第九章 大模型强化学习

修贤超

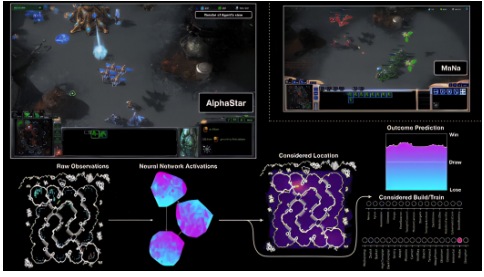
https://xianchaoxiu.github.io

致谢：本教案由张鹏飞协助准备

目录

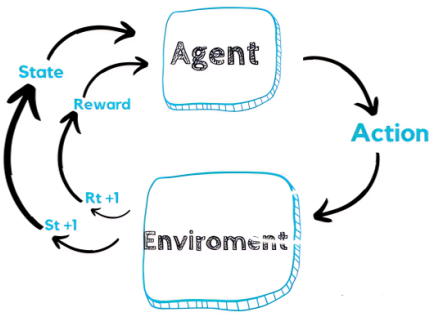
简介

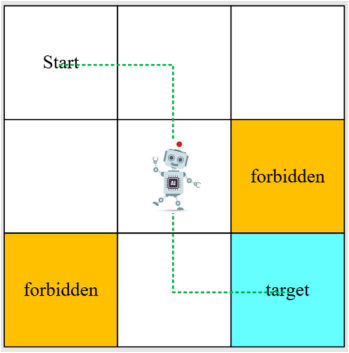
- 强化学习（Reinforcement Learning, RL）是一种模仿人类通过试错学习的 AI 方法，它不需要明确标签，而是通过奖励信号指导学习



简介

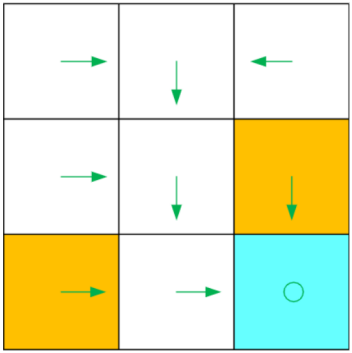
- 强化学习基础概念

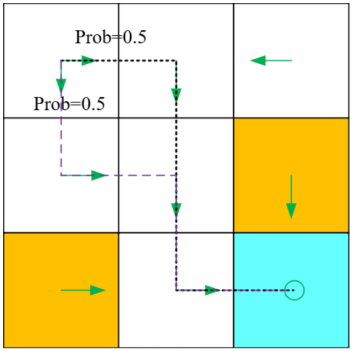

Agent
State
Reward
Action
Rt +1
St +1
Enviroment


Start
forbidden
forbidden
target

简介

- 状态 State / 动作 Action / 策略 Policy / 奖励 Reward

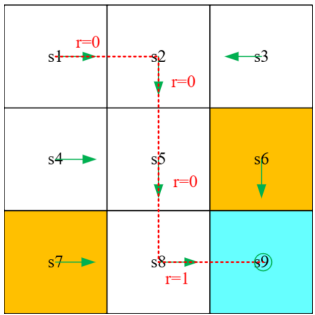


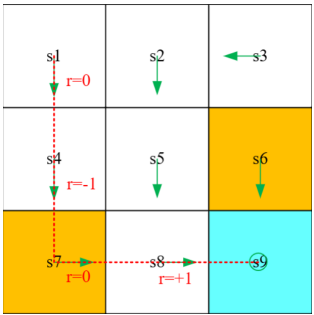


简介

- 轨迹 Trajectory / 回报 Return

$$s_1 \xrightarrow[r=0]{a_2} s_2 \xrightarrow[r=0]{a_3} s_5 \xrightarrow[r=0]{a_3} s_8 \xrightarrow[r=1]{a_2} s_9$$

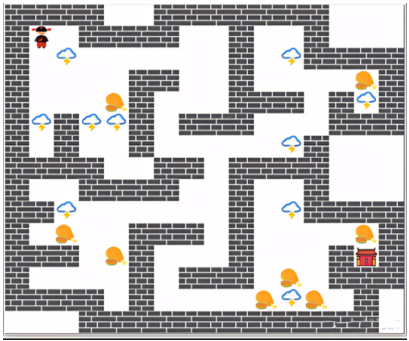




简介

- 探索和利用指智能体在决策时需要在尝试未知动作以获取新信息 (探索) 与选择已知最优动作以获得最大回报 (利用) 之间取得平衡

探索（Exploration）	利用（Exploitation）
发现未知的可能性	使用已知最优策略
有机会获得更高奖励	提高短期收益
但风险更高，效率低	但可能陷入局部最优



目录

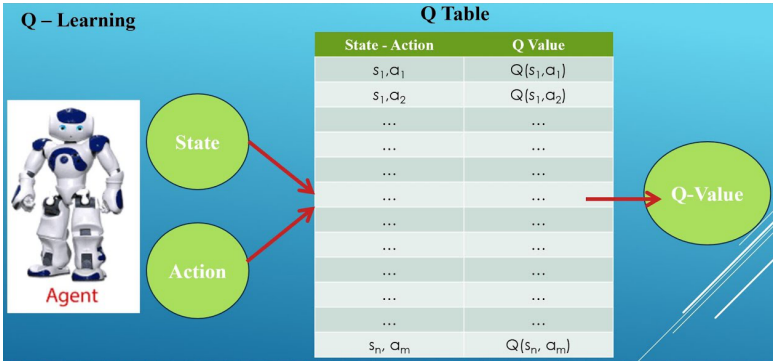
简介

主要算法类型

类型	核心思路	优势	代表算法
基于价值 (Value-based)	计算状态或动作的价值	理论成熟、易实现	Q-Learning, DQN
基于策略 (Policy-based)	直接优化策略函数	连续动作空间更自然	REINFORCE
Actor-Critic 混合	策略+价值双网络	稳定且高效	A3C, PPO
基于模型 (Model-based)	模拟环境预测未来	样本效率高	MuZero

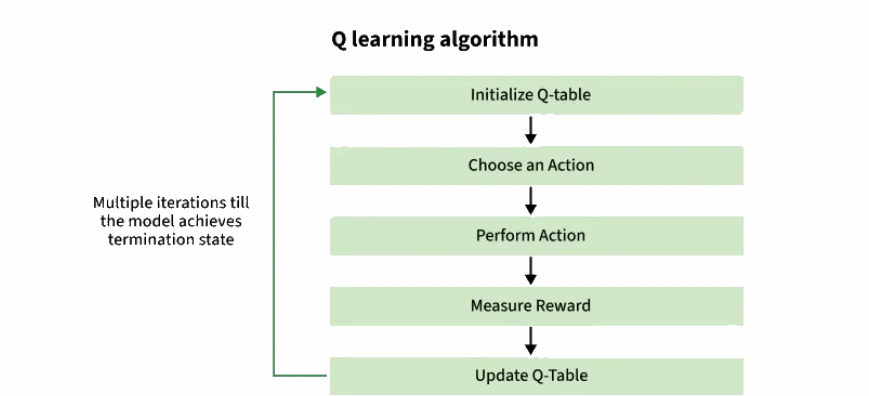
相关算法

- Q-learning 是一种基于价值的强化学习算法, 核心目标是学习出 Q 函数, 也就是状态 s 和动作 a 的组合能带来多少长期奖励



相关算法

- Q-learning 算法流程



相关算法

- https://virtual-labs.github.io/exp-q-learning-iiith/simulation.html

Q[(1,1),L] ← 0.000 + 0.100 * (-0.100 + 0.900 * 0.000 - (0.000)) = -0.010

Previous Iteration

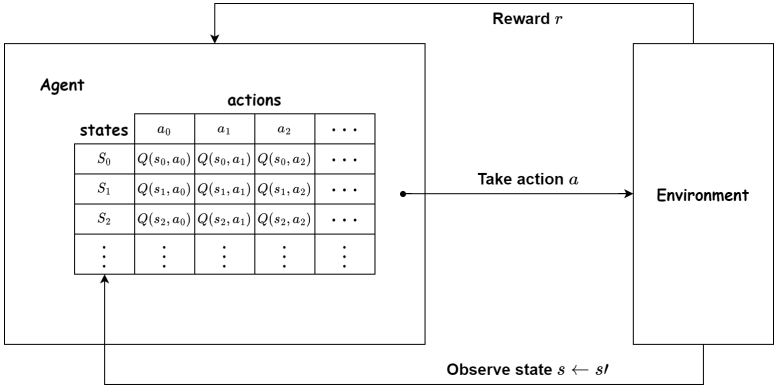
L : 0.000 U : 0.000 R : 0.000 D : 0.000 ←	L : 0.000 U : 0.000 R : 0.000 D : 0.000 ←	1.000
	L : 0.000 U : 0.000 R : 0.000 D : 0.000 ←	-1.000
L : -0.029 U : -0.020 R : -0.027 D : -0.020 ↑	L : -0.020 U : 0.000 R : 0.000 D : 0.000 ↑	L : 0.000 U : 0.000 R : 0.000 D : 0.000 ←

Present Iteration

L : 0.000 U : 0.000 R : 0.000 D : 0.000	L : 0.000 U : 0.000 R : 0.000 D : 0.000	1.000
	L : -0.010 U : 0.000 R : 0.000 D : 0.000	-1.000
L : -0.029 U : -0.020 R : -0.034 D : -0.020	L : -0.020 U : -0.010 R : 0.000 D : 0.000	L : 0.000 U : 0.000 R : 0.000 D : 0.000

相关算法

- Q-learning 靠 Q 表学策略, 不断探索、利用、更新, 最终学出最优路径, 让智能体从乱走变成会走

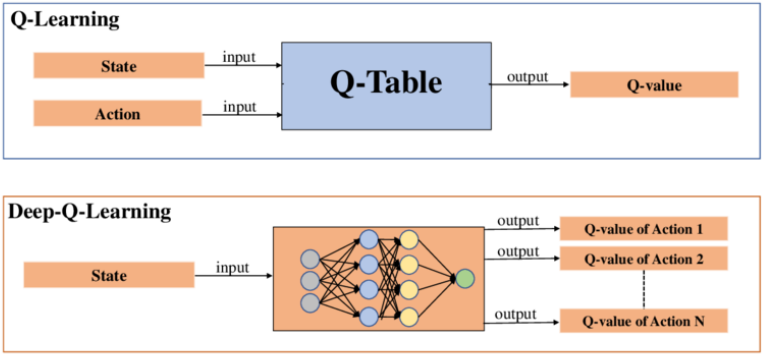


相关算法

- 当状态空间过大时，Q 表根本存不下

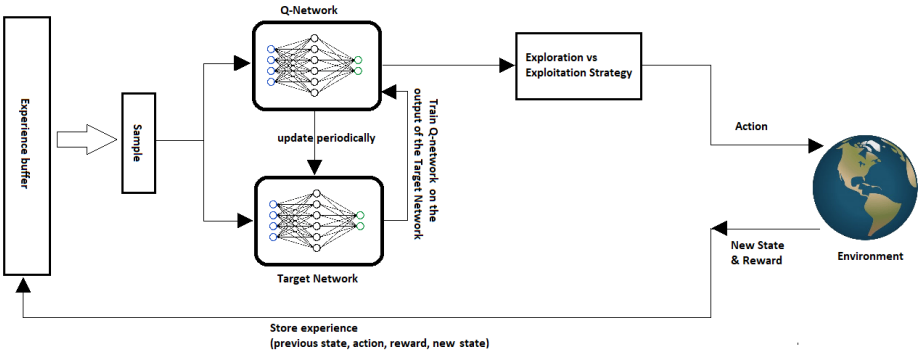
相关算法

- Deep Q Network 把神经网络引入到 Q-learning 中, 用神经网络来近似 Q 函数, 这样就能输入原始图像、输出每个动作的 Q 值



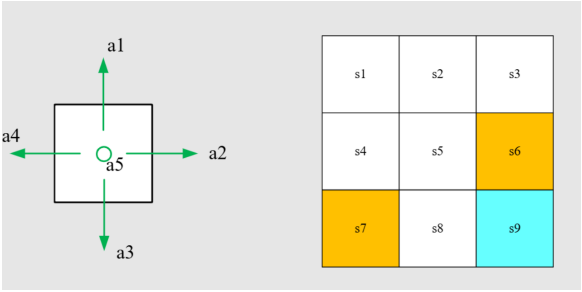
相关算法

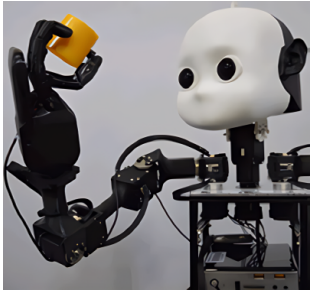
- DQN 算法流程



相关算法

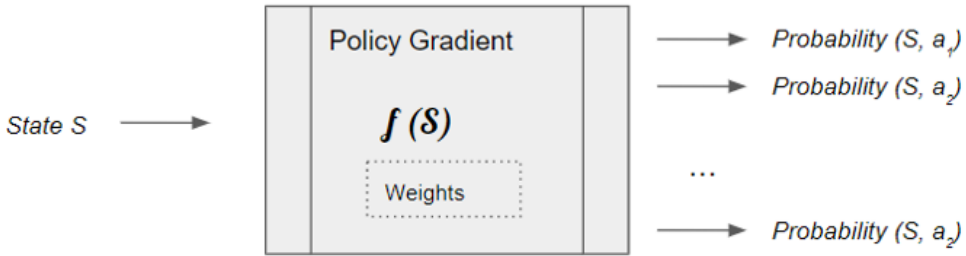
- Policy Gradient是一种基于策略的强化学习算法，是一种通过直接对策略参数进行梯度上升来最大化期望回报的强化学习方法





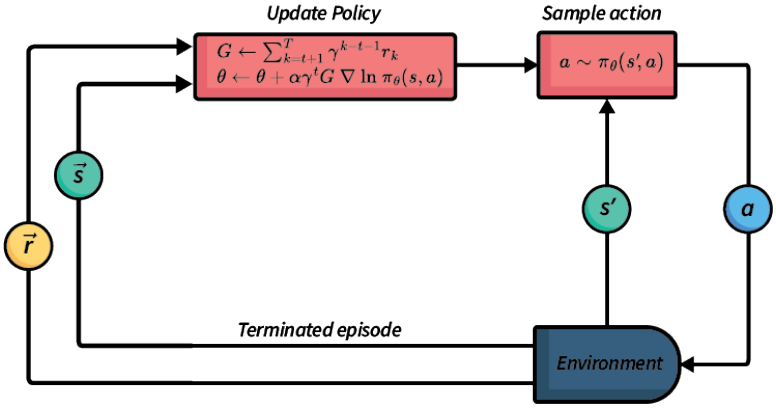
相关算法

- Policy Gradient 不是通过估计价值函数来间接决定动作, 而是直接学习策略函数, 即学习一个函数 $\pi_\theta(a \mid s)$, 描述在状态 s 下采取动作 a 的概率



相关算法

- Policy Gradient 流程



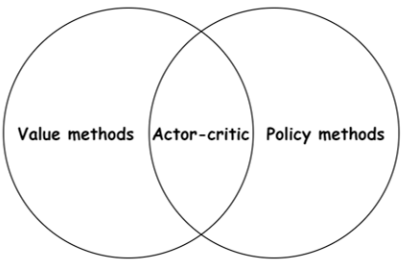
相关算法

- DQN 和 Policy Gradient

方法	思想	优点	局限
DQN	学 Q 值函数 Q(s,a)	稳定，样本效率高	动作必须离散；策略间接得到
Policy Gradient	学策略 π_a	可处理连续动作	但“不稳”，更新波动大

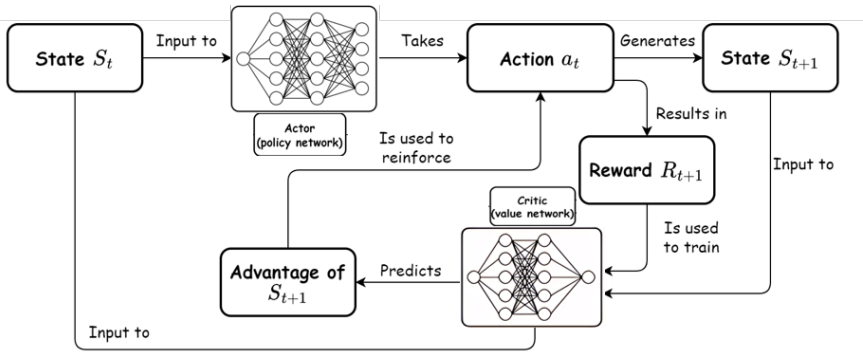
相关算法

- Actor-Critic 用价值估计稳定策略学习



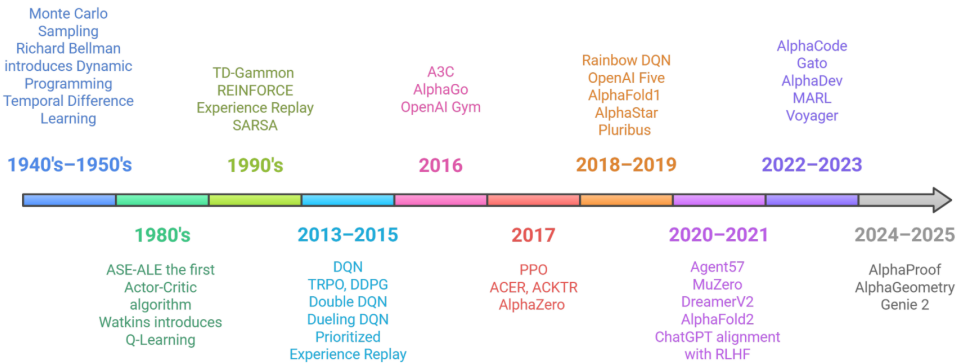
相关算法

- Actor 负责决策, Critic 负责评估, 二者共同优化策略



相关算法

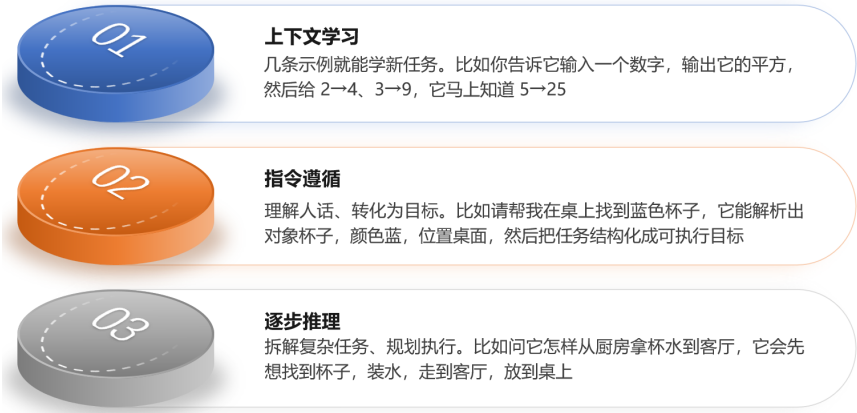
- 总结


Monte Carlo
Sampling
Richard Bellman
introduces Dynamic
Programming
Temporal Difference
Learning
1940's–1950's
TD-Gammon
REINFORCE
Experience Replay
SARSA
1990's
A3C
AlphaGo
OpenAI Gym
2016
Rainbow DQN
OpenAI Five
AlphaFold1
AlphaStar
Pluribus
2018–2019
AlphaCode
Gato
AlphaDev
MARL
Voyager
2022–2023
1980's
ASE-ALE the first
Actor-Critic
algorithm
Watkins introduces
Q-Learning
2013–2015
DQN
TRPO, DDPG
Double DQN
Dueling DQN
Prioritized
Experience Replay
2017
PPO
ACER, ACKTR
AlphaZero
2020–2021
Agent57
MuZero
DreamerV2
AlphaFold2
ChatGPT alignment
with RLHF
2024–2025
AlphaProof
AlphaGeometry
Genie 2

目录

- RL 存在的问题

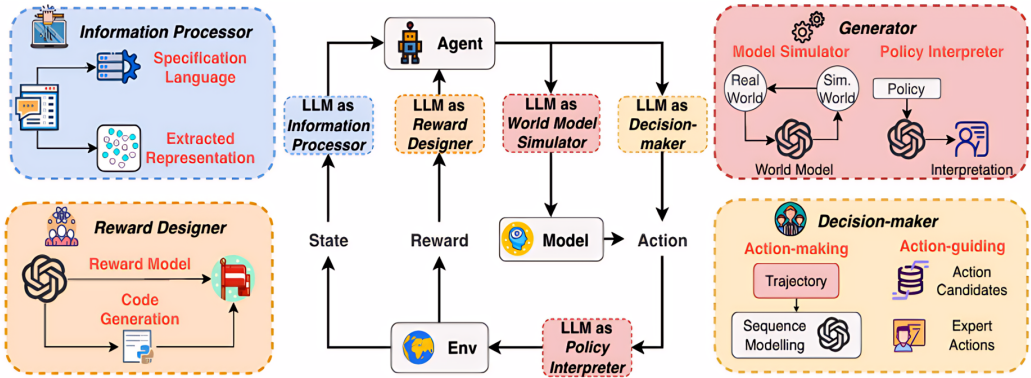
为什么 LLM 能补上 RL 的短板



LLM 增强强化学习的优势

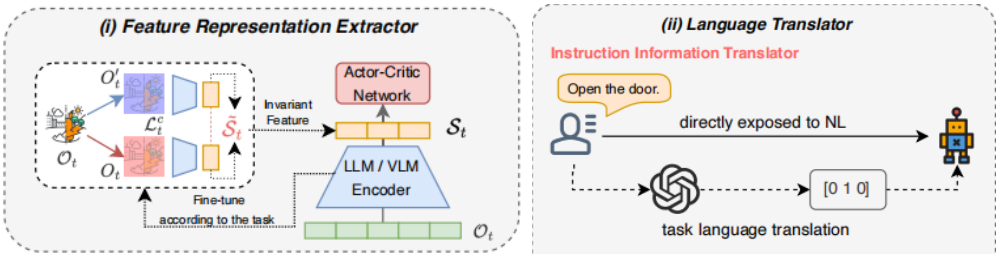
	传统 RL	LLM增强型RL
泛化能力	依赖大量试错	利用语言与知识迁移到新任务
奖励设计	手工定义困难	LLM 可生成或评估奖励信号
规划与推理	短视、局部最优	可分解任务、长程规划
样本效率	需要海量交互	利用先验知识减少探索
可解释性	策略“黑箱”	可用自然语言解释行为

- Survey on Large Language Model-Enhanced Reinforcement Learning: Concept, Taxonomy, and Methods, 2025



大模型强化学习

- LLM 作为信息处理者, 将复杂的输入转化为可理解的结构化信息, 帮助 RL 智能体聚焦核心任务



大模型强化学习

主要应用方向

多模态融合

将图像、文本、语音输入对齐成统一语义表示。例如RT-2，把图像 + 自然语言直接融合成机器人动作

语言理解与任务解析

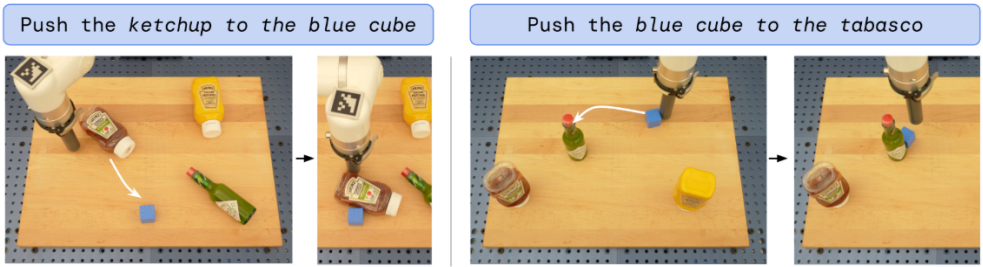
SayCan(Google, 2022)把帮我把饮料放进冰箱分解成找到饮料 → 打开冰箱 → 放入 → 关门

状态总结与经验反思

从交互日志中提炼失败原因，帮助智能体自我纠错与改进策略

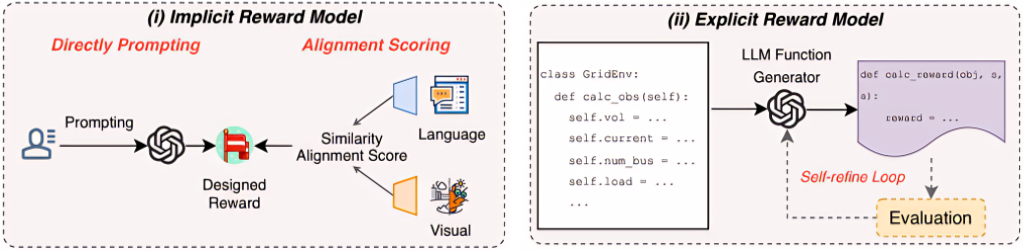
大模型强化学习

- https://robotics-transformer2.github.io/



大模型强化学习

- LLM 作为奖励设计者, 设计、生成或评估智能体的奖励信号, 动态调整奖励函数, 使智能体学习更稳定

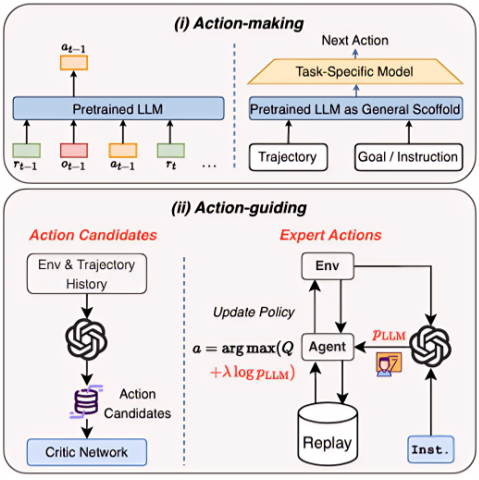


- LLM 作为奖励设计者

应用方向	说明	示例
基于人类反馈的奖励	LLM 模拟人工评审，生成奖励信号	InstructGPT、ChatGPT
语言自评奖励	LLM 自用语言规则评估输出质量	Anthropic Constitutional AI
基于语义对齐的奖励	比较输出与目标在语义上的一致性	文本摘要、视觉描述任务
奖励函数生成与自我修正	LLM 直接生成奖励函数并持续优化	Explicit Reward Model

大模型强化学习

- LLM 作为决策者

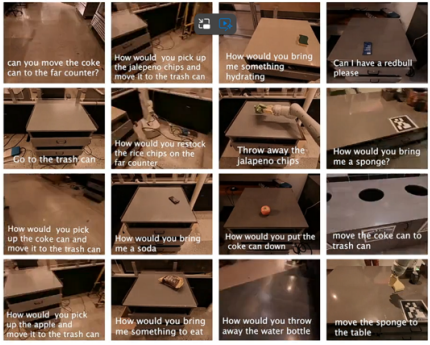


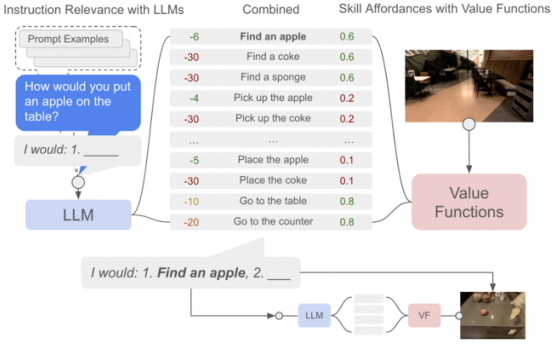
- LLM 作为决策者（Decision Maker）

方法/范式	核心思想	适用场景与要点
语言规划 + 价值筛选	先由 LLM 产出候选步骤/技能，再用价值网络评估可行性/成功率后执行	适合多步骤任务与具身/机器人场景
序列化决策	将决策视为序列生成，基于历史与目标回报直接预测下一动作	适合离线数据、稀疏奖励
ReAct + 工具使用	先写思考链，不确定时先检索/调用工具再行动，降低幻觉、提升可解释性与成功率	适合需要外部知识或工具的任务
高层技能/子目标选择	在大动作空间中提出高层动作/子目标，由价值/约束模块筛选并做安全把关	适合长程规划与分层控制

大模型强化学习

- https://say-can.github.io/



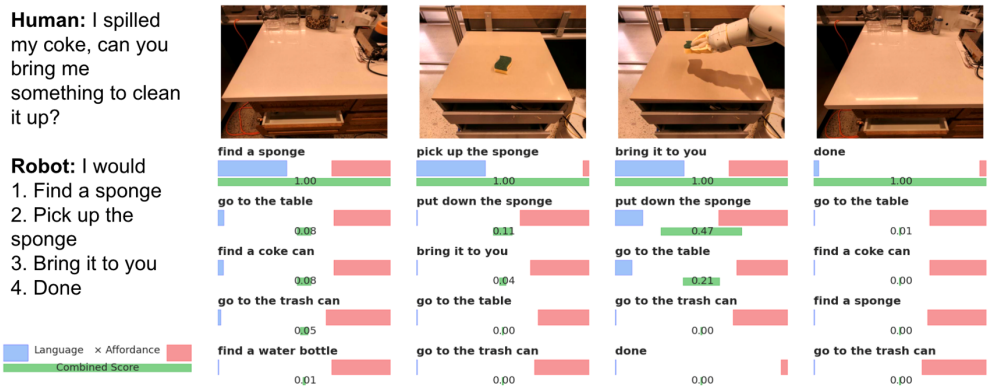


大模型强化学习

- 语言规划 + 价值筛选

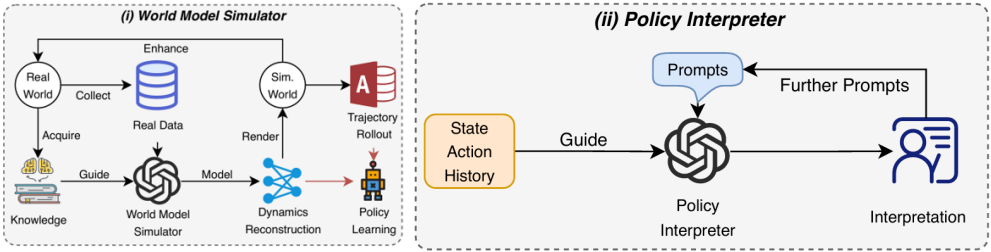
Human: I spilled my coke, can you bring me something to clean it up?

Robot: I would
1. Find a sponge
2. Pick up the sponge
3. Bring it to you
4. Done



大模型强化学习

- LLM 作为生成器



大模型强化学习

- LLM 作为生成器

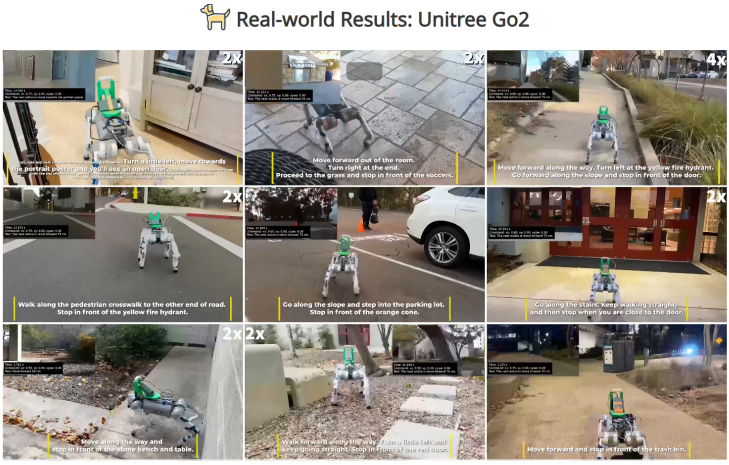
应用领域	功能说明	例子
任务生成	根据目标自动生成任务描述、目标、约束等	TextWorld生成多种任务关卡
世界模型生成	生成环境状态或行为脚本供模拟器使用	TeslaFSD中生成边缘环境用于训练极端场景
技能库生成	生成新技能/子目标并加以实现	Voyager模型不断自创和更新技能库
自适应任务生成	根据智能体执行情况不断调节任务难度	动态任务生成与优化（RL课程生成）

大模型强化学习

- https://www.tesla.com/fsd

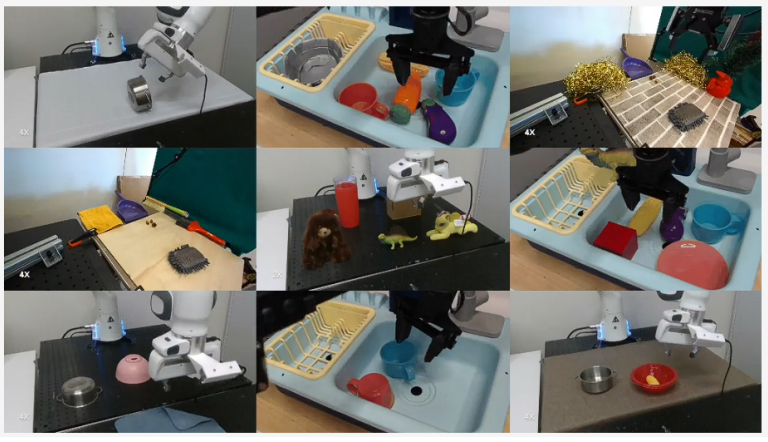
目录

- https://navila-bot.github.io/



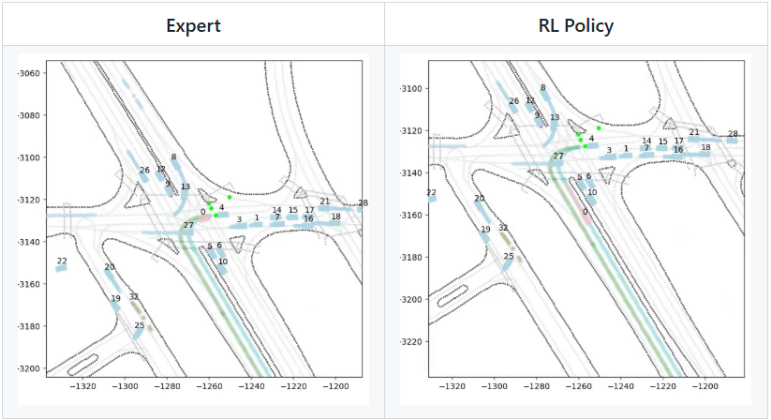
VLA

- https://openvla.github.io/



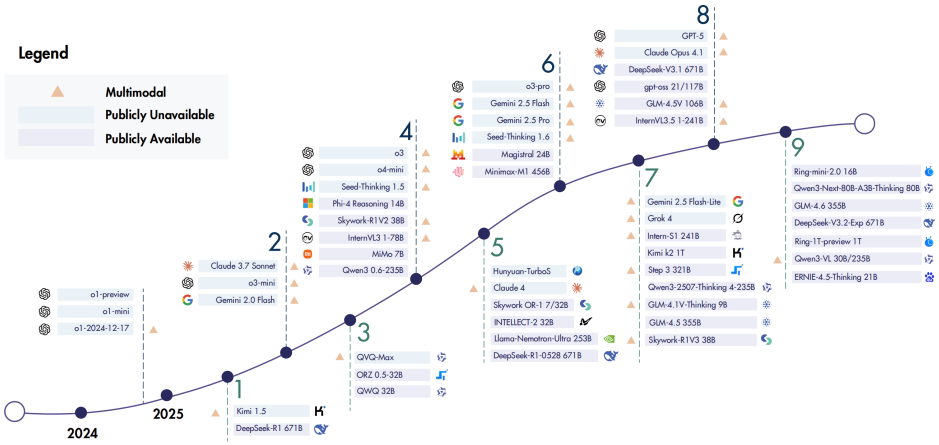
自动驾驶

- https://github.com/valeoai/v-max



展望

- A Survey of Reinforcement Learning for Large Reasoning Models, 2025



Q&A

Thank you!

感谢您的聆听和反馈