 2
Task: put two pillow in armchair. Retrieved (1)
[Step -1] On the sofa 1, you see ……
➢think: ……
➢act: take pillow 1 from sofa 1
[Step 0] You pick up the pillow 1 from the sofa 1.
➢think: I am now in/on: sofa 1
Critical objects I have found:
pillow 1 (sofa 1)
pillow 2 (sofa 1)
Objects I have taken:
pillow 1
Now I have taken the first pillow (1). Next, I need to go to an armchair and put it there ……
➢act: go to armchair 2
[Step 1] On the armchair 2, you see ……
➢think: ……
➢act: put pillow 1 in/on armchair 2
Relative Order Mark
Timestep $t^2 - 1$
Timestep t^2
Timestep $t^2 + 1$
Task: put two pillow in sofa Retrieved (2)
[Step -1] On the armchair 2, ……
[Step 0] You put the pillow 2 ……
[Step 1] On the sofa 1, ……
Retrieved Step Sequence 1
Retrieved Step Sequence 2
History Alignment ($B + F = 2$)
Timestep $t - 2$
On the sofa 1, you see a newspaper 3.
➢think: ……
➢act: go to armchair 1
Timestep $t - 1$
On the armchair 1, you see ……
➢think: ……
➢act: take pillow 1 from armchair 1
Current Input Timestep t
You pick up the pillow 1 from the armchair 1.
➢think: I am now in/on armchair 1
Critical objects I have found:
pillow 1 (armchair 1)
pillow 2 (armchair 1)
Objects I have taken:
pillow 1
Now I have taken the first pillow (1). Next, I need to go to the sofa and put it there ……
➢act:
Prompting
Action Output
go to sofa 1
Thought Retrieval Query
Thought Retrieval Keys
Thought Retrieval Values
Demonstrations (For Decision)
History Aligned LLM Input
LLM Agent

R. Zhou et al. TRAD: Enhancing LLM Agents with Step-Wise Thought Retrieval and Aligned Decision. SIGIR 2024. https://arxiv.org/abs/2403.06221

LLM Agent – ChatGPT for Robotics

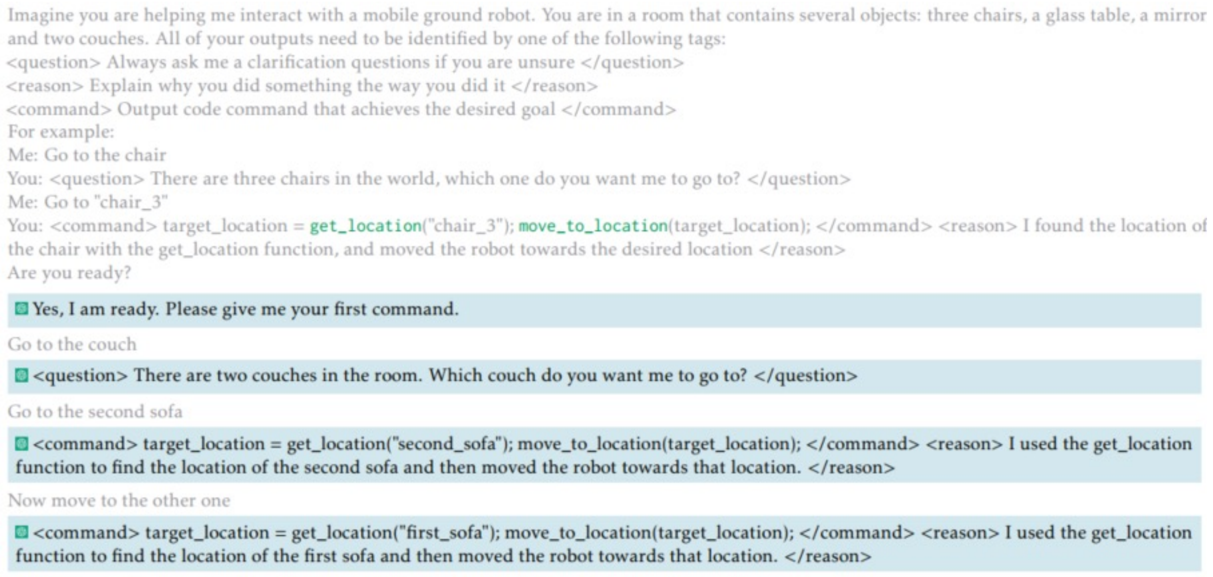

1 Define a task-relevant robot API library*
I want the robot to cook an omelet and serve it to my grandpa
2 Build prompt following engineering principles
3 User *on the loop*: iterate on solution quality and safety
Prompt + dialogue
ChatGPT
Simulation
Evaluate and improve
4 Execute!

```python
def locate_object(obj_name):
    # do something
    return

def move_to_location(X,Y,Z):
    # do something
    return

def cook_item(obj_name):
    # do something
    return

def grab_object(obj_name):
    # do something
    return
```

*APIs should be easily implementable on the robot and have descriptive text names for the LLM. They can be chained together to form more complex functions.

Consider you are a home assistant robot. You goal is to prepare an omelette for an elderly person. You are equipped with functions:
locate_object(obj_name): returns a X,Y,Z tuple representing the location of the desired object defined by string "obj_name";
move_to_location(X,Y,Z): moves the robot's hands to a specific X,Y,Z location in space. Returns nothing;
cook_item(obj_name): cooks a particular item defined by "obj_name". Returns nothing;
grab_object(obj_name): picks a particular object defined by "obj_name". Returns nothing;
Output python code with the sequence of steps that achieves your objective.

```python
import torch

vision_model = torch.load('model')
while(task_not_done):
    # Get the current image
    img = get_current_image()
    # Get the prediction
    prediction = vision_model(img)
    # Get the object name
    obj_name = get_object_name(prediction)
    # Get the location of the object
    X,Y,Z = locate_object(obj_name)
    # Move to the object
    move_to_location(X,Y,Z)
    # Grab the object
    grab_object(obj_name)
    # Move to the cooking station
    move_to_location(X,Y,Z)
    # Cook the object
    cook_item(obj_name)
```



Imagine you are helping me interact with a mobile ground robot. You are in a room that contains several objects: three chairs, a glass table, a mirror and two couches. All of your outputs need to be identified by one of the following tags:
<question> Always ask me a clarification questions if you are unsure </question>
<reason> Explain why you did something the way you did it </reason>
<command> Output code command that achieves the desired goal </command>
For example:
Me: Go to the chair
You: <question> There are three chairs in the world, which one do you want me to go to? </question>
Me: Go to "chair_3"
You: <command> target_location = get_location("chair_3"); move_to_location(target_location); </command> <reason> I found the location of the chair with the get_location function, and moved the robot towards the desired location </reason>
Are you ready?

Yes, I am ready. Please give me your first command.

Go to the couch

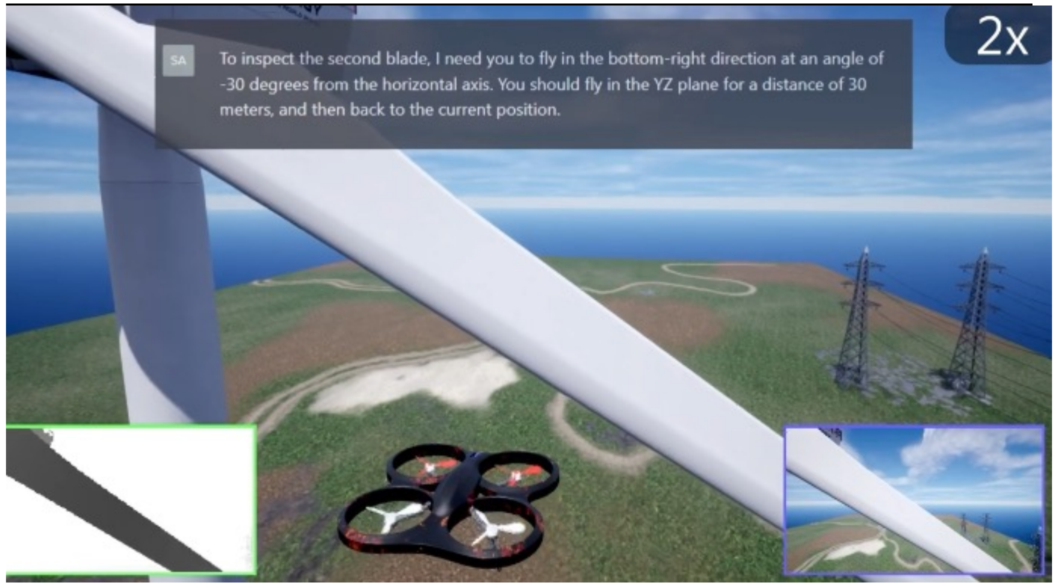
<question> There are two couches in the room. Which couch do you want me to go to? </question>

Go to the second sofa

<command> target_location = get_location("second_sofa"); move_to_location(target_location); </command> <reason> I used the get_location function to find the location of the second sofa and then moved the robot towards that location. </reason>

Now move to the other one

<command> target_location = get_location("first_sofa"); move_to_location(target_location); </command> <reason> I used the get_location function to find the location of the first sofa and then moved the robot towards that location. </reason>

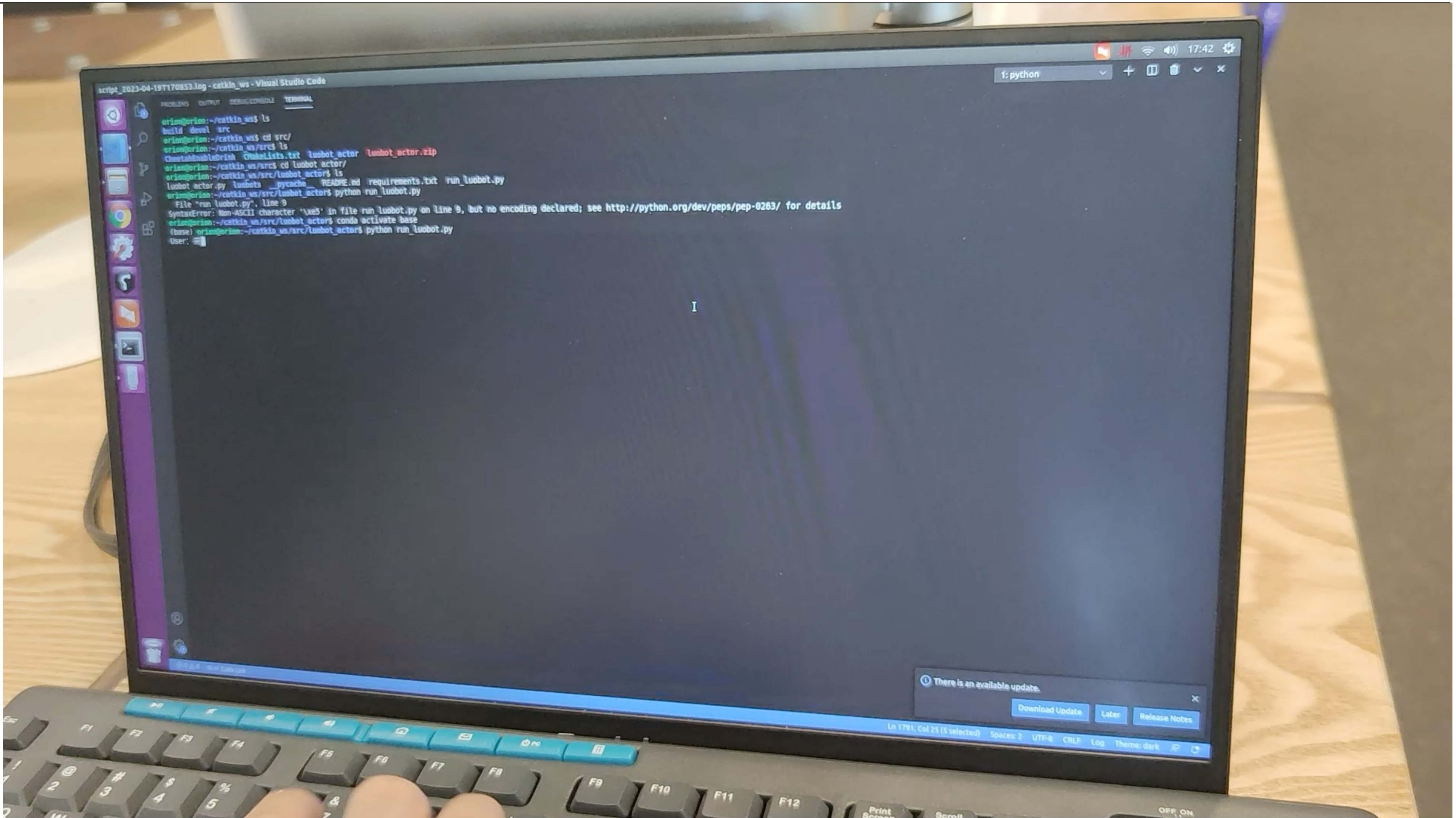


To inspect the second blade, I need you to fly in the bottom-right direction at an angle of -30 degrees from the horizontal axis. You should fly in the YZ plane for a distance of 30 meters, and then back to the current position.

Sai Vemprala et al. ChatGPT for Robotics: Design Principles and Model Abilities. Arxiv 2023.

LLM Agent 可以连接机器人感知和控制

示例：咖啡机器人

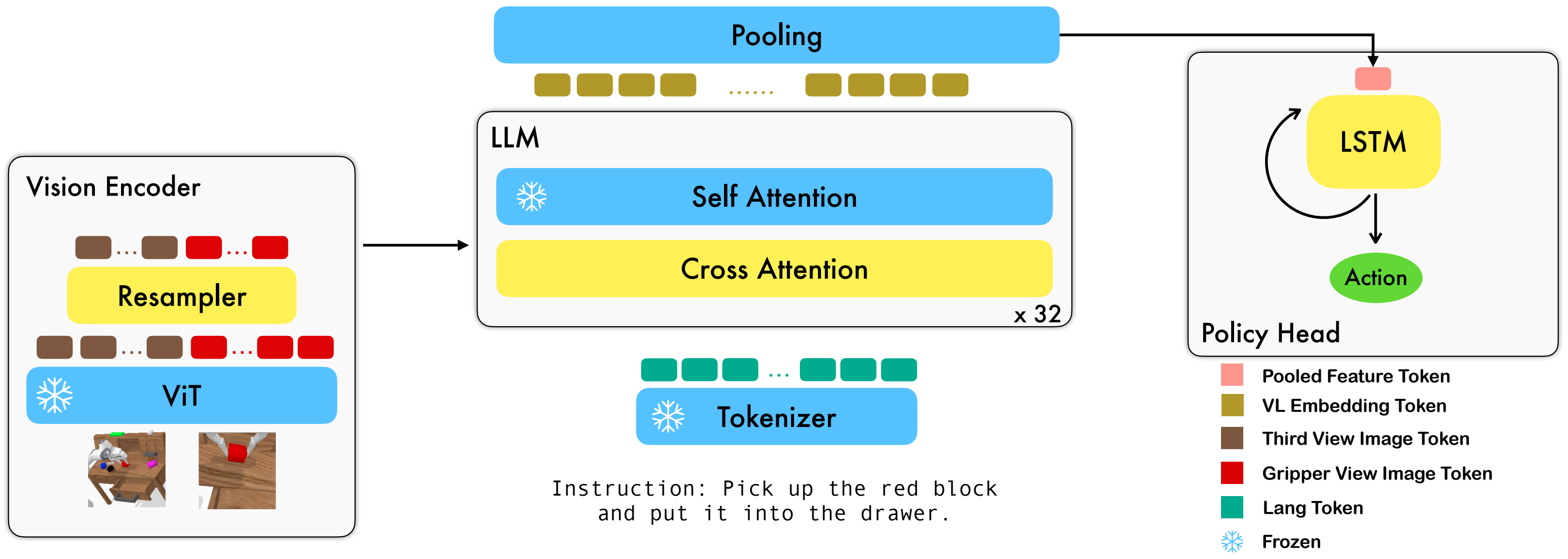

script_2023-04-19T170853.log - catkin_ws - Visual Studio Code
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: python
17:42
There is an available update.
Download Update
Later
Release Notes

```
orion@orion:~/catkin_ws$ ls
build  devel  src
orion@orion:~/catkin_ws$ cd src/
orion@orion:~/catkin_ws/src$ ls
CheetahEnableDrink  CMakeLists.txt  luobot_actor  luobot_actor.zip
orion@orion:~/catkin_ws/src$ cd luobot_actor/
orion@orion:~/catkin_ws/src/luobot_actor$ ls
luobot_actor.py  luobots  __pycache__  README.md  requirements.txt  run_luobot.py
orion@orion:~/catkin_ws/src/luobot_actor$ python run_luobot.py
  File "run_luobot.py", line 9
SyntaxError: Non-ASCII character '\xe5' in file run_luobot.py on line 9, but no encoding declared; see http://python.org/dev/peps/pep-0263/ for details
orion@orion:~/catkin_ws/src/luobot_actor$ conda activate base
(base) orion@orion:~/catkin_ws/src/luobot_actor$ python run_luobot.py
User: 
```

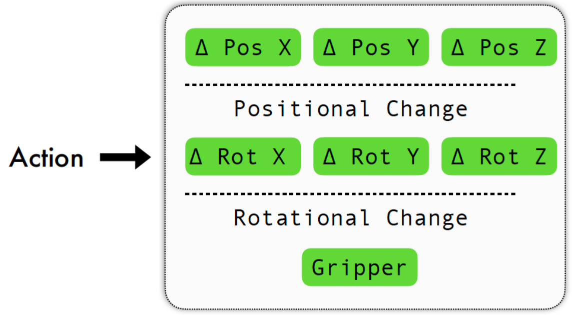
LLM Agent – RoboFlamingo

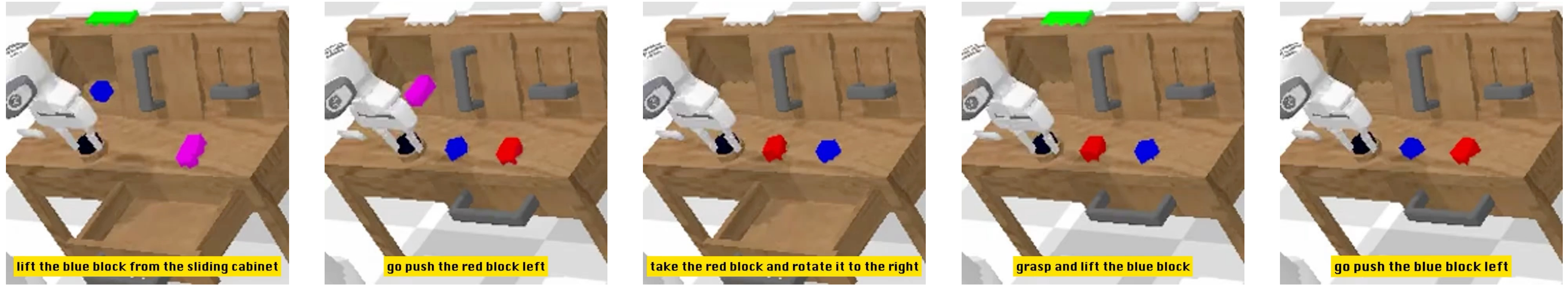
Up to 9B parameters

- 考虑能处理多模态数据的vision language model来做机器人决策控制
- 以开源VLM OpenFlamingo为基座，加入action head并在少量机器人控制数据上微调



$$\ell = \sum_t \mathrm{MSE}(a_t^{pose}, \hat{a}_t^{pose}) + \lambda_{gripper}\mathrm{BCE}(a_t^{gripper}, \hat{a}_t^{gripper})$$



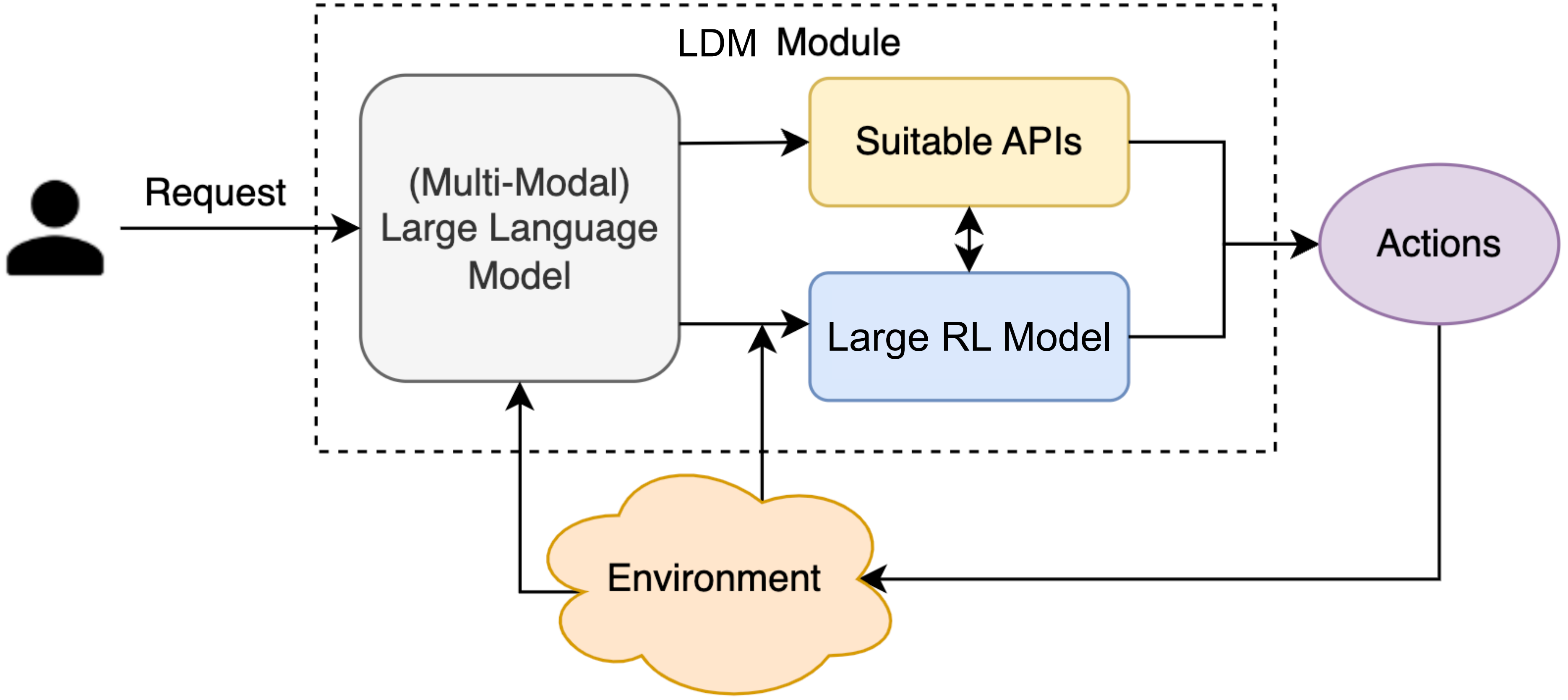


lift the blue block from the sliding cabinet
go push the red block left
take the red block and rotate it to the right
grasp and lift the blue block
go push the blue block left

Xinghang Li et al. Vision-Language Foundation Models as Effective Robot Imitators. ICLR 2024.

决策大模型的两种范式

- ◘ 范式A（LLM Agent）：以大语言模型为中心，配合API选择的决策
- ◘ 范式B（Large RL Models）：以强化学习大模型为中心，做底层的决策控制

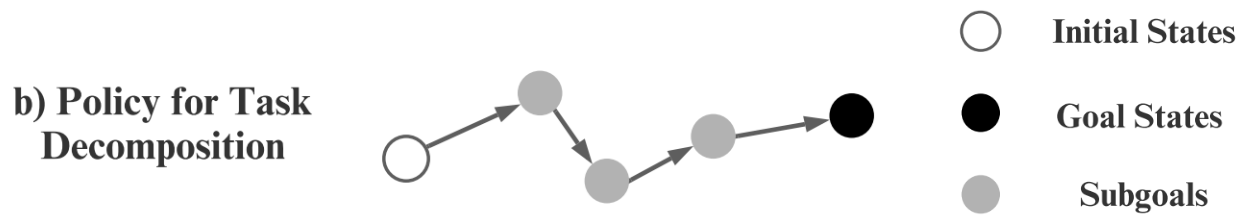


范式B：Large RL Models以强化学习大模型为中心，做底层的决策控制

没有具体的API可以正好解决底层决策，需要使用强化学习策略直接泛化

- 一个想法：如果把每次任务的终点以一种目标信息显式加到智能体的输入和奖励信号中，是否能训练出能完成不同终点的任务呢？

- 基于目标的强化学习（Goal-Conditioned RL，GCRL）
  - 在MDP环境中额外定义一个目标，将此目标显式地告诉智能体，并在训练过程中使用此目标信息，进而在测试阶段给定新的目标，使得智能体能直接泛化来解决此任务。

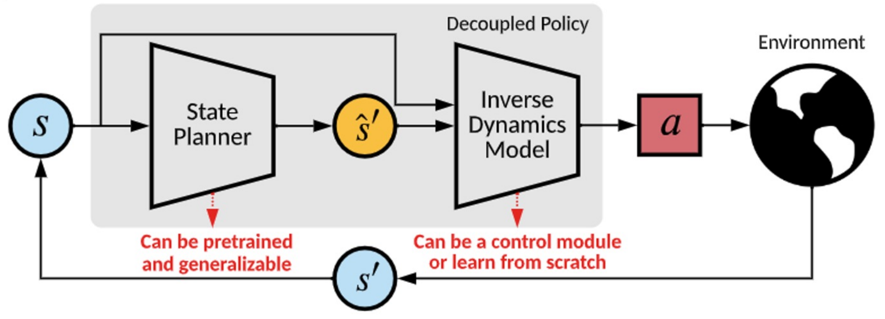


Minghuan Liu, Menghui Zhu, Weinan Zhang. Goal-Conditioned Reinforcement Learning: Problems and Solutions. IJCAI 2022.

- 实现框架：把完成任务的轨迹分拆成序列的任务和完成具体任务的技能

- 由此可以解耦：
  - 任务子目标规划 $p(s'|s)$
  - 达到子目标的技能 $q(a|s,s')$

$$\pi = \underbrace{\mathcal{T}_{\pi}^{-1}}_{\text{inverse dynamics}} \left(\underbrace{\mathcal{T}(\pi_E)}_{\text{state planner}} \right)$$


Decoupled Policy
Environment
s
State Planner
ŝ'
Inverse Dynamics Model
a
s'
Can be pretrained and generalizable
Can be a control module or learn from scratch

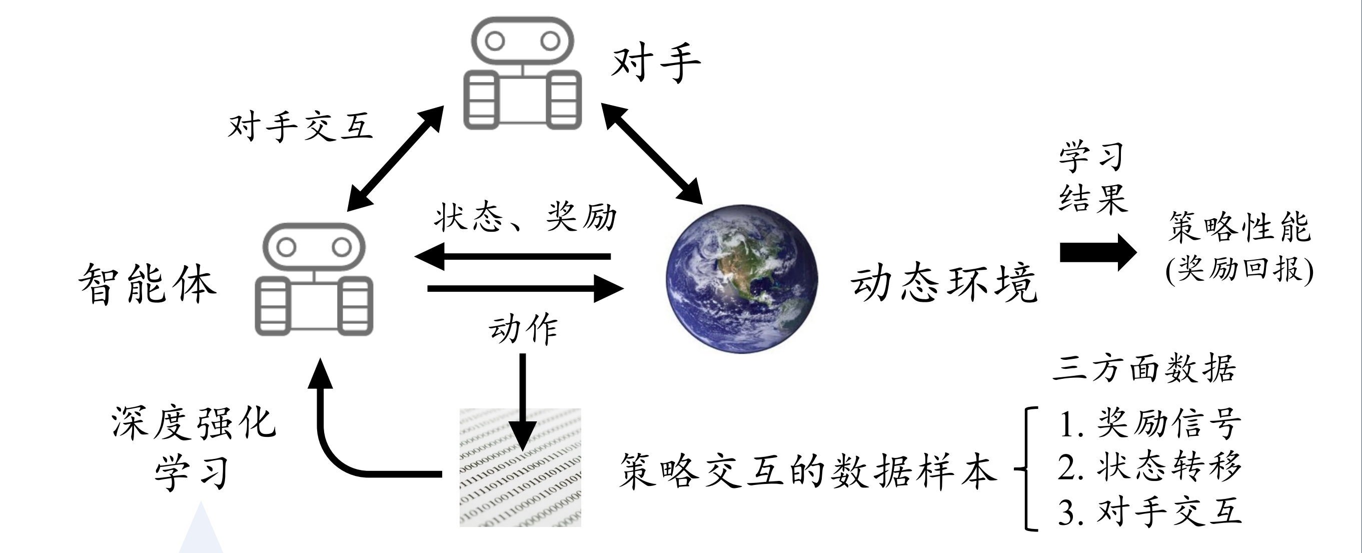
Minghuan Liu et al. Plan Your Target and Learn Your Skills: Transferable State-Only Imitation Learning via Decoupled Policy Optimization. ICML 2022.

范式B：Large RL Models以强化学习大模型为中心，做底层的决策控制

没有具体的API可以正好解决底层决策，需要使用强化学习策略直接泛化

◻ 强化学习大模型

1. 策略大模型
   - Gato、DB1、AdA
2. 价值大模型
   - Q-Transformer
3. 环境大模型
   - TDM、Genie
4. 多智能体大模型
   - MADT、MAT



$$最优策略 = \arg\max_{策略} \mathbb{E}_{(状态,动作)\sim策略交互的数据分布}[奖励函数(状态,动作)]$$

强化学习大模型：从序列建模来看强化学习任务

- 从序列建模看强化学习的策略方法

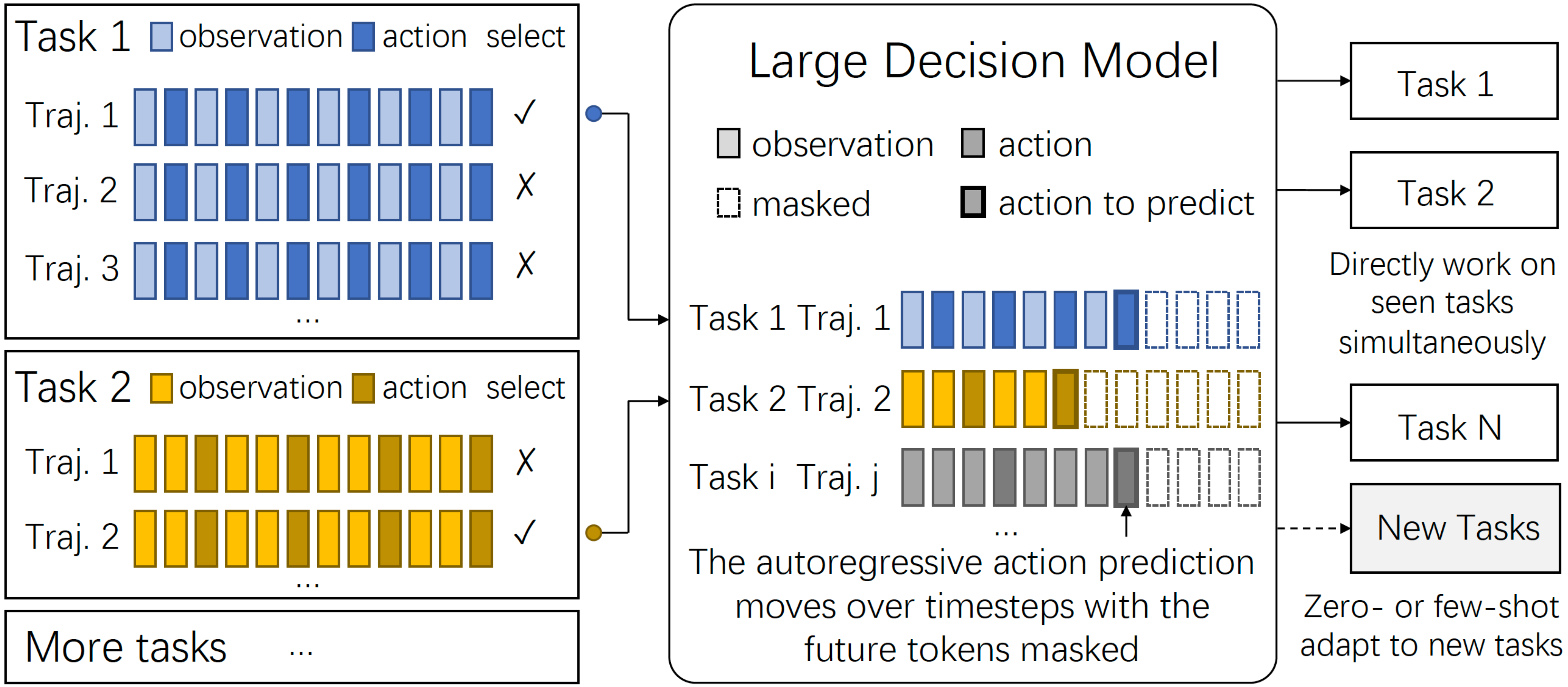
$$\theta \leftarrow \theta + \alpha \ \mathbb{E}_{(\mathbf{s},\mathbf{a})\sim\rho^{\mu}}\left[m(\mathbf{s},\mathbf{a})\nabla_{\theta}\log\pi_{\theta}(\mathbf{a}|\mathbf{s})\right]$$

- In sequence modeling, $m(s,a)=1$
- In traditional RL, $m(s,a)$ is the value function corresponds to $Q^{\pi}(s,a)$
  - Learned by TD propagation
  - Need the agent to interact with environment
- In large decision-making models of offline RL, $m(s,a)$ is a masking (or weighting) function
  - Just predefined by rule
  - No need to interact with environment (but require huge amount of data)

- How the masking function (or selection function) can be defined?
  - By learning a value function, e.g., critic regularized regression
  - By rules
  - By data, directly
  - Multi-agent credit assignment (attribution)
  - From good trajectories (of GATO) to specific good (s,a) pairs

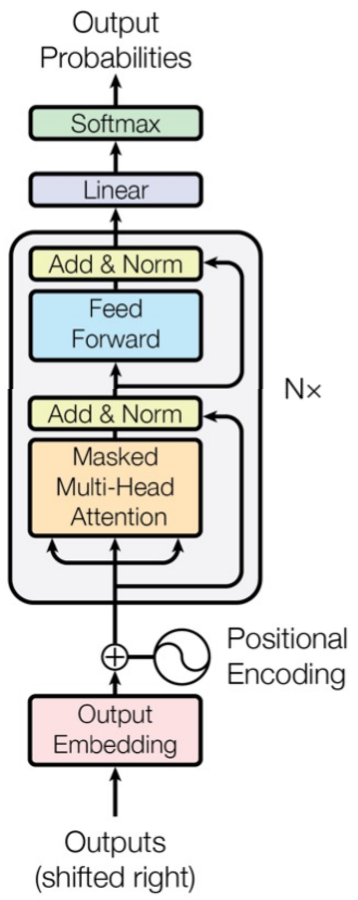
强化学习大模型：从序列建模来看强化学习任务

- 从序列建模看强化学习的策略方法

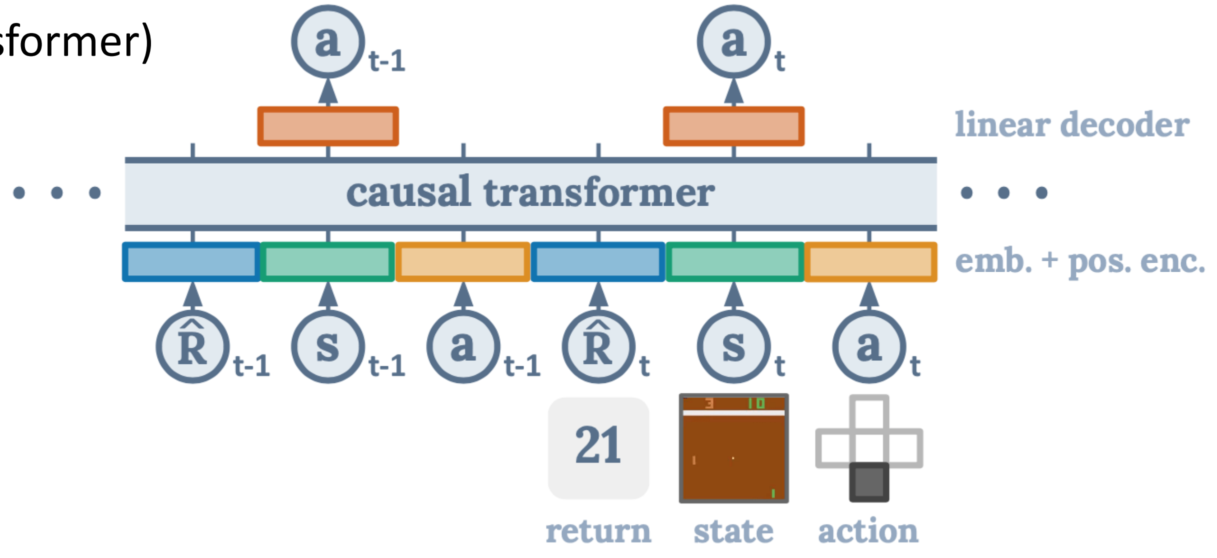


强化学习大模型：从序列建模来看强化学习任务

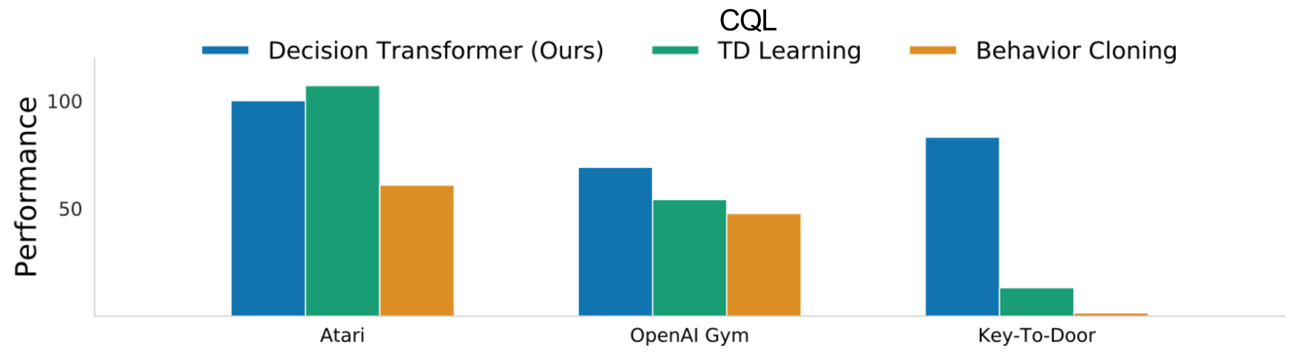
- Re-build RL task as a sequence prediction problem
- Causal transformer: GPT (i.e., decoder-only transformer)



Decoder-only Transformer的架构



Decision Transformer的基本架构

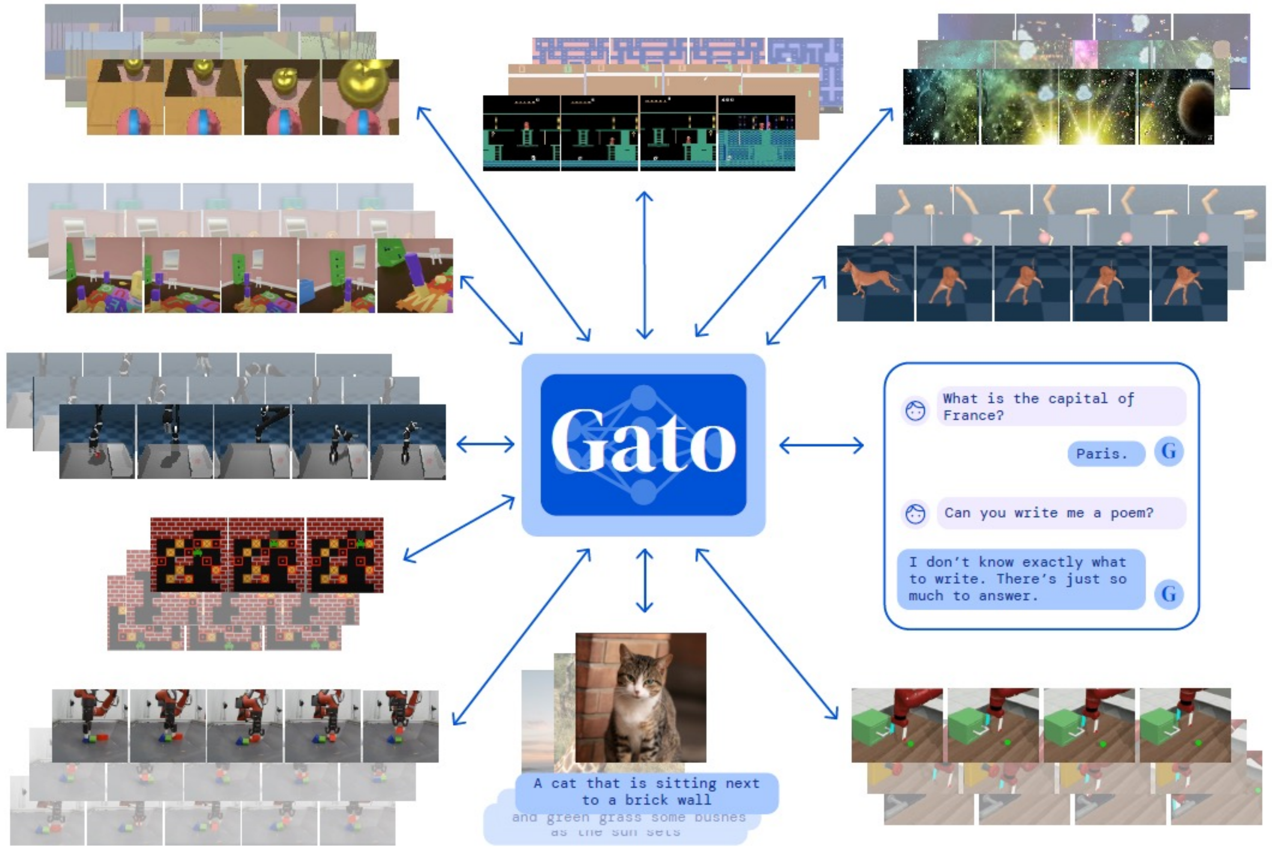


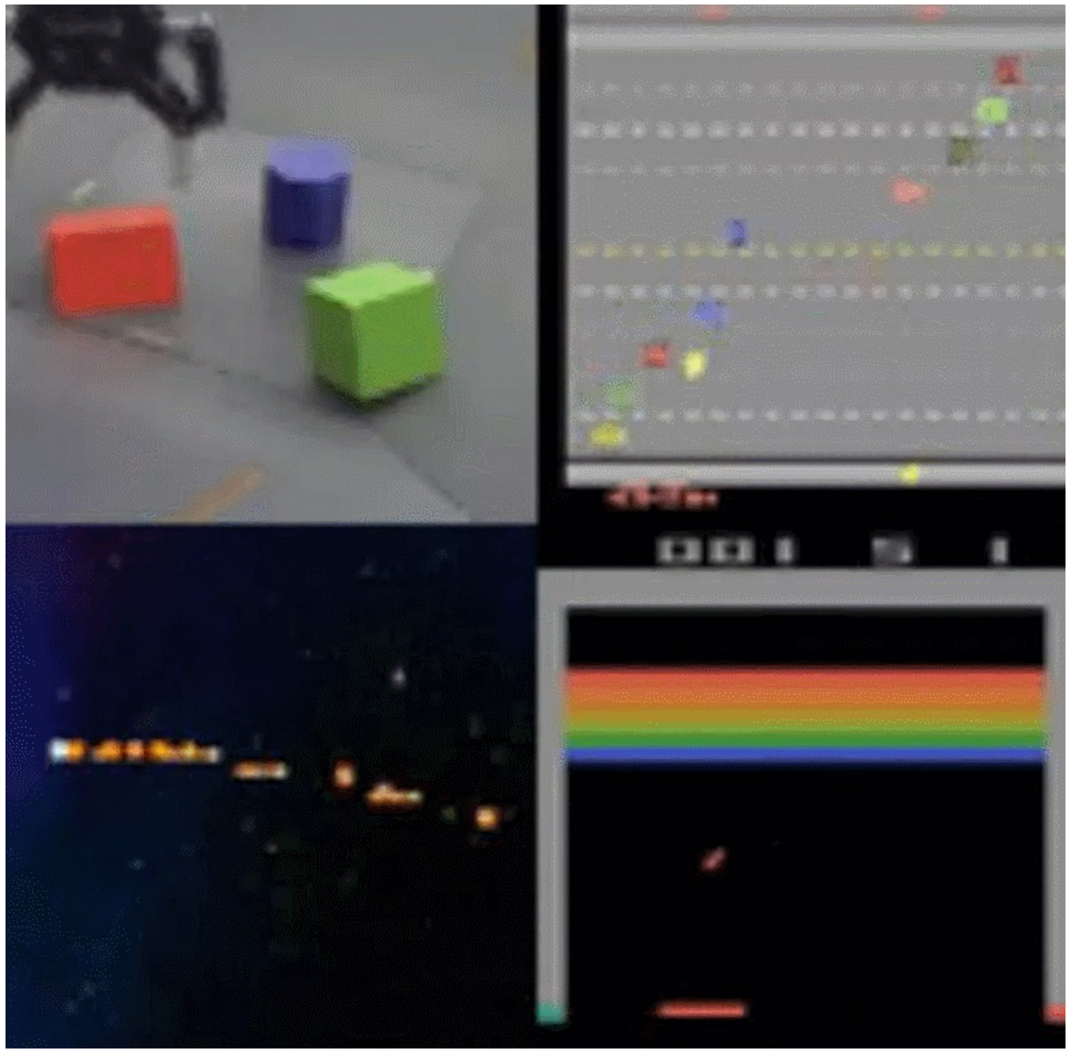
Decision Transformer在一些离线强化学习的任务中已经取得很不错的效果

Chen et al. Decision Transformer: Reinforcement Learning via Sequence Modeling. NeurIPS 2021.

策略大模型：Gato

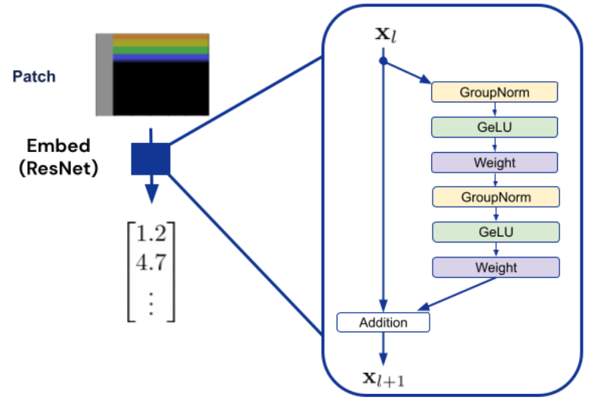
- DeepMind于2022年5月发布Gato决策大模型，使用同一个GPT网络（11.8亿参数）同时完成约600个任务
  - 采样每个任务的专家数据
  - 使用完全有监督学习离线训练
  - 使用部分专家轨迹做prompt


Gato
What is the capital of France?
Paris.
Can you write me a poem?
I don't know exactly what to write. There's just so much to answer.
A cat that is sitting next to a brick wall
and green grass some bushes as the sun sets



Reed et al. A Generalist Agent. Arxiv 2022.

策略大模型：Gato – 多模态数据离散化



- ResNet embedding for image patches

- Text and discrete data are directly embedded
- Image patches are embedded via ResNet
- Tensor data are mu-law encoded and discretized, then embedded

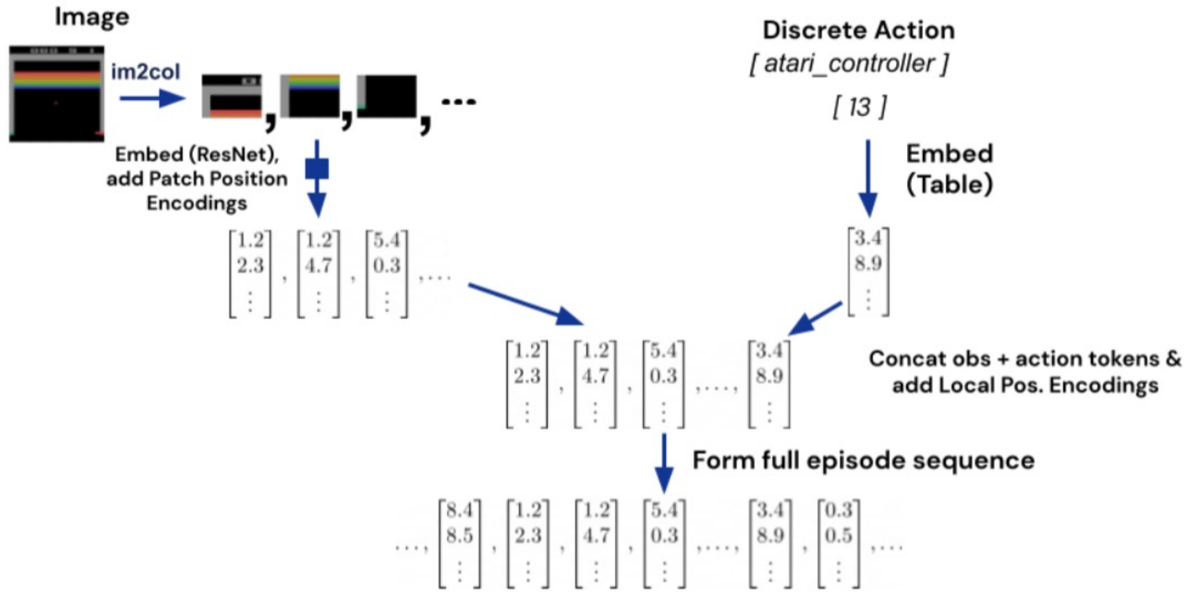
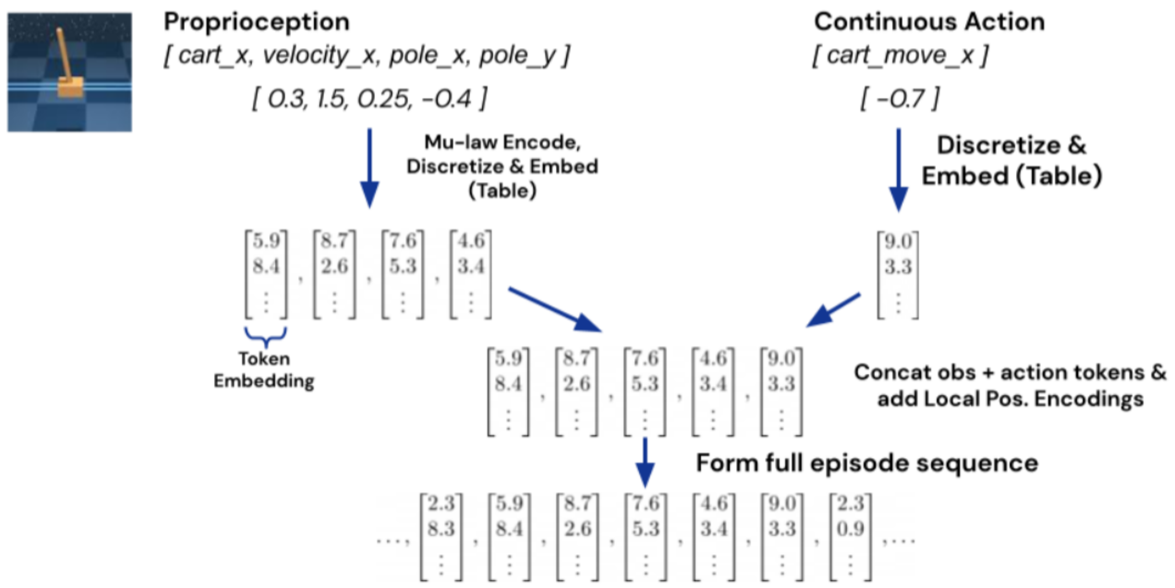
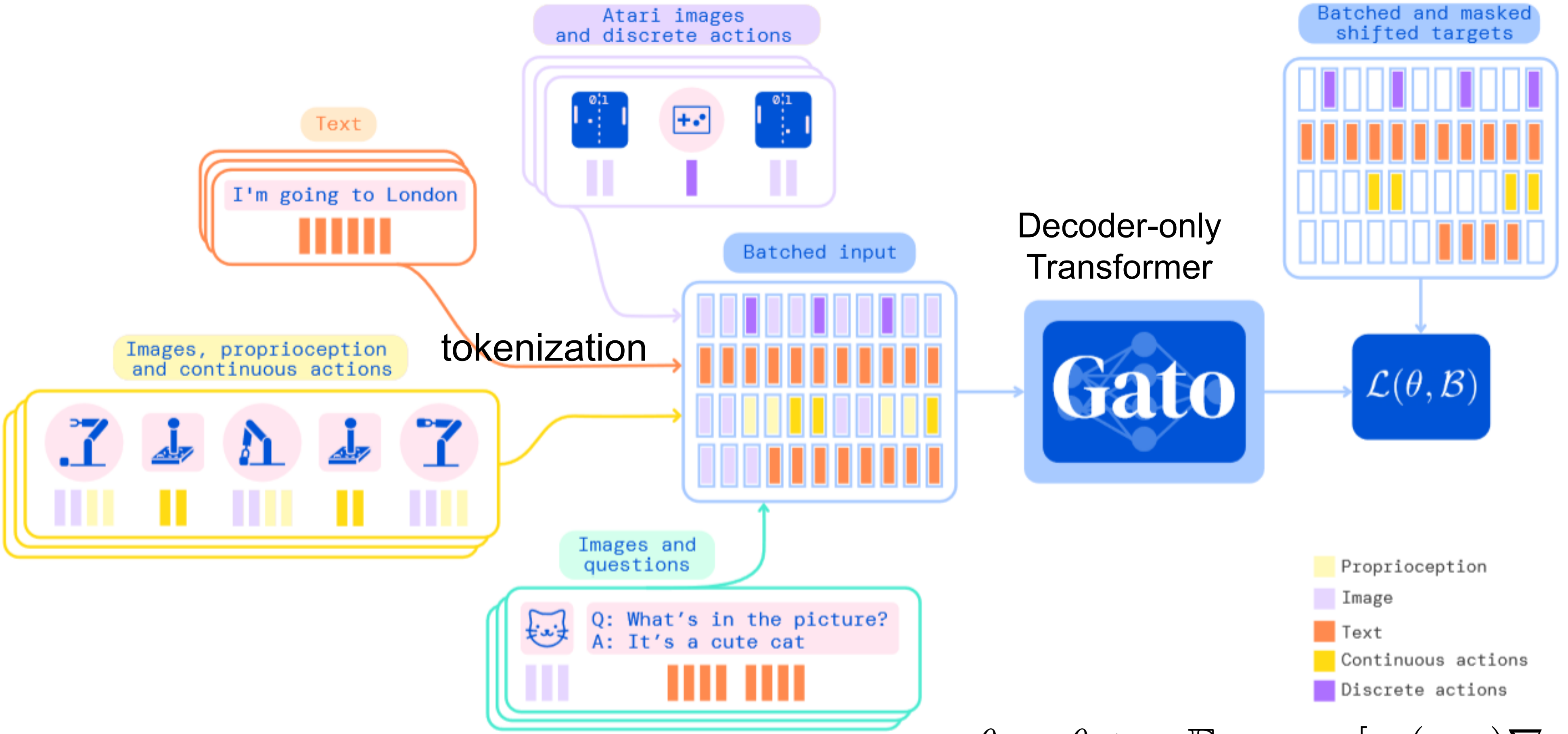


Image
im2col
Embed (ResNet), add Patch Position Encodings
Discrete Action
[atari_controller]
[13]
Embed (Table)
Concat obs + action tokens & add Local Pos. Encodings
Form full episode sequence


Proprioception
[cart_x, velocity_x, pole_x, pole_y]
[0.3, 1.5, 0.25, -0.4]
Mu-law Encode, Discretize & Embed (Table)
Token Embedding
Continuous Action
[cart_move_x]
[-0.7]
Discretize & Embed (Table)
Concat obs + action tokens & add Local Pos. Encodings
Form full episode sequence

策略大模型：Gato – 数据离散化后的统一训练框架

- 首先得到每个任务的专家轨迹数据（可以通过强化学习得到该任务的最优策略作为专家）
- GATO将所有决策轨迹数据做离散化，再统一喂入GPT架构模型中训练，预测动作

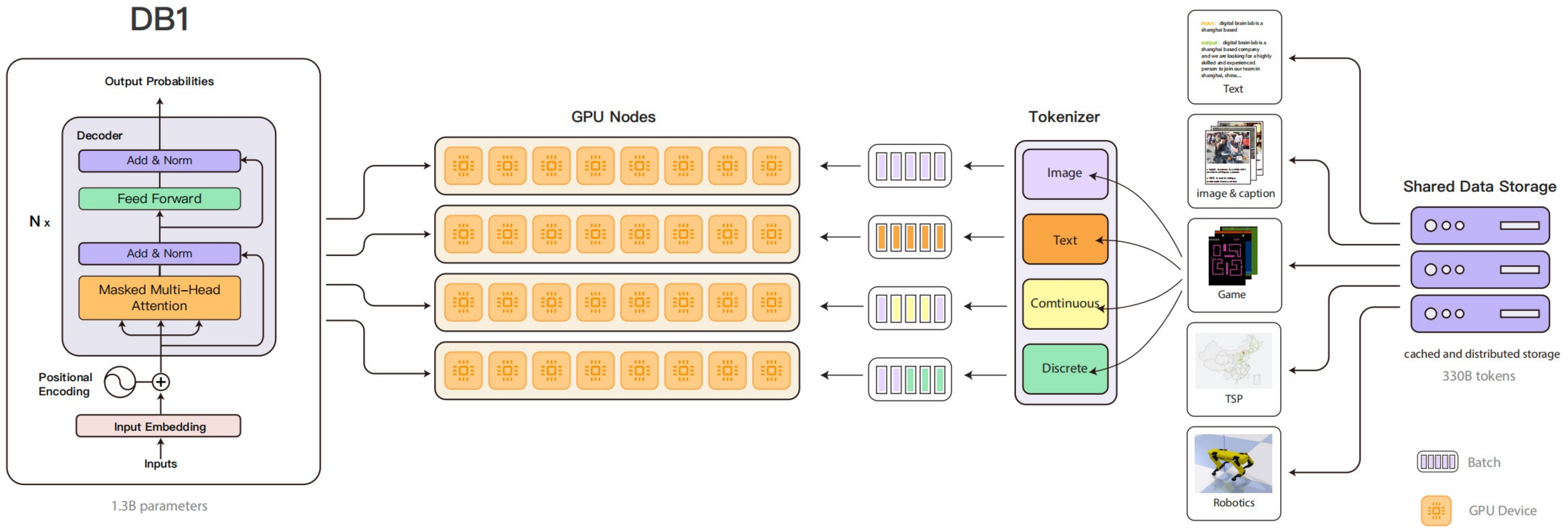

Atari images and discrete actions
Text
I'm going to London
Batched input
Decoder-only Transformer
Batched and masked shifted targets
tokenization
Images, proprioception and continuous actions
Gato
$\mathcal{L}(\theta, \mathcal{B})$
Images and questions
Q: What's in the picture?
A: It's a cute cat
Proprioception
Image
Text
Continuous actions
Discrete actions

Reed et al. A Generalist Agent. Arxiv 2022.

$$\theta \leftarrow \theta + \alpha\, \mathbb{E}_{(\mathbf{s},\mathbf{a})\sim\rho^{\mu}}[m(\mathbf{s},\mathbf{a})\nabla_\theta \log \pi_\theta(\mathbf{a}|\mathbf{s})]$$

策略大模型：DB1

- 2022年12月，上海交通大学、数字大脑研究院、上科大、UCL联合发布多模态决策大模型DB1
  - 智能体设置：单智能体和多智能体
  - 训练模型：离线训练或在线训练
  - 参数量：13.2亿
  - 实验任务数量：870个
  - 任务类型：游戏、控制、图文、对话、运筹优化
  - 模型性能：76.7%任务超过专家性能50%

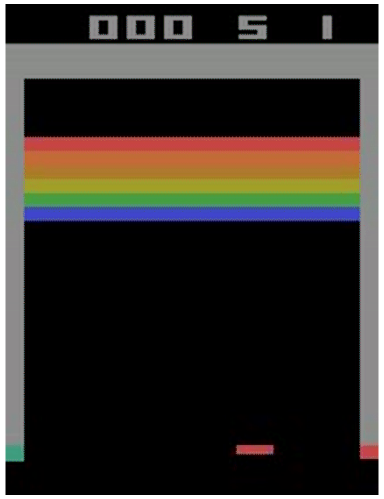

DB1
Output Probabilities
Decoder
Add & Norm
Feed Forward
Add & Norm
Masked Multi-Head Attention
N x
Positional Encoding
Input Embedding
Inputs
1.3B parameters
GPU Nodes
Tokenizer
Image
Text
Comtinuous
Discrete
input: digital brain lab is a shanghai based
output: digital brain lab is a shanghai based company and we are looking for a highly skilled and experienced person to join our team in shanghai, china...
Text
image & caption
Game
TSP
Robotics
Shared Data Storage
cached and distributed storage
330B tokens
Batch
GPU Device

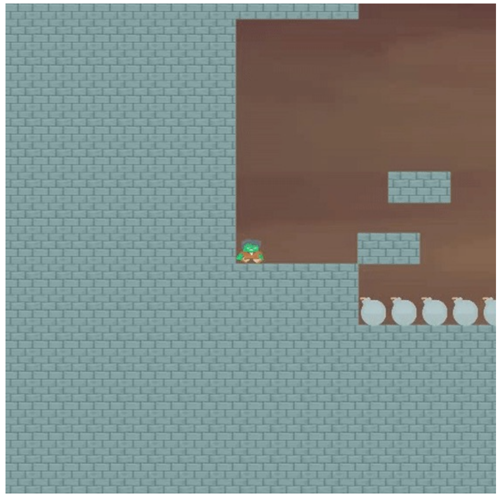
Ying Wen et al. On Realization of Intelligent Decision-Making in the Real World: A Foundation Decision Model Perspective. Arxiv 2022. https://arxiv.org/abs/2212.12669

策略大模型：DB1

DB1 surpasses half of expert performance on **76.7%** of 870 tasks

Task Suite	Tasks	Trajectories	Tokens	Sample weights
DeepMind Lab	411	3M	171B	13%
ALE Atari	52	32K	1.4B	13%
Sokoban	1	910	2.8M	1.7%
BabyAI	55	5.5M	33.5B	13%
DM Control Suite	25	675K	20.2B	5.2%
Meta-World	42	88.2K	1.9B	11.7%
Procgen Benchmark	16	1.6M	7.2B	6.5%
Modular RL	38	843.6K	13.9B	10.4%
TSP	230	2.3M	108.5B	11.7%





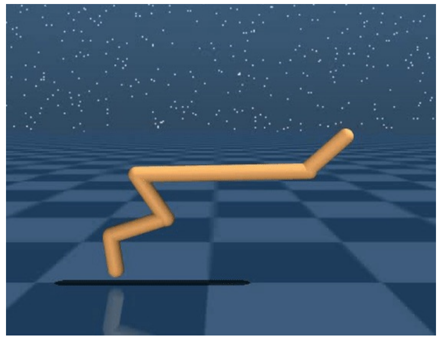
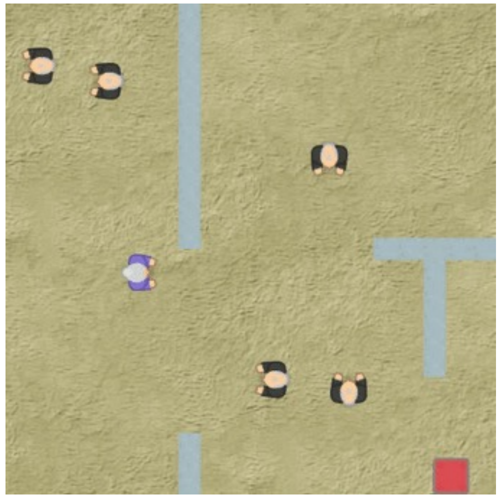


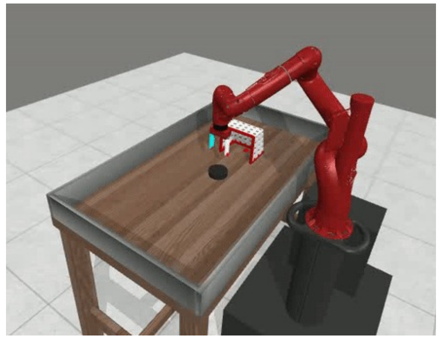
Image Captioning

Label: A man on a motorcycle dressed in black leather.
DB1: A man is riding a motorcycle down a street.

Label: There are two zebras that are seen standing together.
DB1: A zebra standing in a field of grass.

Label: A woman looking at a new laptop.
DB1: A woman in a white shirt and tie is sitting on a couch.





Text Generation

Prompt: Digital Brain Lab is a Shanghai based
DB1: Digital Brain Lab is a Shanghai based company and we are looking for a highly skilled and experienced person to join our team in shanghai, China.

Prompt: A wiki is a
DB1: A wiki is a collection of all the info about a given character and their background in order for you and other readers (easy to find). This wiki also provides links, which you might use in your quest.

Ying Wen et al. On Realization of Intelligent Decision-Making in the Real World: A Foundation Decision Model Perspective. Arxiv 2022. https://arxiv.org/abs/2212.12669

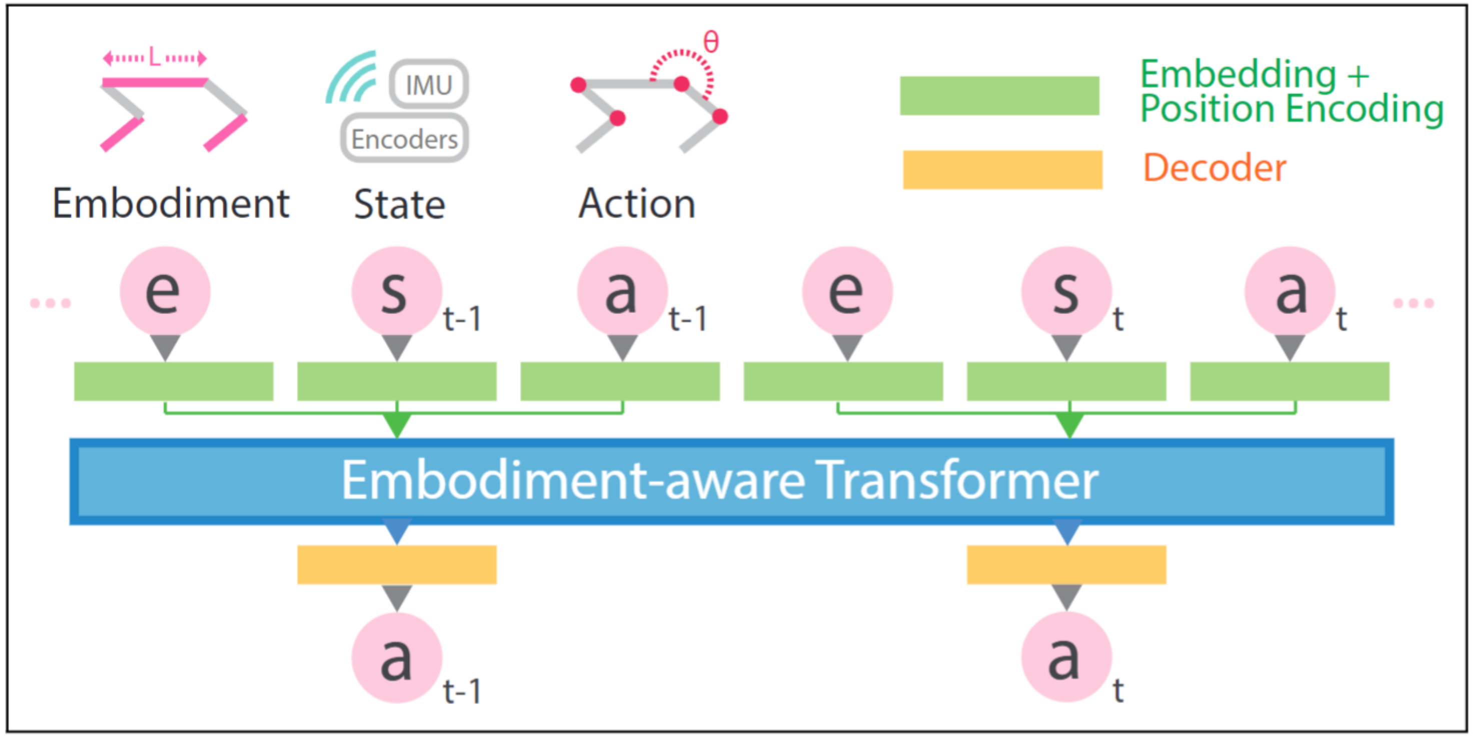
策略大模型：TERT (Terrain Transformer)

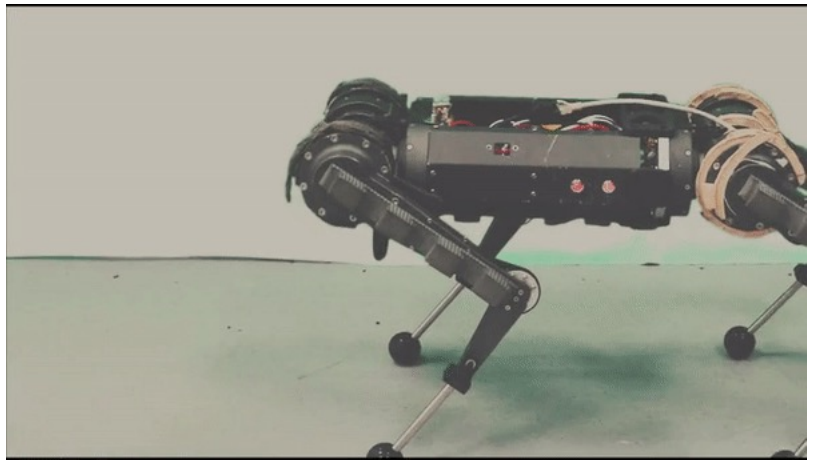
- 使用不同模拟地形行走数据，在同一个Transformer控制器上做特权蒸馏学习，完成sim2real迁移。

Yu, Chen, Weinan Zhang et al. Multi-embodiment Legged Robot Control as a Sequence Modeling Problem. ICRA 2023.

策略大模型：EAT/ADAPT

- 将机器人具身形态编码到运动控制策略中，当策略以大模型实现时，机器人可在具身形态上做有效泛化，甚至在关节损坏时仍能保持运动能力



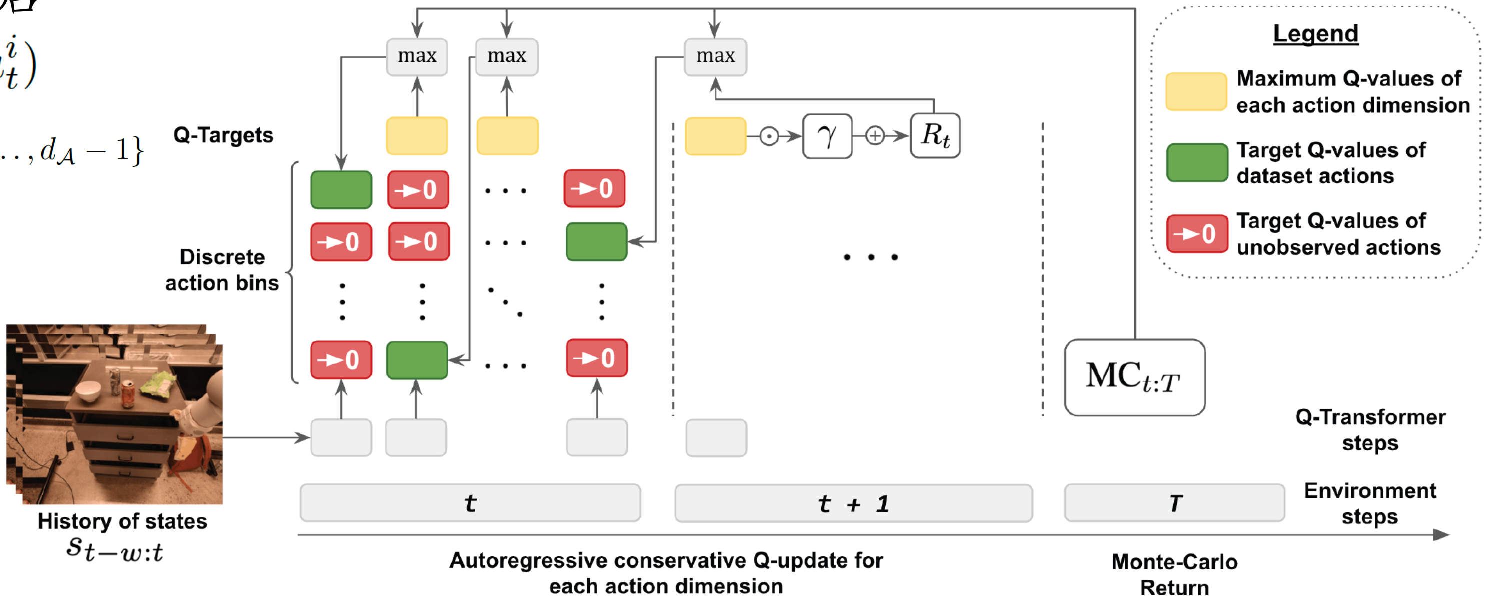


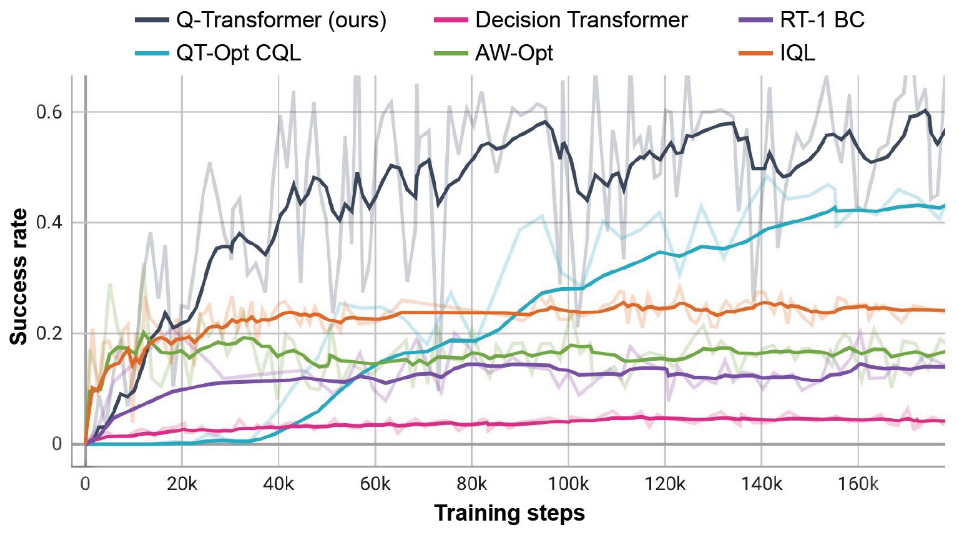
Yu, Chen, Weinan Zhang et al. Multi-embodiment Legged Robot Control as a Sequence Modeling Problem. ICRA 2023.
Wu, Xinyuan et al. Adaptive Control Strategy for Quadruped Robots in Actuator Degradation Scenarios. DAI 2023.

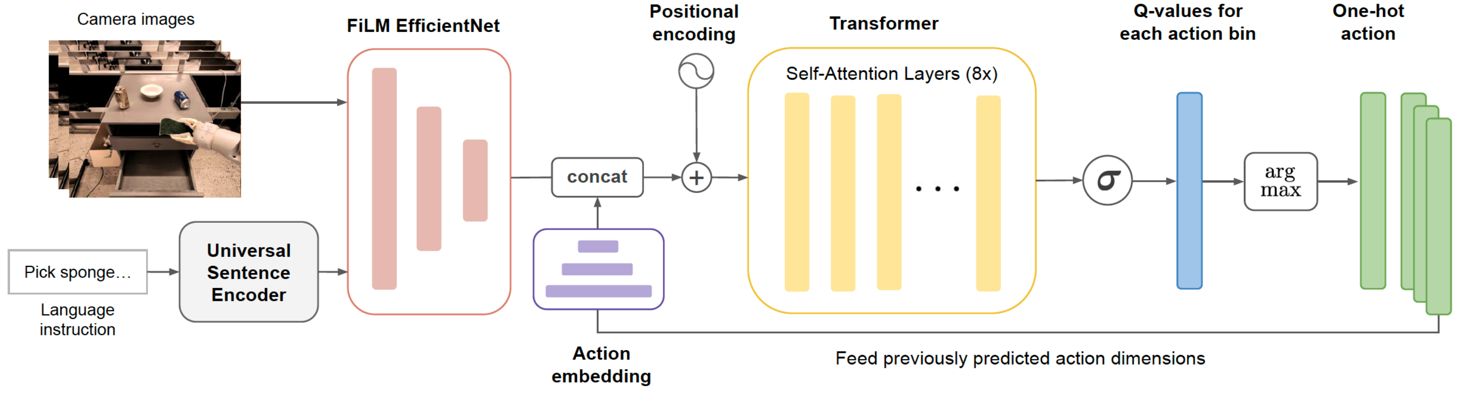
价值大模型：Q-Transformer

- Q-Transformer将动作逐位离散化，然后使用Transformer建模 $Q(s_{t-w:t}, a_t^{1:i-1}, a_t^i)$

$$Q(s_{t-w:t}, a_t^{1:i-1}, a_t^i) \leftarrow \begin{cases} \max\limits_{a_t^{i+1}} Q(s_{t-w:t}, a_t^{1:i}, a_t^{i+1}) & \text{if } i \in \{1, \dots, d_A - 1\} \\ R(s_t, a_t) + \gamma \max\limits_{a_{t+1}^1} Q(s_{t-w+1:t+1}, a_{t+1}^1) & \text{if } i = d_A \end{cases}$$


Legend
Maximum Q-values of each action dimension
Target Q-values of dataset actions
Target Q-values of unobserved actions
Q-Targets
Discrete action bins
max
γ
R_t
→0
$\mathrm{MC}_{t:T}$
History of states
$s_{t-w:t}$
Q-Transformer steps
Environment steps
t
$t+1$
T
Autoregressive conservative Q-update for each action dimension
Monte-Carlo Return





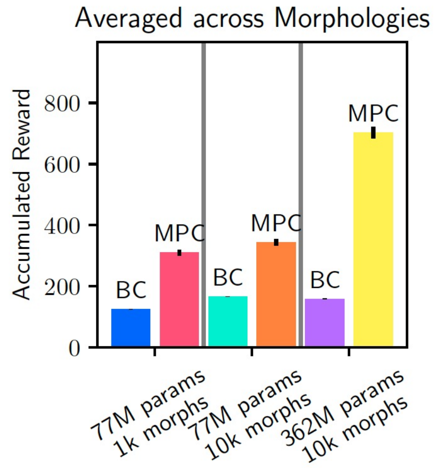
35M parameters

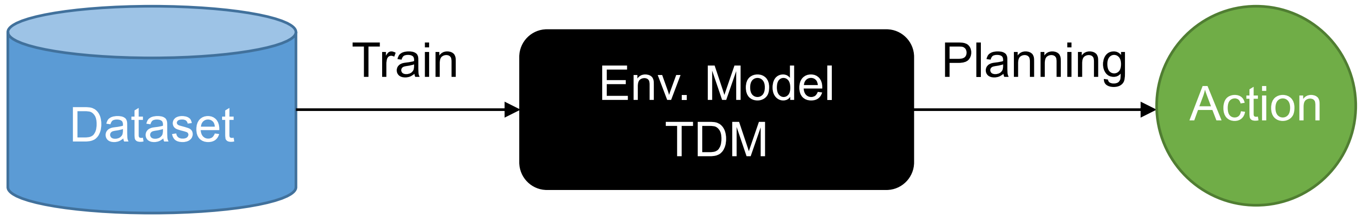
Yevgen Chebotar et al. Q-Transformer: Scalable Offline Reinforcement Learning via Autoregressive Q-Functions. CoRL 2023.

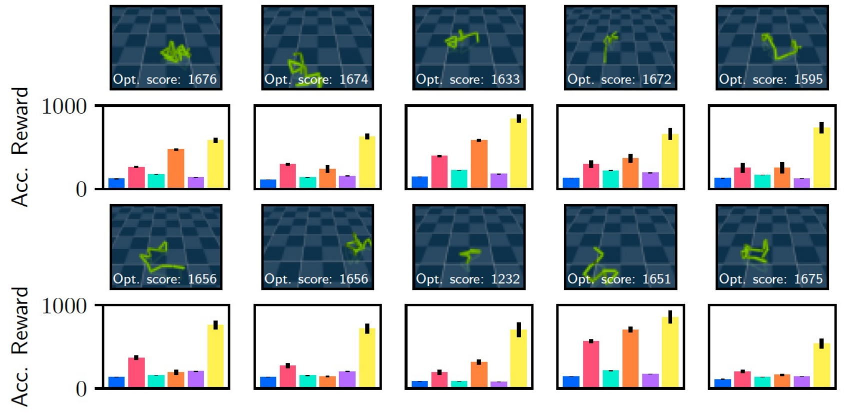
环境大模型：TDM

Up to 362M parameters

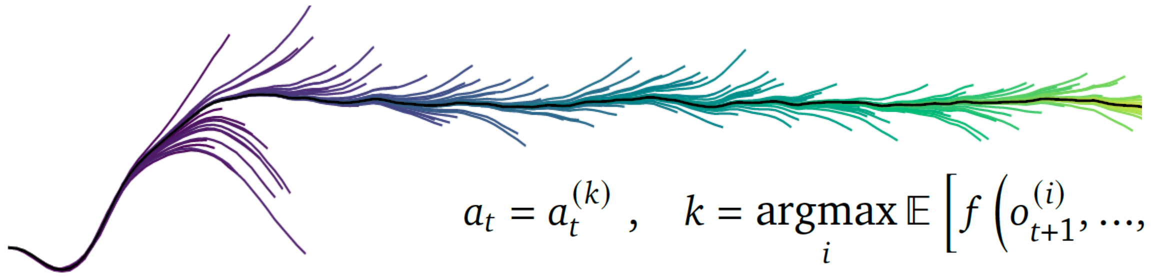
- 2023年5月，DeepMind发布环境大模型研究工作：Transformer Dynamics Model (TDM)，使用基于Transformer的序列模型构建环境动态模型。
- 基本结论：相比于Gato拟合专家策略，TDM拟合环境下一步观测可能更加容易
  - Condition on (h_t, o_t), obtain r_t: Reward model
  - Condition on (h_t, o_t, r_t), obtain a_t: BC policy
  - Condition on (h_t, o_t, r_t, a_t), obtain o_{t+1}: Dynamics model (TDM)







- 基于此做random action shooting的Model Prediction Control (MPC) Planning，就能取得很好的跨环境泛化效果



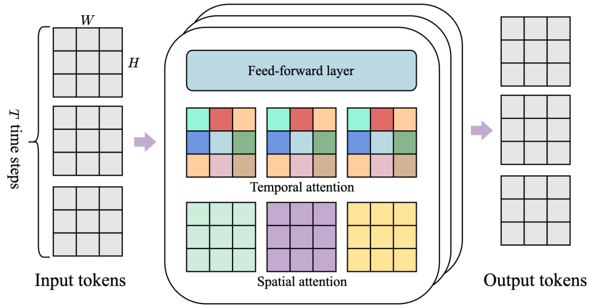
$$a_t = a_t^{(k)}, \quad k = \operatorname*{argmax}_i \mathbb{E}\left[f\left(o_{t+1}^{(i)}, \dots, o_{t+N}^{(i)}, a_t^{(i)}, \dots, a_{t+N-1}^{(i)}\right) \middle| a_t^{(i)}, \dots, a_{t+N-1}^{(i)}, o_t, h_t\right]$$

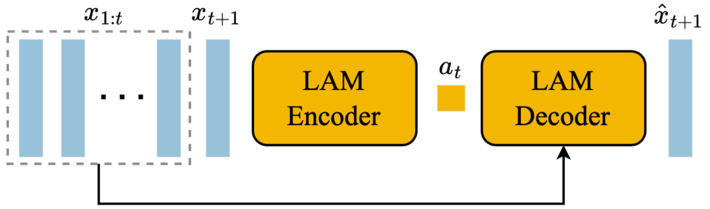
Schubert et al. A Generalist Dynamics Model for Control. Arxiv 2023.

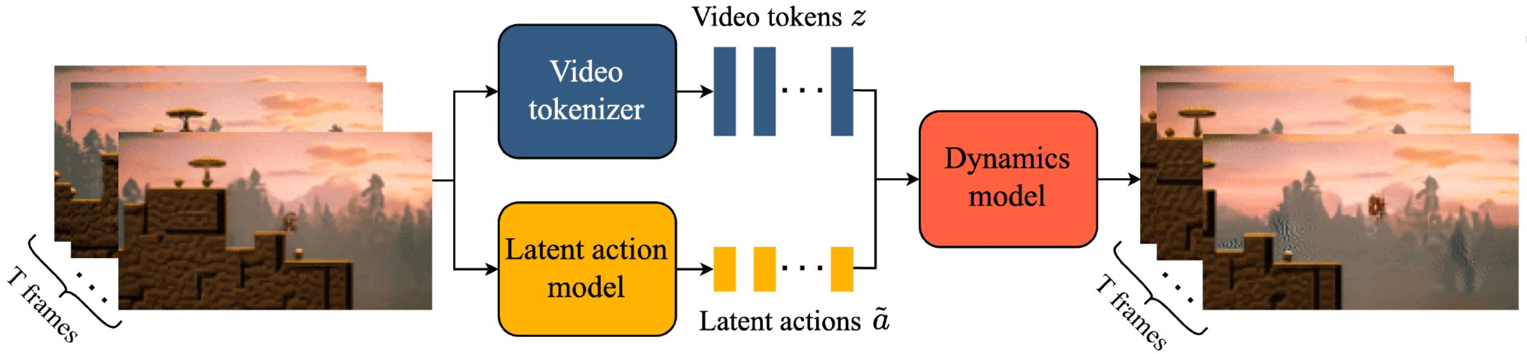
环境大模型：Genie

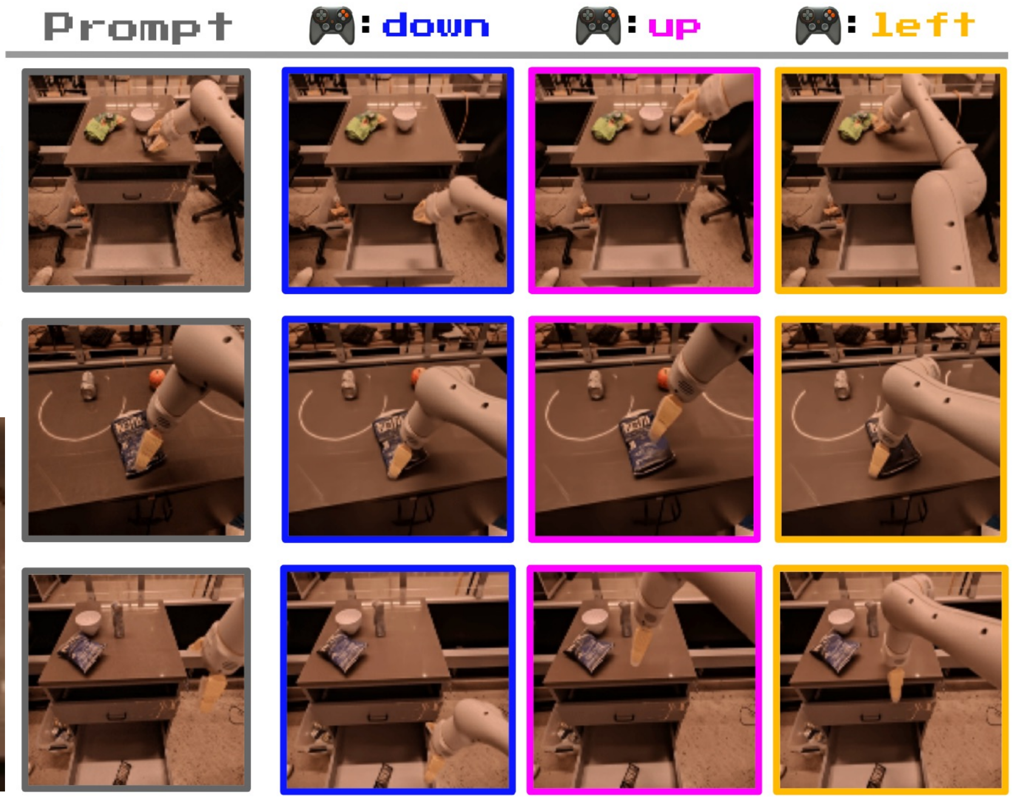
Up to 11B parameters

- 2024年2月，DeepMind发布环境大模型研究工作：Genie。使用基于ST-Transformer的序列模型和Latent Action Model构建环境动态模型，以视频生成的方式做环境模拟。





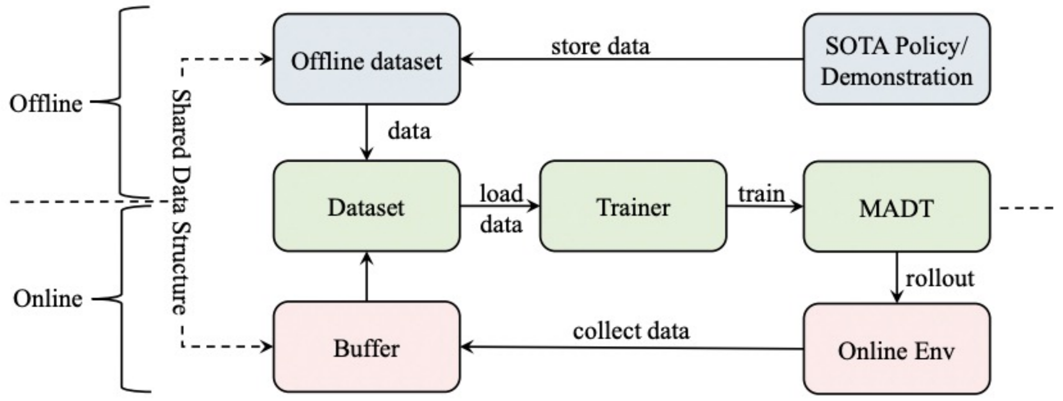




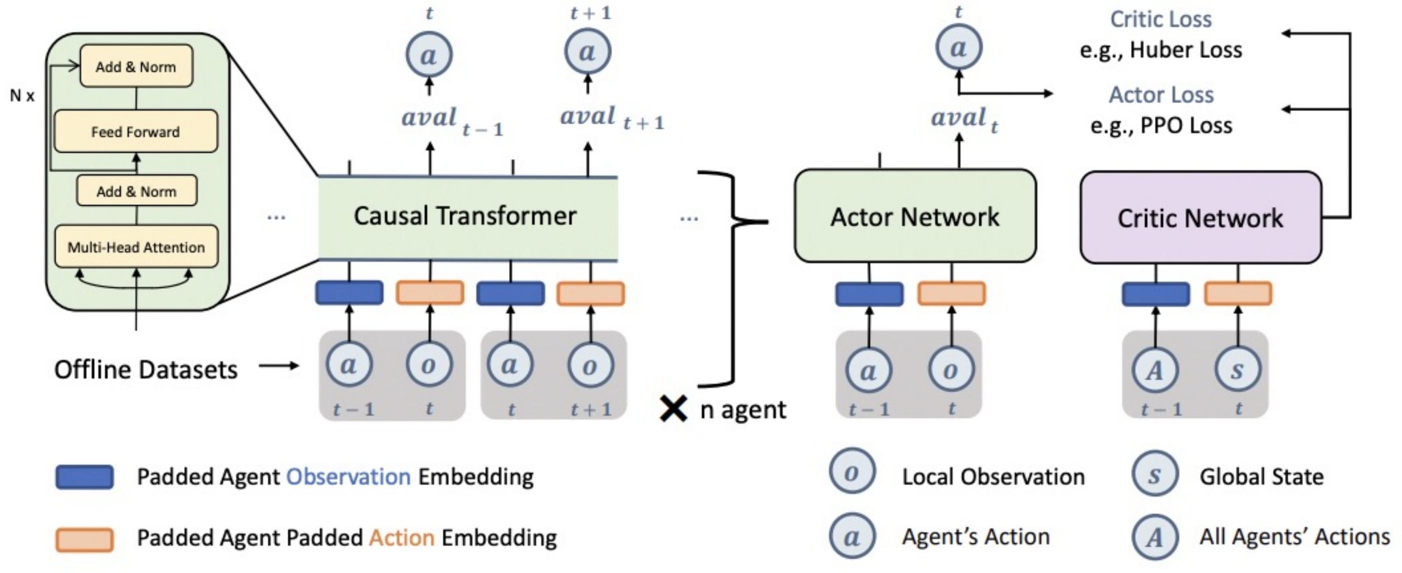
Jake Bruce et al. Genie: Generative Interactive Environments. Arxiv 2024.

多智能体大模型：MADT

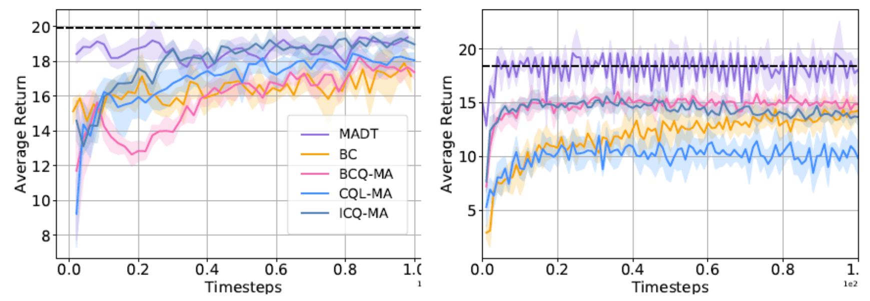
- 2021年12月，中科院、上海交通大学、数字大脑研究院、北京大学、UCL等高校团队发布首个多智能体强化大模型MADT，使用一个GPT模型完成多个星际争霸对战任务。



MADT训练流程

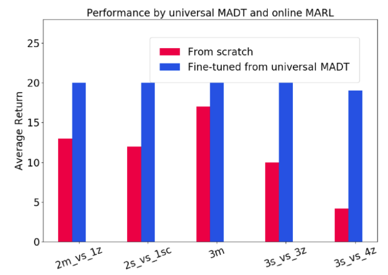


MADT架构



(a) 2s3z (easy) (b) 3s5z (hard)

离线训练效果



预训练微调效果

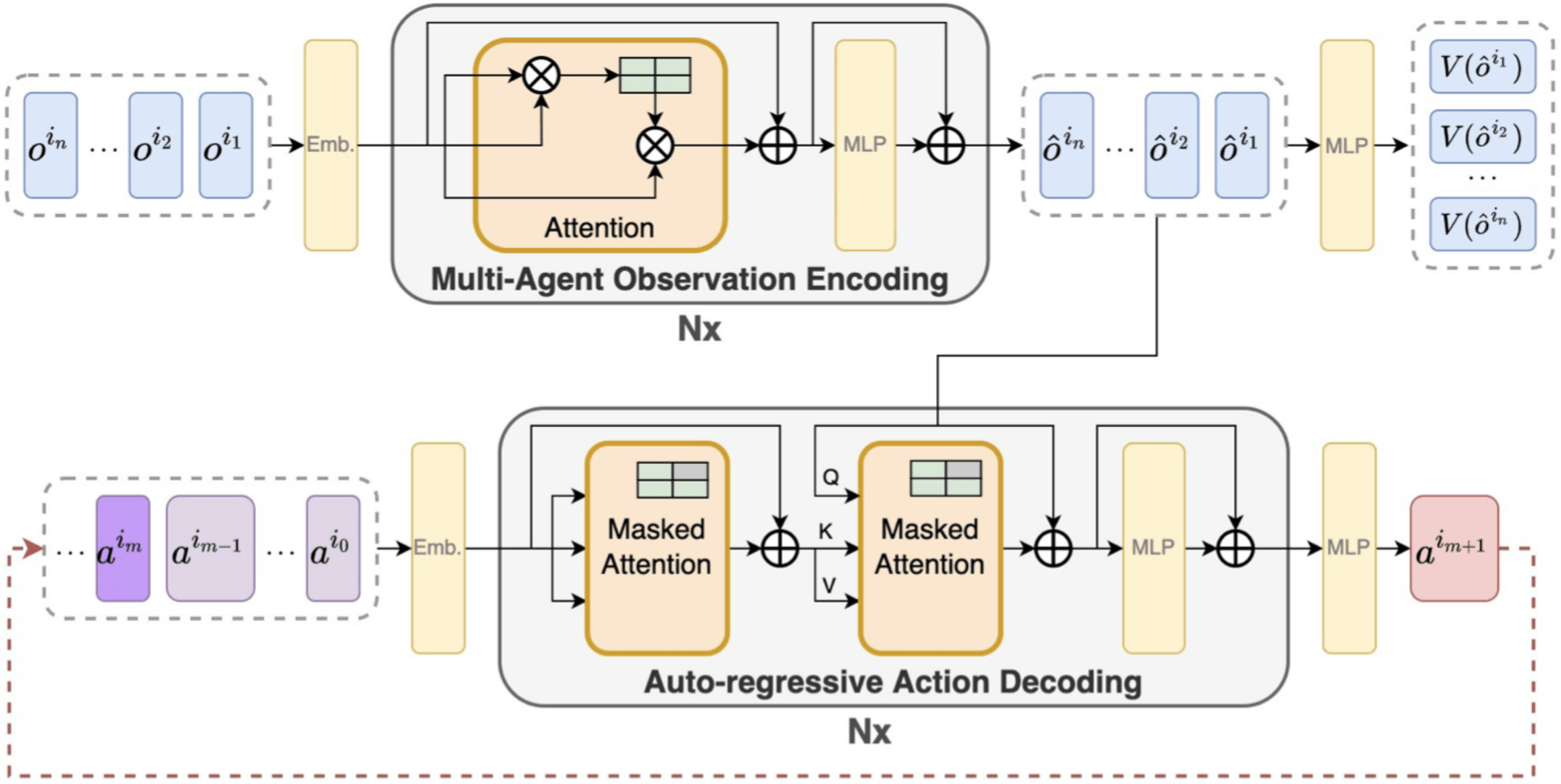
Meng, Linghui, et al. "Offline Pre-trained Multi-Agent Decision Transformer: One Big Sequence Model Conquers All StarCraftII Tasks." arXiv:2112.02845 (2021).

多智能体大模型：MAT

- 基于HAPPO中的优势函数分解定理

$$A_{\boldsymbol{\pi}}^{i_{1:m}}\left(s, \boldsymbol{a}^{i_{1:m}}\right)=\sum_{j=1}^{m} A_{\boldsymbol{\pi}}^{i_j}\left(s, \boldsymbol{a}^{i_{1:j-1}}, a^{i_j}\right)$$

- Multi-Agent Transformer (MAT)
  - 将多智能体合作决策问题抽象为序列建模问题。
  - 作为桥梁连接了多智能体强化学习（MARL）与当前最先进的序列模型Transformer，从而使序列模型方面的许多先进研究成果也能被应用于MARL领域。
  - 是一个在线学习算法，无需预收集数据集。
  - 拥有完整的Transformer（encoder-decoder）结构，同时也提供了纯encoder与decoder的实现，以便各类扩展算法使用。



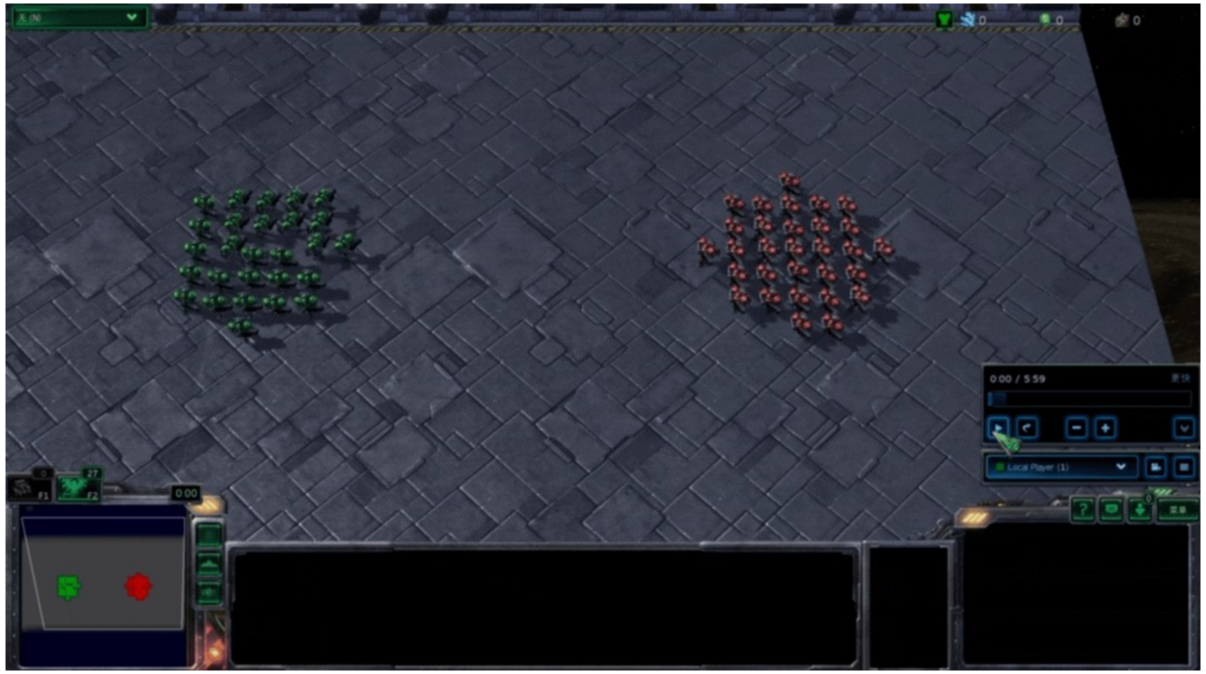
o^{i_n} … o^{i_2} o^{i_1}
Emb.
Attention
Multi-Agent Observation Encoding
Nx
MLP
$\hat{o}^{i_n}$ … $\hat{o}^{i_2}$ $\hat{o}^{i_1}$
MLP
$V(\hat{o}^{i_1})$
$V(\hat{o}^{i_2})$
…
$V(\hat{o}^{i_n})$
… a^{i_m} $a^{i_{m-1}}$ … a^{i_0}
Emb.
Masked Attention
Q
K
V
Masked Attention
MLP
MLP
$a^{i_{m+1}}$
Auto-regressive Action Decoding
Nx

Kuba, J. G., et al. Trust Region Policy Optimisation in Multi-Agent Reinforcement Learning. ICLR 2022.
Wen, M., et al. Multi-Agent Reinforcement Learning is A Sequence Modeling Problem. NeurIPS 2022.

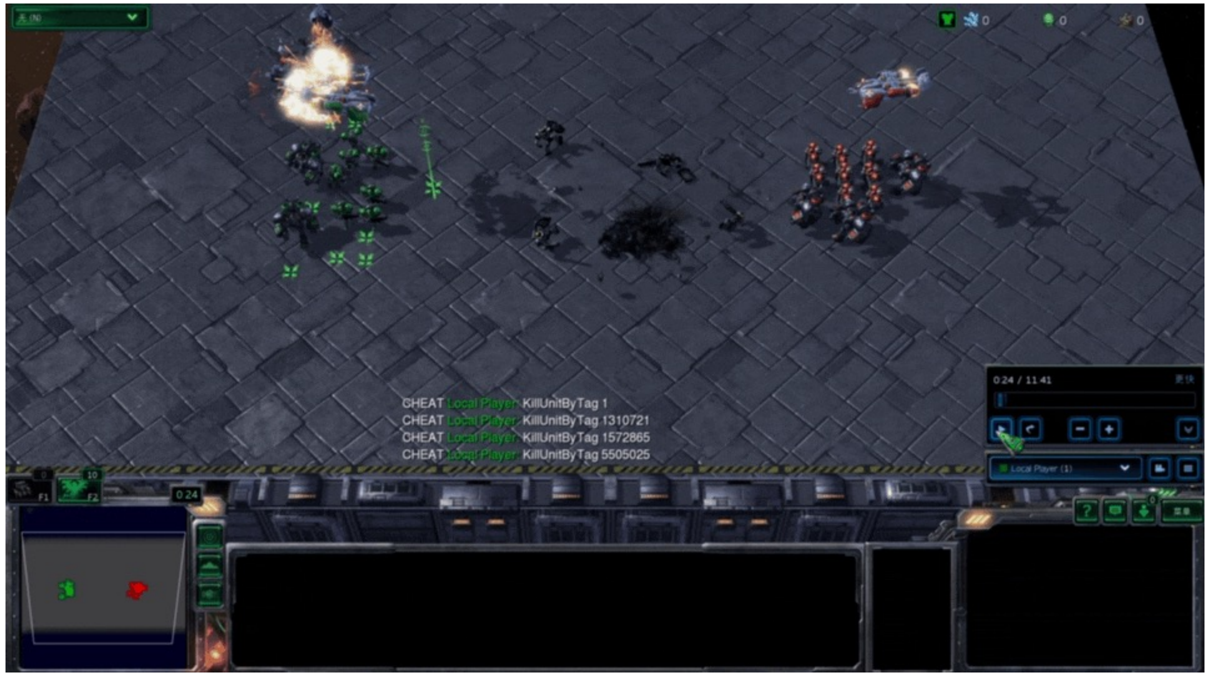
多智能体大模型：MAT在星际争霸战斗任务性能表现

☐ 单任务性能可视化

27小兵 vs 30小兵



1医疗机2机甲7小兵 vs 1医疗机3机甲8小兵



```
CHEAT Local Player: KillUnitByTag 1
CHEAT Local Player: KillUnitByTag 1310721
CHEAT Local Player: KillUnitByTag 1572865
CHEAT Local Player: KillUnitByTag 5505025
```

☐ 多任务Few-Shot性能表现（少样本快速适应新任务）

星际争霸是在8个任务上训练后并在6个新任务上进行测试，相较其他算法，在新任务样本较少时MAT能取得最好的表现

Table 2: Median evaluation win rate and the standard deviation on the SMAC benchmark for pre-trained models with different number of online examples.

Methods	MAT				MAPPO				MAT-from scratch			
#examples	0%	1%	5%	10%	0%	1%	5%	10%	0%	1%	5%	10%
5m vs 6m	0.0(0.0)	0.0(0.0)	**5.8**(3.1)	18.8(7.1)	0.0(0.0)	0.0(0.0)	4.3(3.8)	**21.9**(12.2)	0.0(0.0)	0.0(0.0)	1.9(1.3)	3.8(2.1)
8m	100(0.0)	100(1.2)	100(0.3)	100(2.1)	100(0.0)	100(1.4)	100(0.3)	100(1.4)	0.0(0.0)	10.6(23.8)	92.5(3.7)	100(1.4)
27m vs 30m	0.0(0.0)	6.3(2.4)	**53.8**(16.4)	**71.2**(8.2)	**9.4**(3.6)	**15**(5.9)	26.2(7.8)	26.8(9.7)	0.0(0.0)	0.0(0.0)	0.0(0.3)	0.3(15.6)
2s vs 1sc	0.0(0.0)	15.6(13.8)	100(9.7)	100(0.0)	0.0(0.0)	**43.1**(17.6)	100(1.1)	100(1.8)	0.0(0.0)	19.3(33.3)	96.3(6.2)	100(0.3)
1c3s5z	**3.1**(1.8)	**5.6**(5.0)	**82.5**(5.5)	**100**(2.7)	**3.1**(1.8)	4.3(4.9)	73.8(13.0)	97.5(2.1)	0.0(0.0)	7.5(4.8)	87.5(3.9)	100(1.4)
MMM2	0.0(3.6)	0.0(1.8)	**33.8**(13.7)	**62.5**(12.1)	0.0(0.0)	0.0(1.4)	13.8(7.0)	36.2(9.6)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.7)

在MuJoCo机器人环境中模拟关节故障的情况下，实验显示MAT能够更快地适应意外情况的发生

Table 3: Average evaluation score and standard deviation on Multi-Agent MuJoCo for pre-trained models with different number of online examples.

Methods	MAT				MAPPO				MAT-from scratch			
#examples	0%	1%	5%	10%	0%	1%	5%	10%	0%	1%	5%	10%
back foot	2100(89)	2837(95)	**4691**(235)	**5646**(79)	**2936**(301)	**3017**(135)	3221(119)	3304(129)	-0.44(0.4)	-5.18(11)	670(1098)	1635(1184)
back shin	**4005**(316)	**4143**(230)	**6077**(209)	**7176**(74)	2406(32)	2542(108)	2796(137)	2955(127)	-0.31(0.1)	-3.95(17)	743(537)	1252(1123)
back thigh	**5361**(45)	**5641**(150)	**7101**(119)	**7460**(61)	3043(79)	3060(143)	3217(33)	3353(71)	-0.54(0.3)	-4.87(7.7)	930(589)	2067(861)
fore foot	**1313**(512)	**1955**(232)	**4856**(146)	**6054**(172)	623(44)	970(185)	2025(371)	2480(239)	-0.37(0.2)	-2.25(7.9)	1821(157)	2877(106)
fore shin	**2435**(13)	**2617**(71)	**3851**(57)	**4373**(83)	1715(55)	2457(125)	3096(59)	3310(54)	-0.15(0.06)	-0.96(6.0)	1461(101)	3003(316)
fore thigh	**5631**(321)	**6448**(417)	**7952**(109)	**8347**(81)	3087(110)	3171(83)	3340(52)	3519(59)	-0.29(0.3)	0.82(14)	1021(177)	2600(215)

Wen, M., et al. Multi-Agent Reinforcement Learning is A Sequence Modeling Problem. NeurIPS 2022.

多智能体大模型：MAT在谷歌足球上的表现

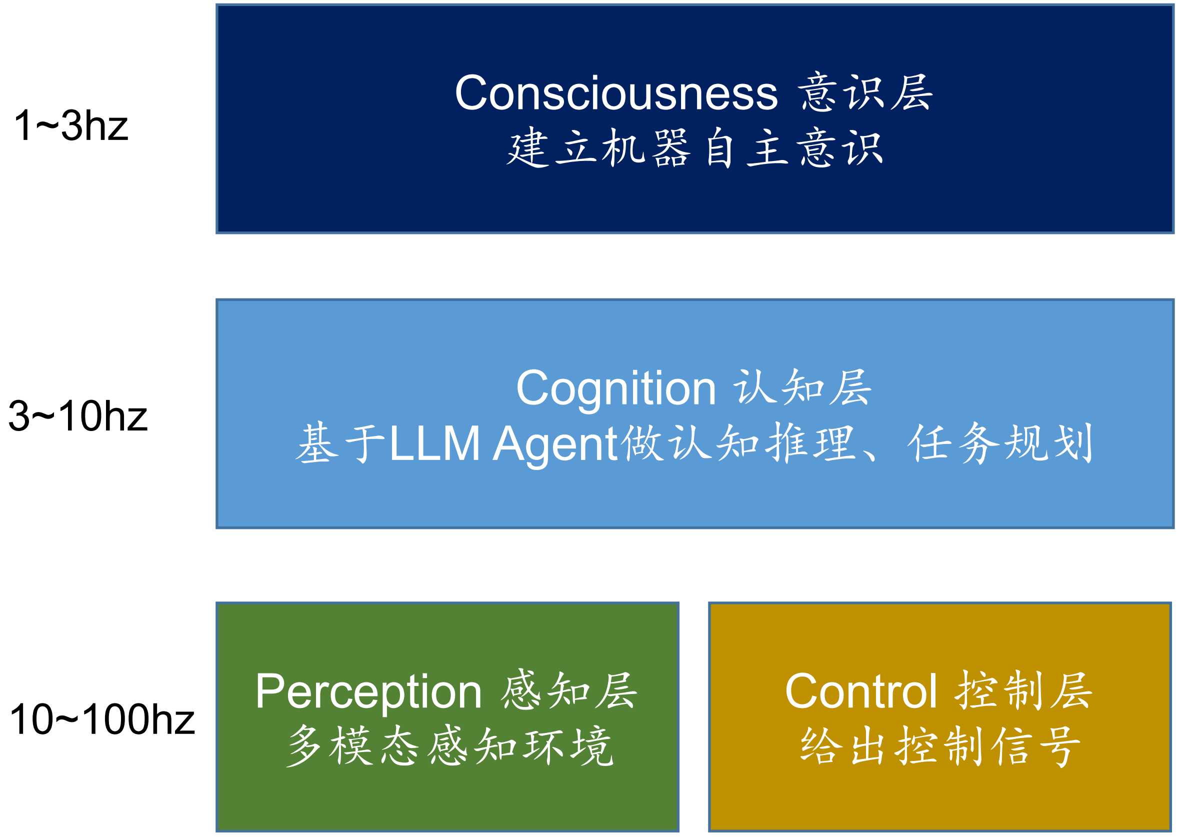
Algorithm	Off-Policy / On-Policy	Value-Based / Policy-Based	Agent Update Scheme	Training Mode	Execution Mode
QMIX [28]	Off-Policy	Value-Based	Simultaneous	Centralized	Decentralized
QPLEX [32]	Off-Policy	Value-Based	Simultaneous	Centralized	Decentralized
IPPO [27]	On-Policy	Policy-Based	Simultaneous	Decentralized	Decentralized
MAPPO [17]	On-Policy	Policy-Based	Simultaneous	Centralized	Decentralized
HAPPO [31]	On-Policy	Policy-Based	Sequential	Centralized	Decentralized
A2PO [12]	On-Policy	Policy-Based	Sequential	Centralized	Decentralized
MAT [11]	On-Policy	Policy-Based	Simultaneous	Centralized	Centralized

Task	IPPO	MAPPO	HAPPO	A2PO	MAT	QMIX	QPLEX
pass & shoot	93.7(0.9)	92.9(2.6)	94.0(3.6)	93.2(1.2)	**96.6**(0.8)	95.6(5.8)	88.1(8.2)
run pass & shot	73.7(13.4)	66.0(6.3)	70.4(7.2)	79.9(6.0)	**81.1**(5.7)	58.1(23.6)	68.8(14.2)
3 vs 1	**91.7**(3.1)	90.0(3.2)	91.4(3.9)	87.6(1.4)	88.5(2.0)	86.9(8.2)	81.9(6.7)
corner	50.4(10.3)	50.5(7.2)	47.9(9.9)	59.7(6.2)	**71.0**(8.1)	20.0(18.7)	28.8(16.7)
ct-easy	85.7(6.6)	88.9(6.6)	78.0(14.8)	80.9(7.3)	**89.3**(5.8)	57.5(18.9)	43.3(27.2)
ct-hard	71.6(4.9)	81.3(9.6)	75.2(10.6)	80.7(4.2)	**87.0**(6.0)	56.3(18.2)	33.8(25.6)
5 vs 5	99.1(0.6)	96.0(1.8)	98.0(1.3)	95.0(2.9)	**99.3**(1.2)	0.0(0.0)	0.0(0.0)
11 vs 11	52.7(2.4)	45.4(2.7)	52.1(4.8)	50.1(3.6)	**59.7**(3.6)	0.0(0.0)	0.0(0.0)

Song, Yan et al. Boosting Studies of Multi-Agent Reinforcement Learning on Google Research Football Environment: the Past, Present, and Future. AAMAS 2024.

两种范式在机器人Agent上统一：多层联合工作和训练的新架构

- 范式A（LLM Agent）：以大语言模型为中心，配合API选择的决策
- 范式B（Large RL Models）：以强化学习大模型为中心，做底层的决策控制



https://wholebody-b1.github.io/

两种范式在机器人Agent上统一：多层联合工作和训练的新架构

- 范式A（LLM Agent）：以大语言模型为中心，配合API选择的决策
- 范式B（Large RL Models）：以强化学习大模型为中心，做底层的决策控制

1~3hz

Consciousness 意识层
建立机器自主意识

3~10hz

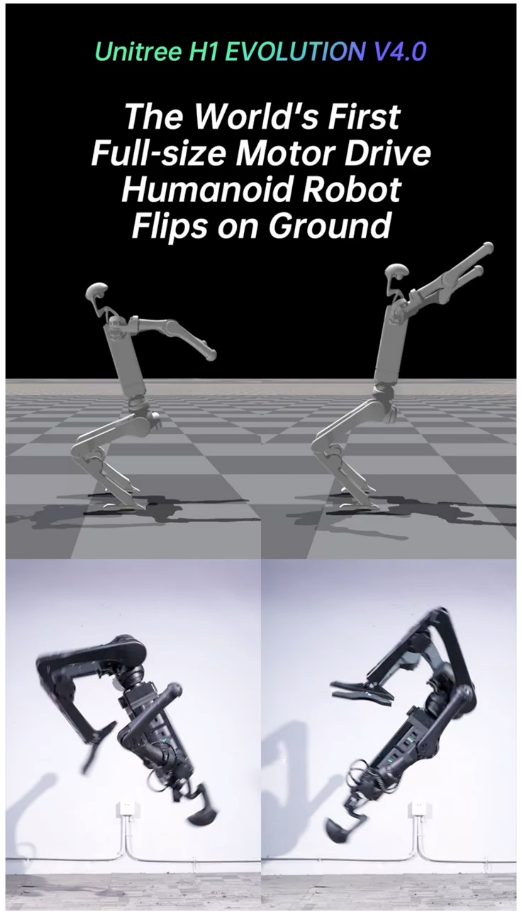
Cognition 认知层
基于LLM Agent做认知推理、任务规划

10~100hz

Perception 感知层
多模态感知环境

Control 控制层
给出控制信号

Intelligence



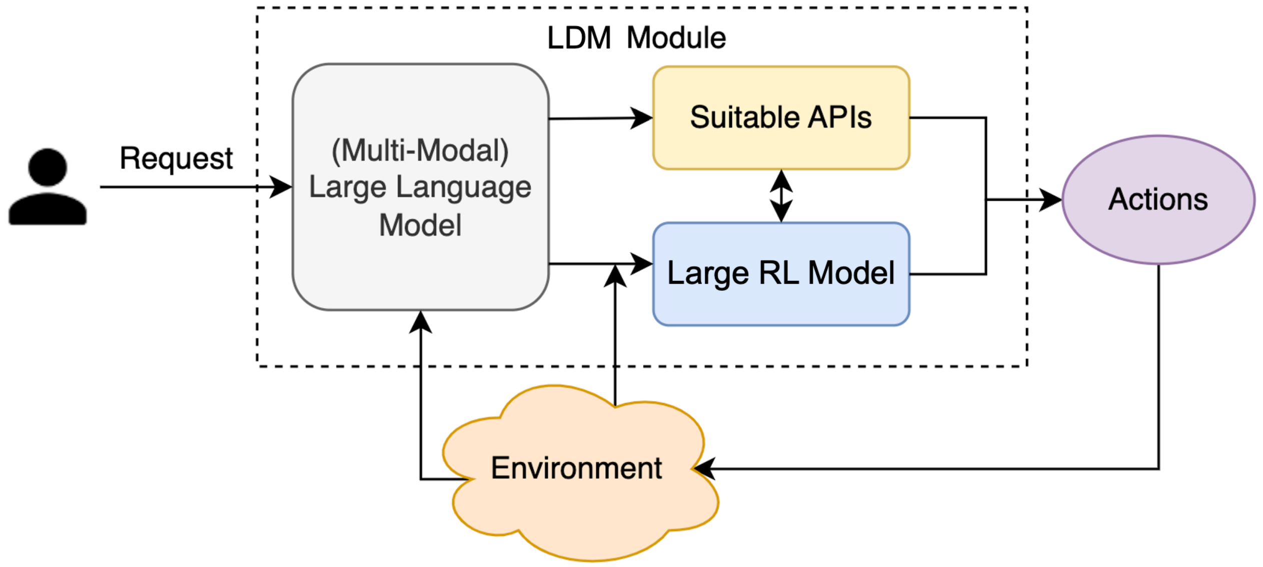
Agility

总结：决策式大模型，打造通用决策智能体的基础能力

- 决策智能面临开放变化环境情况下的泛化问题
- 决策式大模型：使用大模型解决决策任务中的环境变化、开放环境、策略泛化性问题

决策大模型是新一代人工智能底层技术：

- 范式A（LLM Agent）：以大语言模型为中心，配合API选择的决策
  - 反思、工具使用、规划、多智能体协作
  - 检索增强、对齐、MCTS
- 范式B（Large RL Models）：以强化学习大模型为中心，做底层的决策控制
  - 策略大模型、价值大模型、环境大模型
  - 多智能体交互大模型
- 范式融合：多层联合工作和训练的新架构，发展新一代具身智能技术



交大团队部分相关论文

1. Large Decision Models. IJCAI 2023. https://www.ijcai.org/proceedings/2023/0808.pdf
2. On Realization of Intelligent Decision-Making in the Real World: A Foundation Decision Model Perspective. CAAI AIR 2023. https://arxiv.org/abs/2212.12669
3. Sim-to-Real Transfer for Quadrupedal Locomotion via Terrain Transformer. ICRA 2023. https://arxiv.org/abs/2212.07740
4. Multi-embodiment Legged Robot Control as a Sequence Modeling Problem. ICRA 2023. https://arxiv.org/abs/2212.09078
5. Multi-agent reinforcement learning is a sequence modeling problem. NeurIPS 2022. https://arxiv.org/abs/2205.14953
6. Large Sequence Models for Decision-Making: A Survey. FCS 2023. https://arxiv.org/abs/2306.13945
7. Offline pre-trained multi-agent decision transformer: One big sequence model conquers all starcraftii tasks. MIR 2023. https://arxiv.org/abs/2112.02845
8. Bootstrapped Transformer for Offline Reinforcement Learning. NeurIPS 2022. https://arxiv.org/abs/2206.08569
9. GEAR: A GPU-Centric Experience Replay System for Large Reinforcement Learning Models. ICML 2023. http://proceedings.mlr.press/v202/wang23aj/wang23aj.pdf
10. Adaptive Control Strategy for Quadruped Robots in Actuator Degradation Scenarios. DAI 2023. https://doi.org/10.1145/3627676.3627686
11. Alphazero-like tree-search can guide large language model decoding and training. Arxiv 2023. https://arxiv.org/abs/2309.17179
12. Vision-Language Foundation Models as Effective Robot Imitators. ICLR 2024. https://arxiv.org/abs/2311.01378
13. TRAD: Enhancing LLM Agents with Step-Wise Thought Retrieval and Aligned Decision. SIGIR 2024. https://arxiv.org/abs/2403.06221