大模型自主智能体：从 AgentBench 到 AutoGLM

Xiao Liu

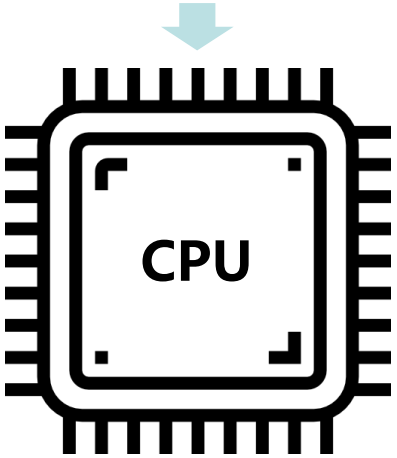
2024.11.06

What is LLM in essence?

- LLM 的工作本质：一个通用的智能处理器（CPU）

指令 + *数据*

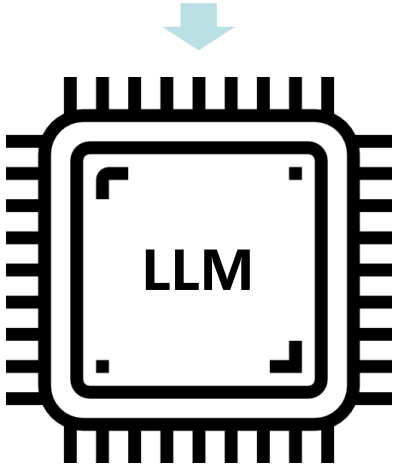
IN: Add *11 13*



计算结果

OUT: 24

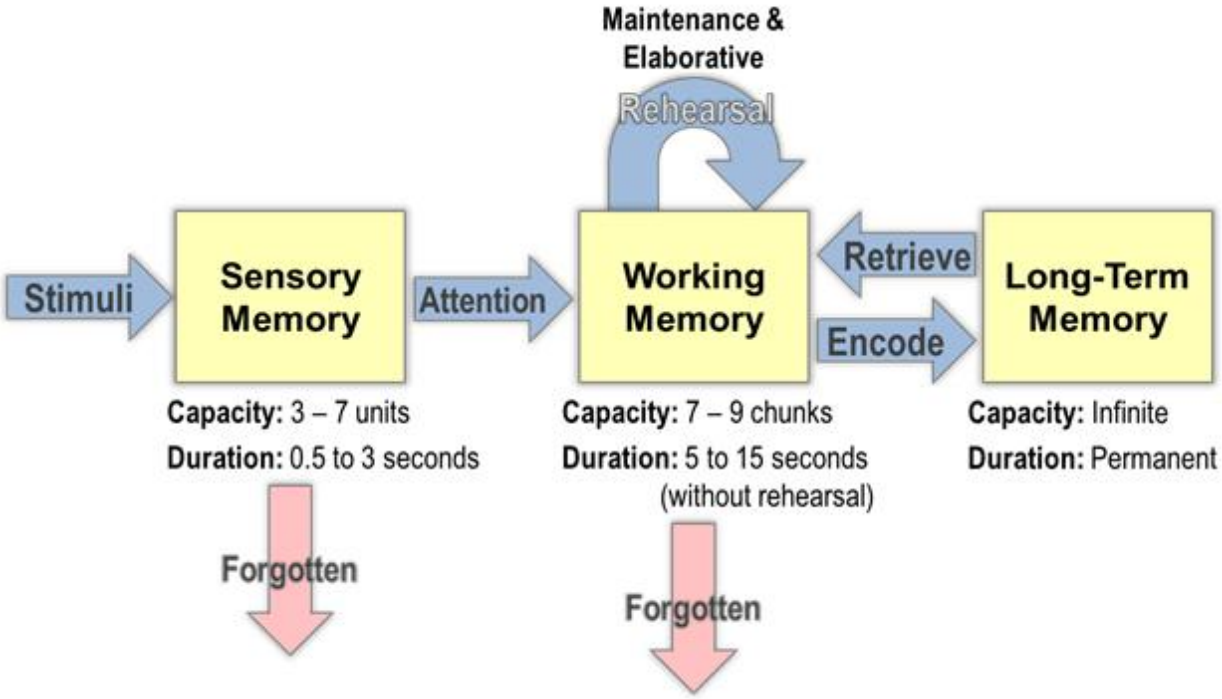
IN: 帮我总结一下这篇文章 + [*输入的文章内容*]



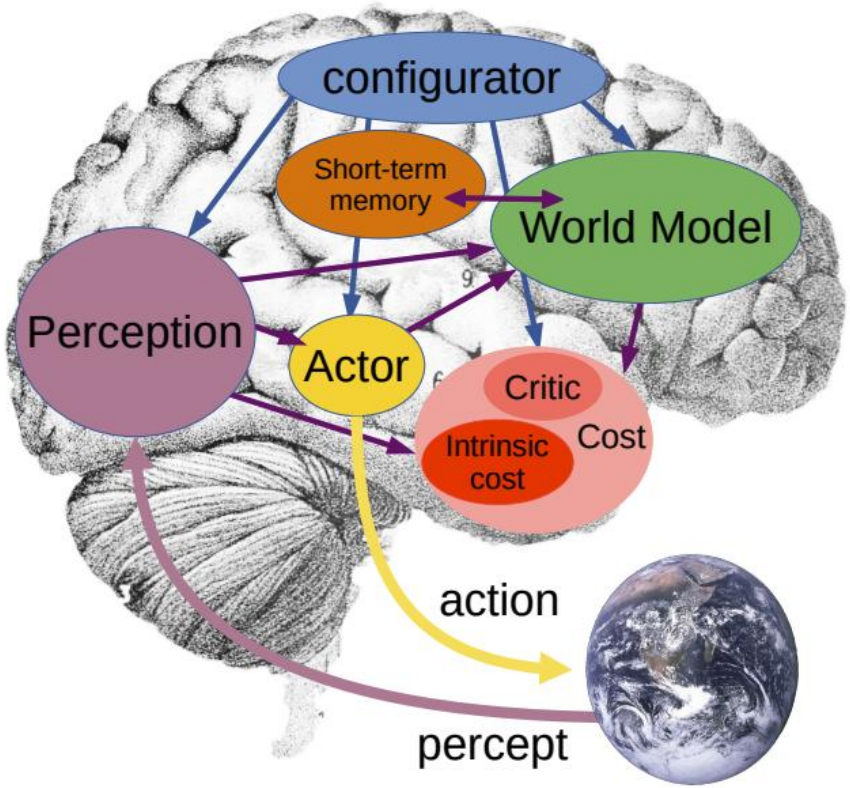
OUT: 这篇文章主要讲了：1) xxx; 2) xxx; 3) xxx;

- 光有 CPU 不能造出计算机！

Real Autonomous Intelligence: System 1&2



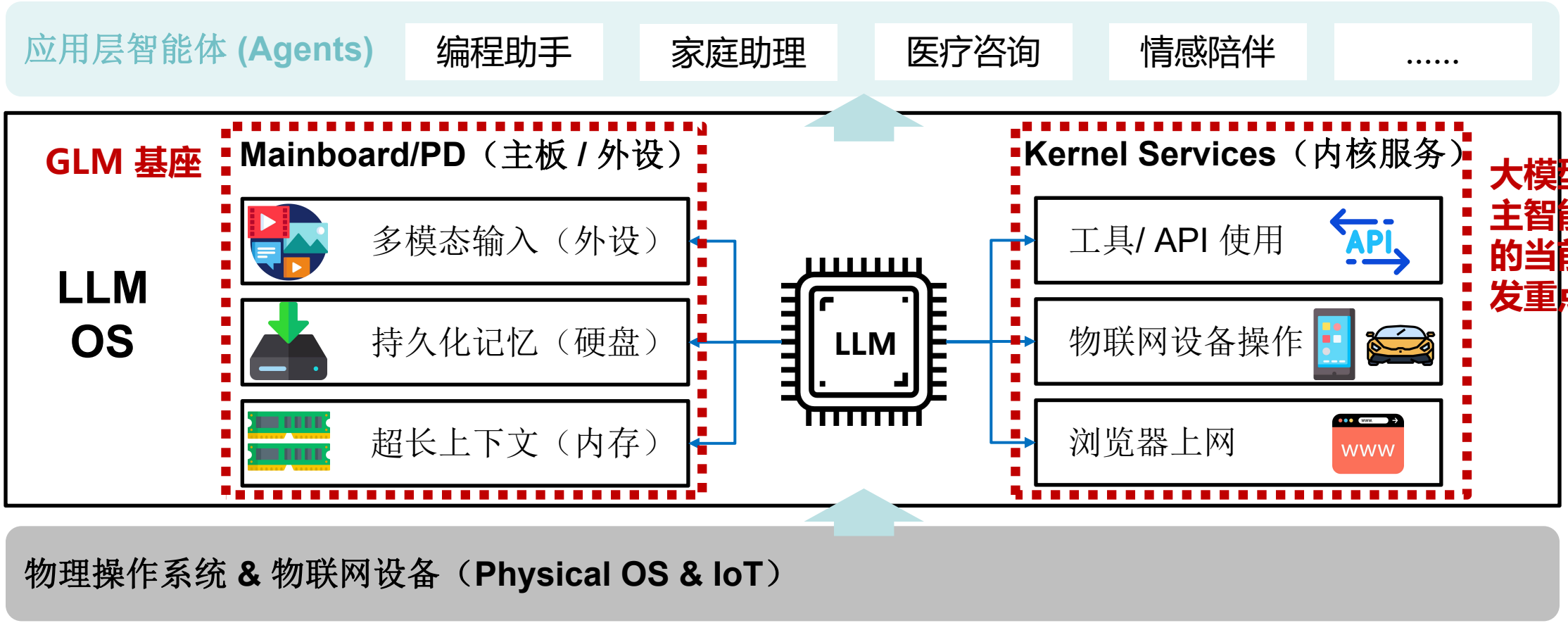
Baddeley's Model of Working Memory



LeCun, Yann. "A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27." *Open Review* 62.1 (2022).

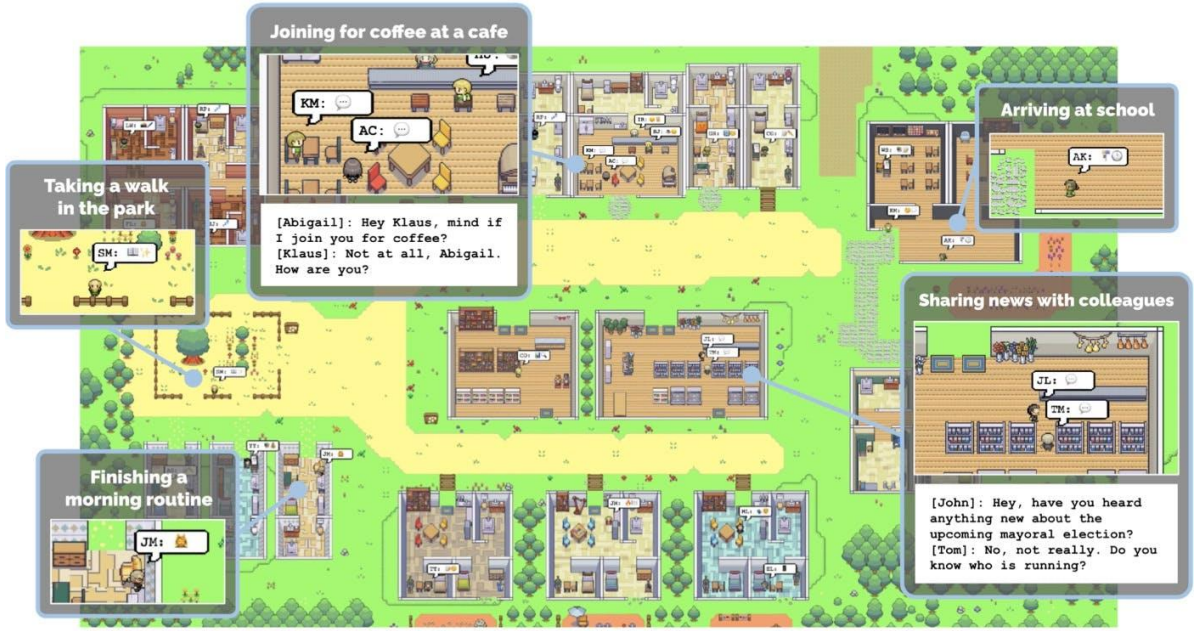
What we really want

- 以 LLM 为中心构建的通用计算系统（通用人工智能）



LLM-as-Agent：大模型作为智能体

- 大模型具有作为自主智能体的潜力
  - Stanford Generative Agents: A *"Westworld"* with 25 agents
  - Auto-GPT：GitHub 史上星标最多的 AI 项目



OpenAI's AGI Roadmap

OpenAI Imagines Our AI Future

Stages of Artificial Intelligence

Level 1	Chatbots, AI with conversational language
Level 2	Reasoners, human-level problem solving
Level 3	Agents, systems that can take actions
Level 4	Innovators, AI that can aid in invention
Level 5	Organizations, AI that can do the work of an organization

Source: Bloomberg reporting

OpenAI o1
初步达到的水平

下一步计划
(2025-2026)

Content

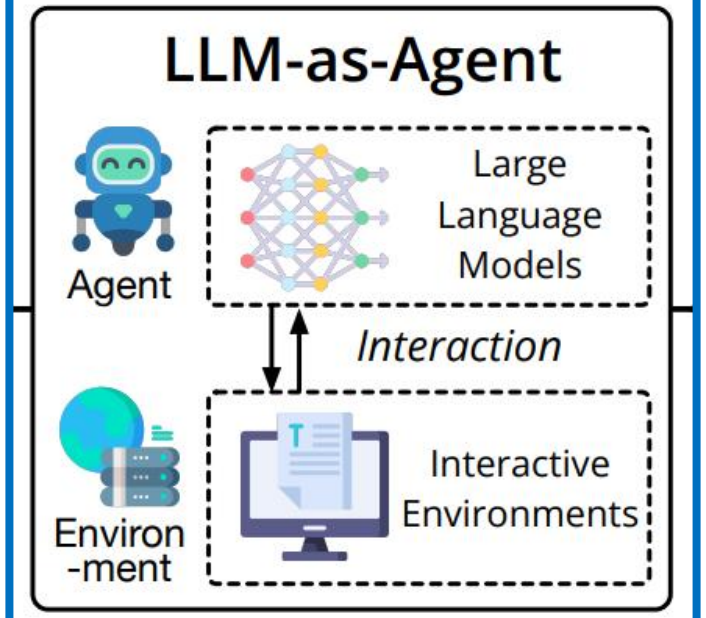
- **大模型有多擅长作为智能体?**
- 如何提高大模型作为智能体的水平?
- 如何在在线环境中评估智能体?
- 如何在线环境中持续提升智能体能力?

挑战：大模型有多擅长作为智能体?

- Missing A Systematic LLM-as-Agent Benchmark
  - Tasks & Datasets: no standard tasks
  - Models: no comprehensive coverage of LLMs

Real-world Mission

- High-level goals
- Multi-turn interactions
- Planning & Reasoning
- ……



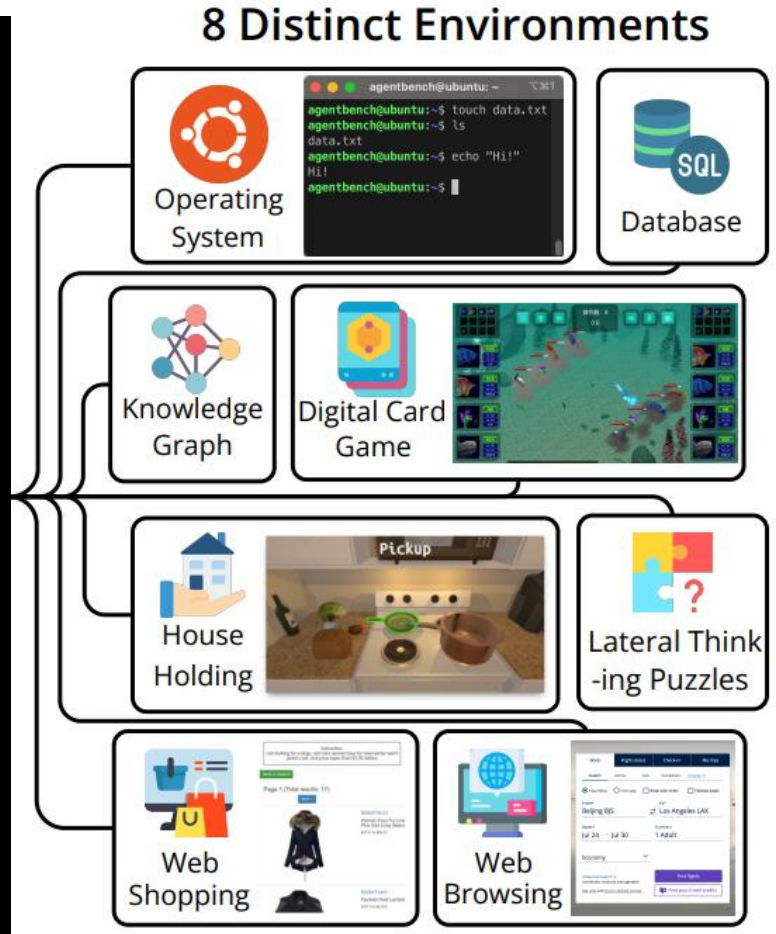
Standard Evaluation

- Diverse environments
- Quantitative results
- Unified prompting
- ……

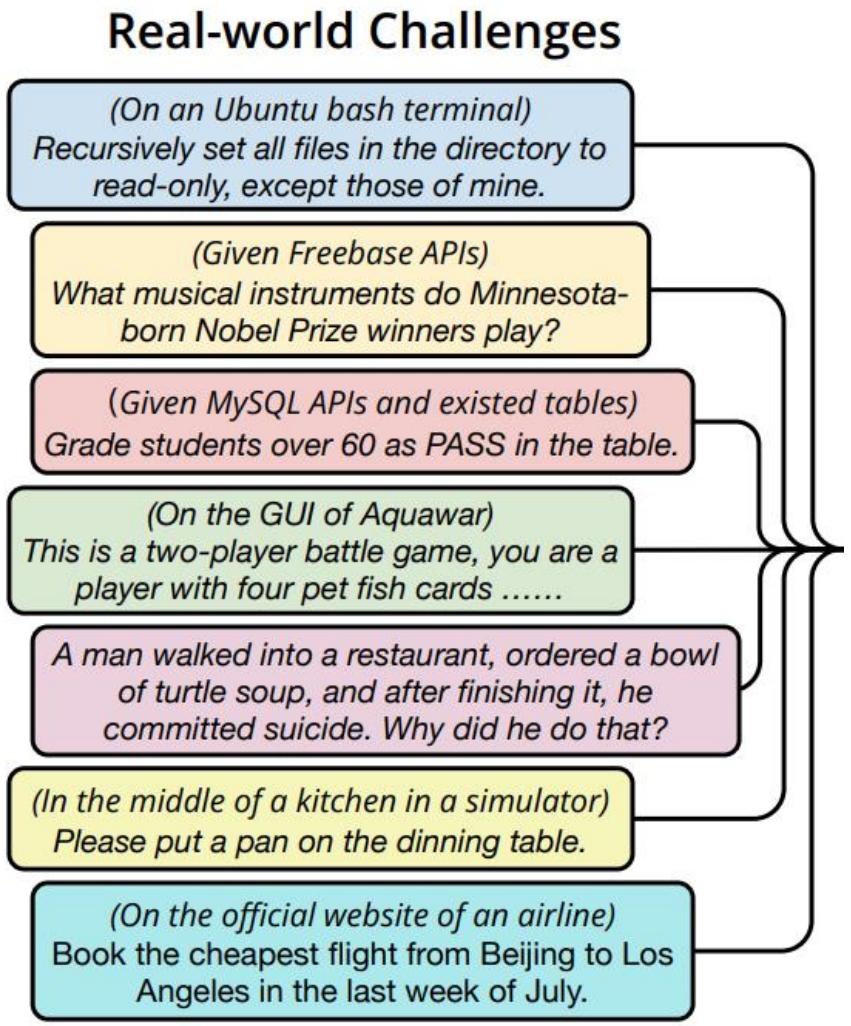
AgentBench：评估大模型作为智能体潜力

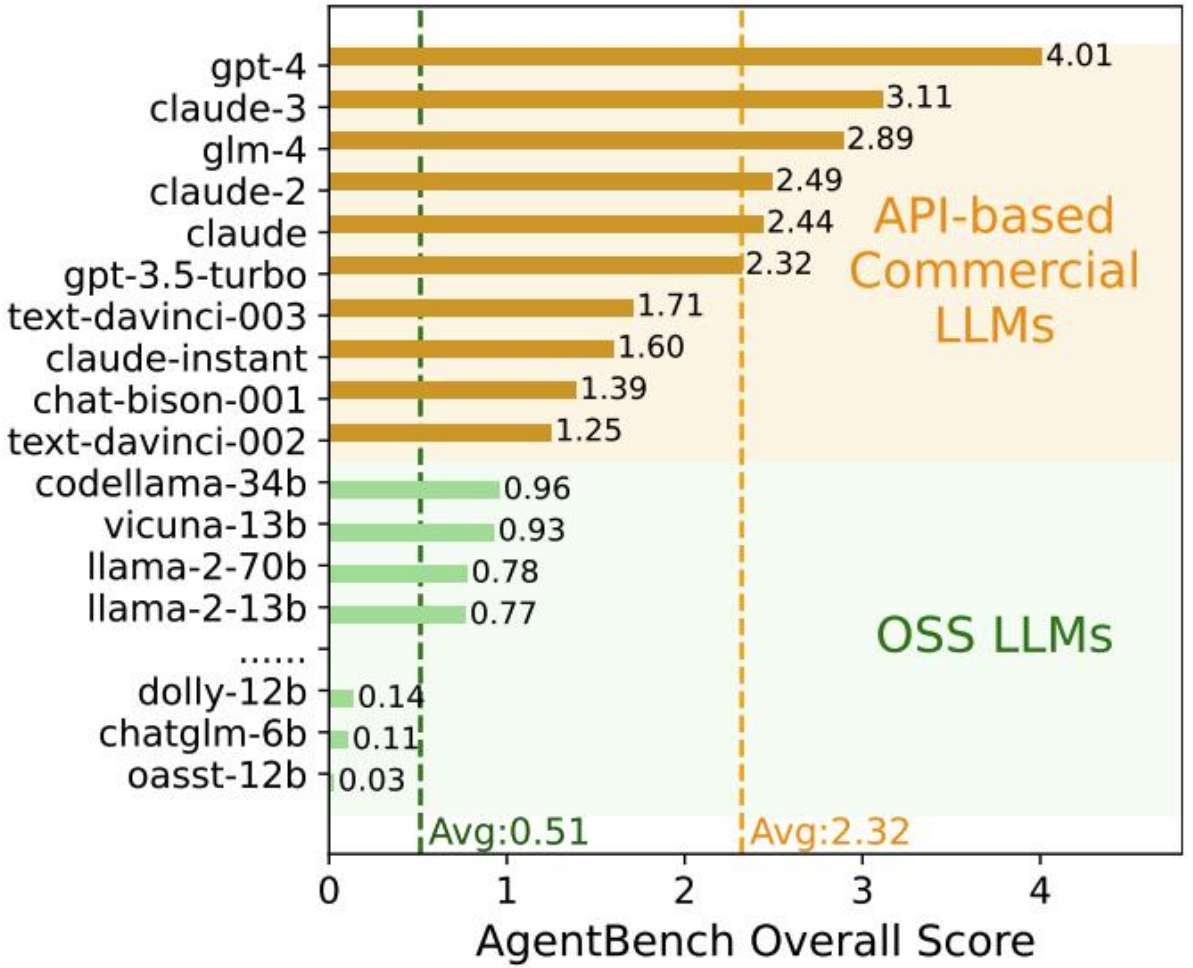
- AgentBench: Evaluating LLMs as Agents (23.08, ICLR’ 24)

AgentBench



AgentBench：评估大模型作为智能体潜力





Content

- 大模型有多擅长作为智能体?
- **如何提高大模型作为智能体的水平?**
- 如何在在线环境中评估智能体?
- 如何在线环境中持续提升智能体能力?

挑战：如何提高大模型作为智能体的水平？

- 代表性场景：GUI（图形化用户交互界面）
  - 个人电脑、智能手机、浏览器
- 代表性数据集：Mind2Web（离线采集人类完成网页浏览轨迹）

Task Description:

Show me the reviews for the auto repair business closest to 10002.

Action Sequence:

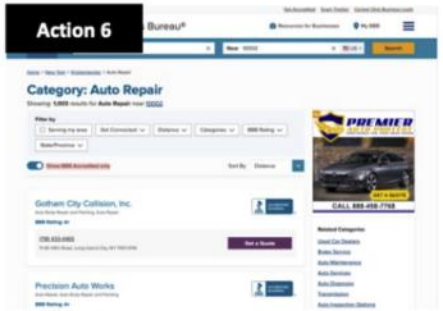
	Target Element	Operation
1.	[searchbox] *Find*	TYPE: *auto repair*
2.	[button] *Auto Repair*	CLICK
3.	[textbox] *Near*	TYPE: *10002*
4.	[button] *10002*	CLICK
5.	**[button] Search**	**CLICK**
6.	[switch] *Show BBB Accredited only*	CLICK
7.	[svg]	CLICK
8.	[button] Sort By	CLICK
9.	**[link] *Fast Lane 24 Hour Auto Repair***	**CLICK**
10.	[link] *Read Reviews*	CLICK

Webpage Snapshots:

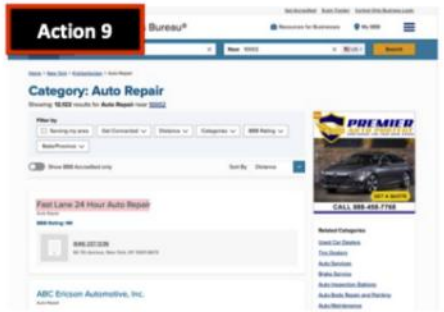
<input name="find_text" type="search">

Auto Repair

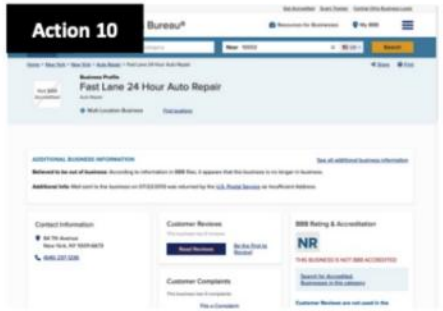
<button>Search</button>



<button>Show BBB Accredited only</button>



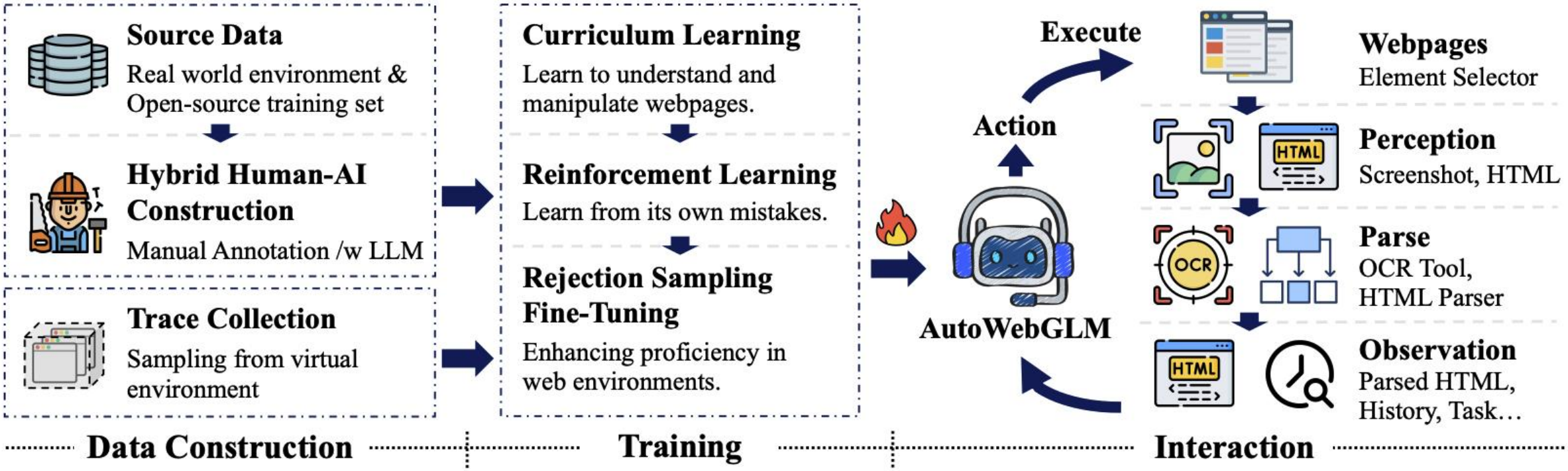
Fast Lane 24 Hour Auto Repair



Read Reviews

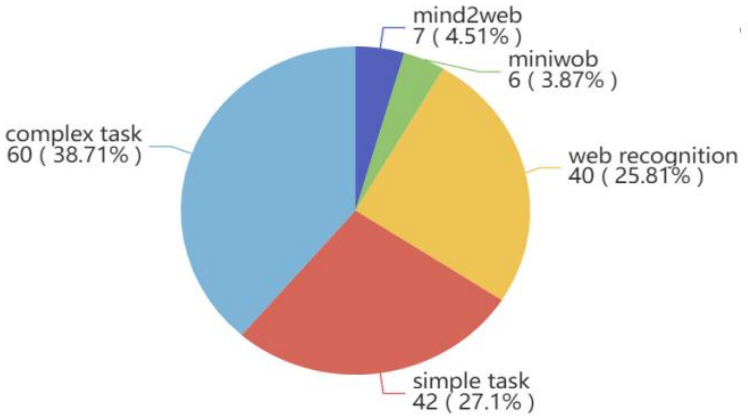
AutoWebGLM：自主网页浏览智能体

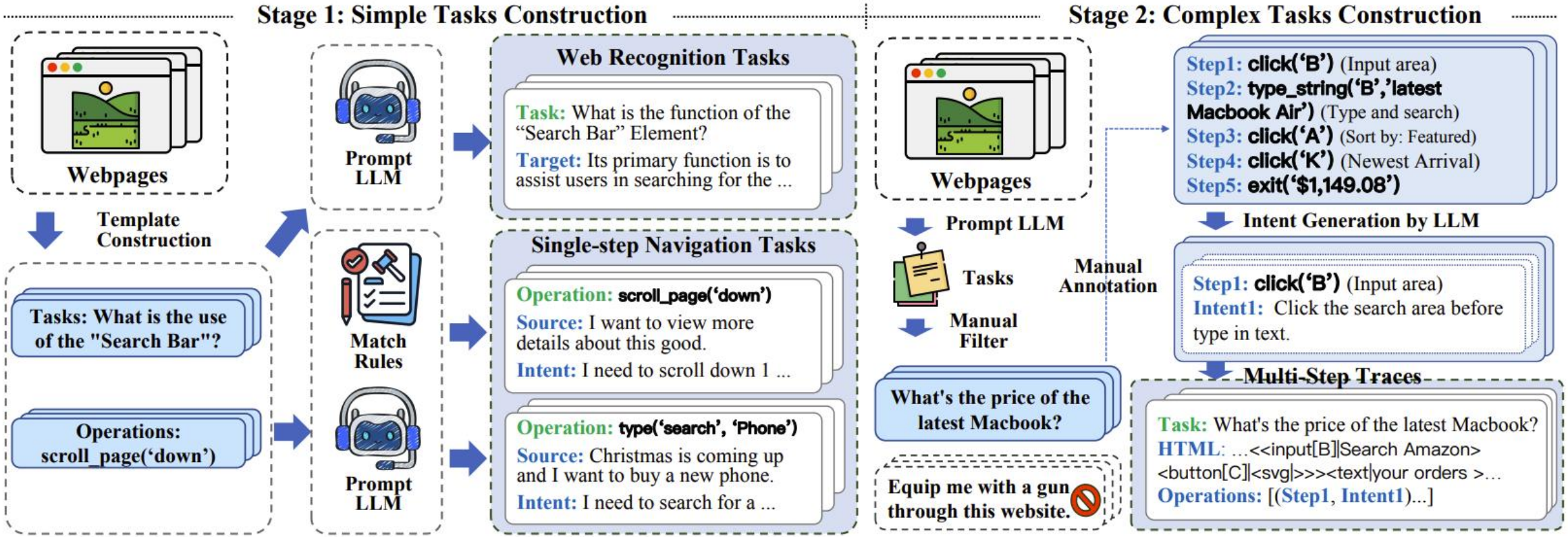
- 构建大模型自主智能体十分具有挑战性
  - 数据构造：缺乏智能体轨迹数据用于训练
  - 训练方法：传统 SFT 训练效果差
  - 真实部署：在真实网络环境中如何真正运行起来



数据搜集

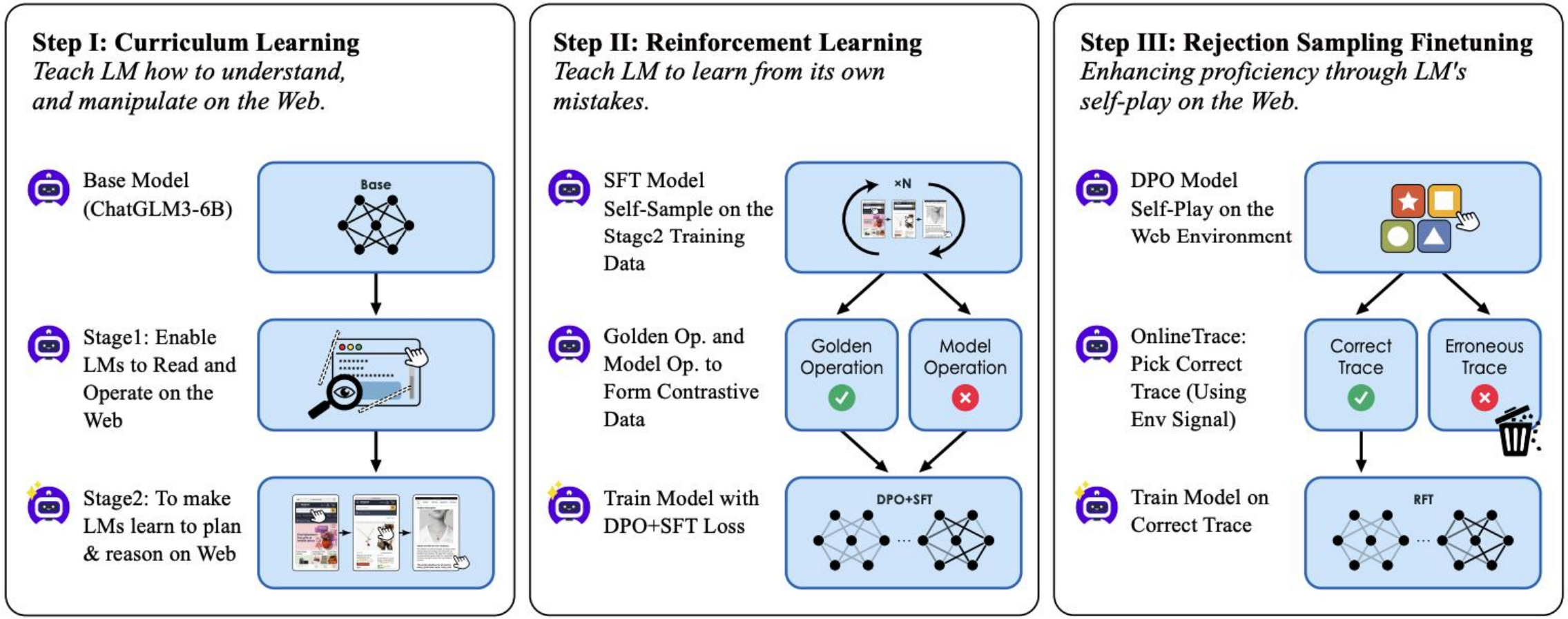
- 第一阶段：简单任务数据搜集（全自动）
- 第二阶段：复杂任务数据搜集（人工+AI辅助）





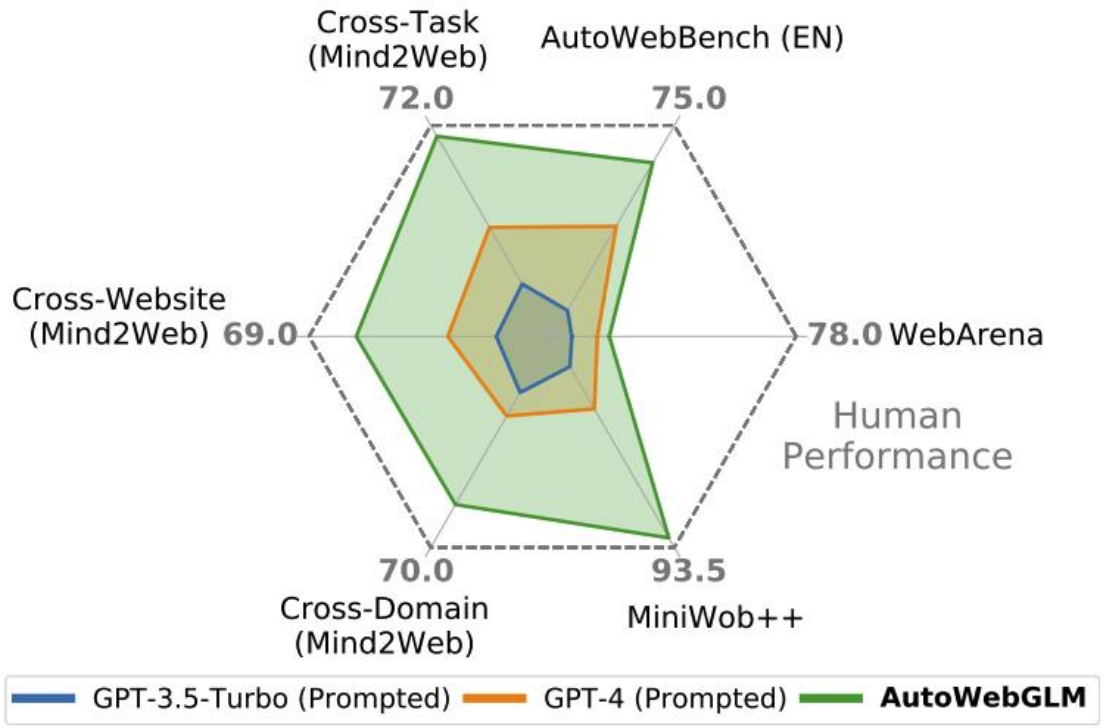
训练框架

- 课程学习 -> DPO 偏好学习 -> 拒绝采样微调



实验结果

- 基于 ChatGLM3-6B 训练
- 在 Mind2Web, WebArena, 以及 AutoWebBench 超越 GPT-4
  - 在最有挑战性的 WebArena 上，和人类表现仍有差距

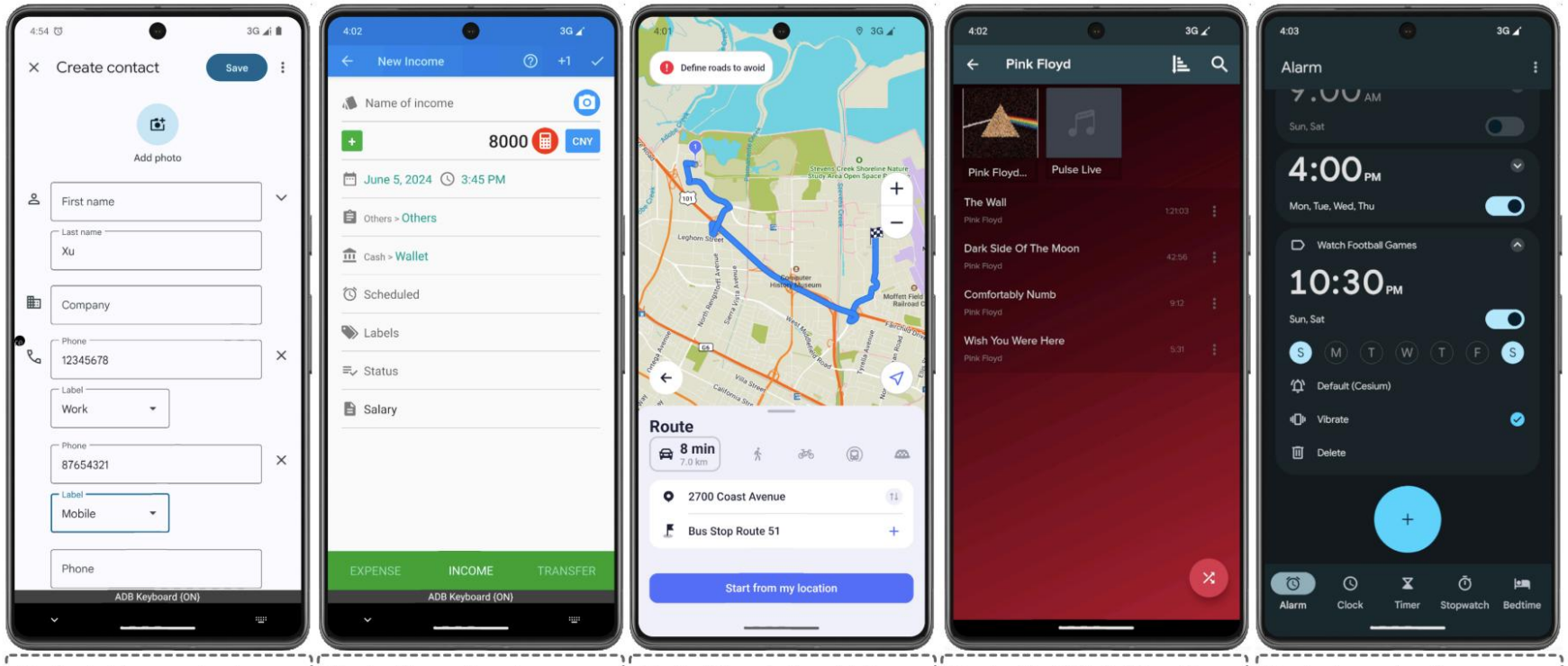


Content

- 大模型有多擅长作为智能体?
- 如何提高大模型作为智能体的水平?
- **如何在在线环境中评估智能体?**
- 如何在线环境中持续提升智能体能力?

挑战：离线数据集难以有效评估智能体水平

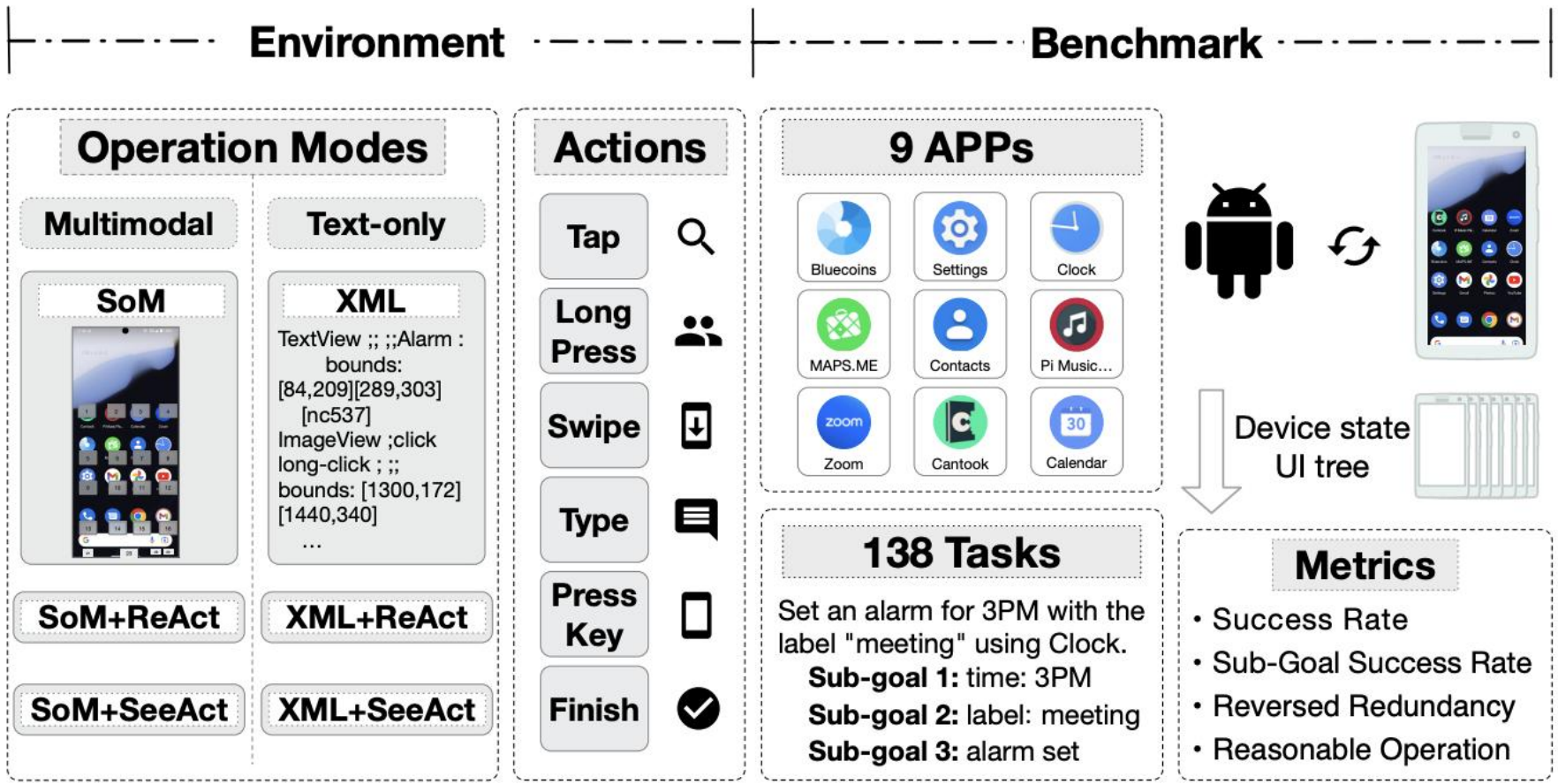
- 过去的GUI智能体评估工作的问题：
  - **可交互性和可复现性**：需要平衡
  - **真实成功率评估**：同一个任务有多个不同的可行解
  - **忽视开源模型的训练和提升**：普遍基于闭源 API 搭建智能体



Task: Add a contacts whose name is Xu, set the working phone…	Task: Record an income of 8000 CNY in the books, and mark it as…	Task: Check the driving distance and time between Bus stop of…	Task: Sort Pink Floyd's songs by duration time in descending order.	Task: I need set an 10:30PM clock every weekend, and label it…
Sub-Goals: · Name: Xu · Working phone number · Mobile phone number	Sub-Goals: · Enter: New income · Cash: 8000 · Note: …	Sub-Goals: · Driving distance · Driving time	Sub-Goals: · Page: ARTISTS · Artist: Pink Floyd · Order: Descending	Sub-Goals: · Time: 10:30PM · Frequency: Weekend · Label: …

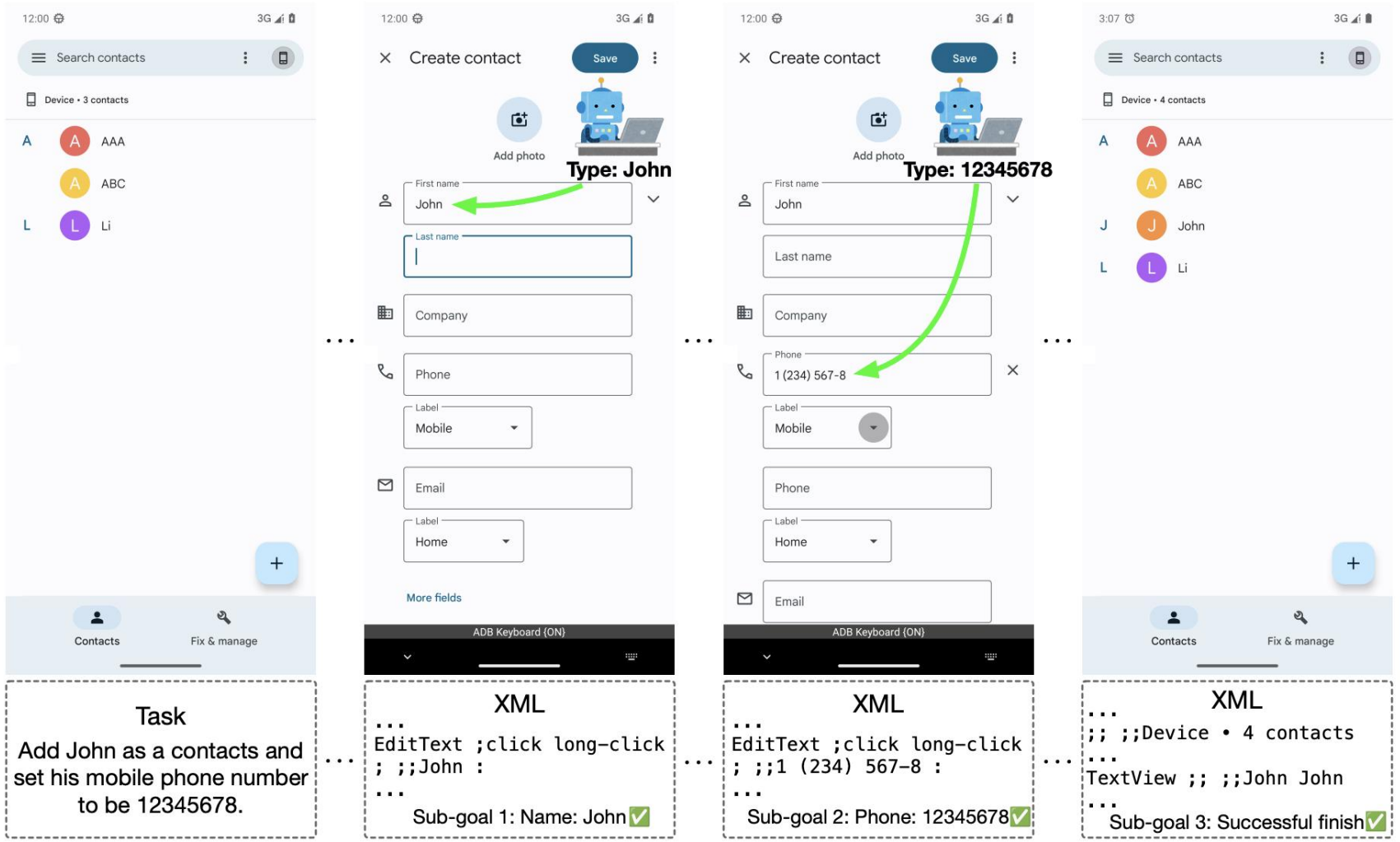
AndroidLab：交互环境和基准测试

- 交互环境
  - 通过利用XML信息和set-of-mark方法，构造了在文本模型和多模态模型完全等价的操作空间，保证公平比较
  - 进一步适配了ReAct和SeeAct两种推理后输出的框架
- 基准测试
  - 共包含 9个app，138个任务，覆盖多样化的实际使用场景
  - 保证复现性：固定的设备状态、时间设置及固定应用使用记录



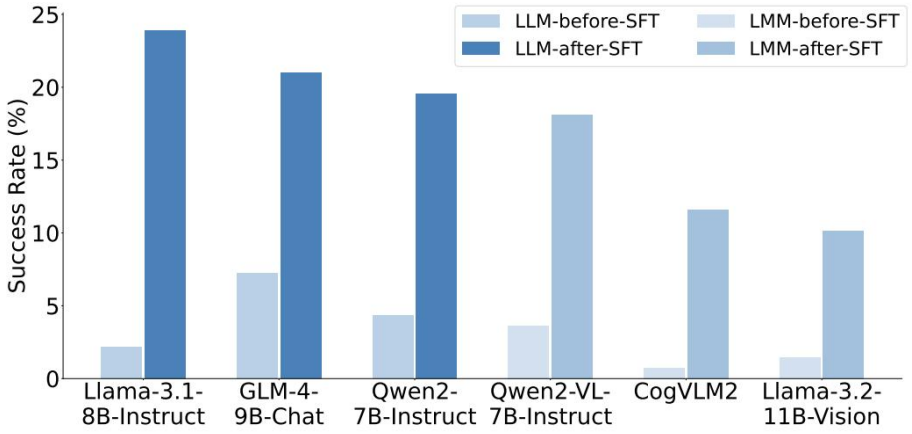
AndroidLab：评价指标

- **任务完成率：**每个任务仅在所有子目标均成功完成时计为成功。
- **子目标成功率：**将任务拆分为多个子目标，评估模型在每个步骤表现。
- **冗余率：**比较模型的操作路径与人类最优路径的长度，计算冗余操作的程度。
- **合理操作比率：**评估每次操作是否合理，无效操作（如点击无效区域）视为不合理。



AndroidLab：评估结果

- 数据收集过程
  - 任务生产与扩展
  - 自动化探索
  - 人工标注和校验
- 指令微调训练
  - 在LLM和LMM都达到了超过15%的平均增长
  - 最优开源模型接近gpt-4o能力



(b) Success Rates of before and after fine-tuning by Android Instruct.

Mode	Model	SR	Sub-SR	RRR	ROR
XML	GPT-4o	25.36	30.56	**107.45**	86.56
	GPT-4-1106-Preview	**31.16**	**38.21**	66.34	86.24
	Gemini-1.5-Pro	18.84	22.40	57.72	83.99
	Gemini-1.0	8.70	10.75	51.80	71.08
	GLM4-PLUS	27.54	32.08	92.35	83.41
	LLaMA3.1-8B-Instruct	2.17	3.62	-	52.77
	Qwen2-7B-Instruct	4.35	4.95	-	67.26
	GLM4-9B-Chat	7.25	9.06	54.43	58.34
XML+SFT	LLaMA3.1-8B-**ft**	23.91	30.31	75.58	92.46
	Qwen2-7B-**ft**	19.57	24.40	77.31	92.48
	GLM4-9B-**ft**	21.01	26.45	74.81	**93.25**
SoM	GPT-4o	**31.16**	**35.02**	87.32	85.36
	GPT-4-Vision-Preview	26.09	29.53	99.22	78.79
	Gemini-1.5-Pro	16.67	18.48	105.95	**91.52**
	Gemini-1.0	10.87	12.56	72.52	76.70
	Claude-3.5-Sonnet	28.99	32.66	**113.41**	81.16
	Claude-3-Opus	13.04	15.10	81.41	83.89
	CogVLM2	0.72	0.72	-	17.97
	LLaMA3.2-11B-Vision-Instruct	1.45	1.45	-	50.76
	Qwen2-VL-7B-Instruct	3.62	4.59	-	84.81
SoM+SFT	CogVLM2-**ft**	11.59	16.06	57.37	85.58
	LLaMA3.2-11B-Vision-**ft**	10.14	12.98	61.67	87.85
	Qwen2-VL-7B-Instruct-**ft**	18.12	22.64	65.23	88.29

Content

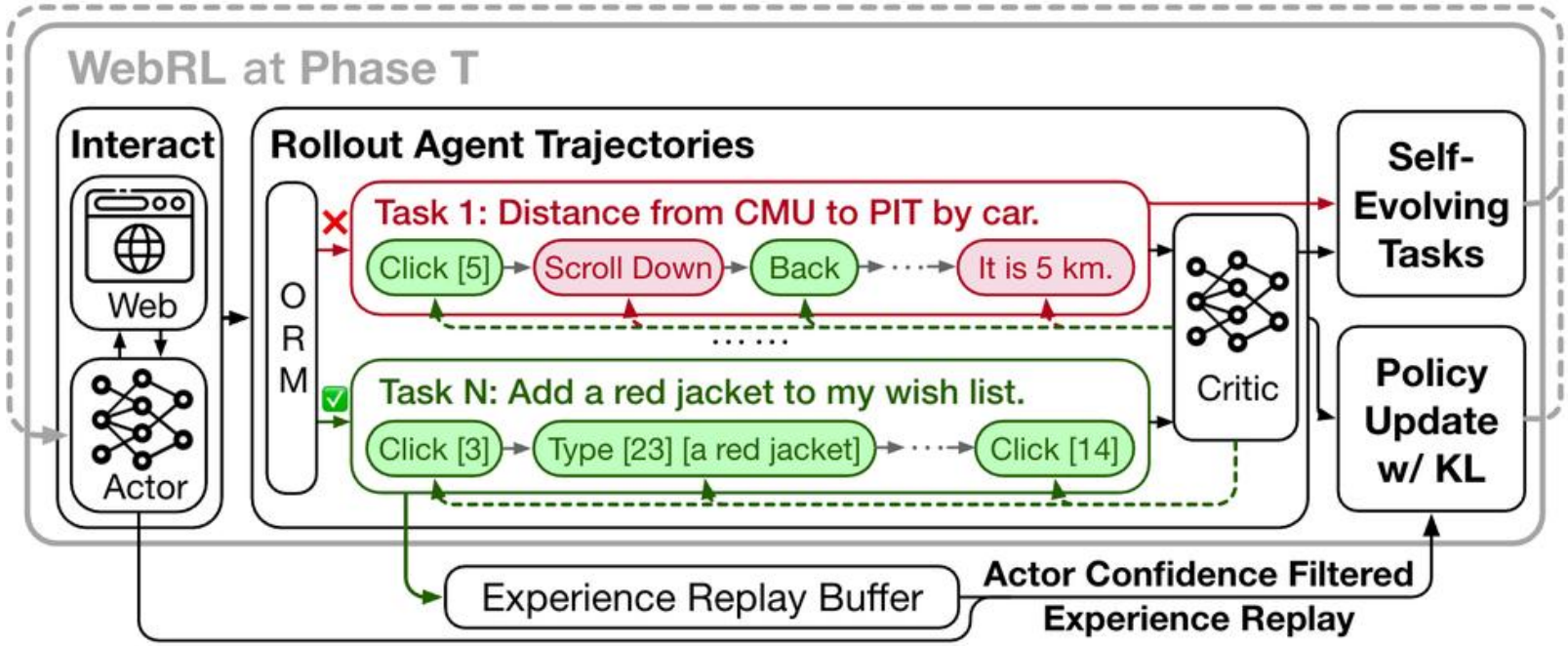
- 大模型有多擅长作为智能体?
- 如何提高大模型作为智能体的水平?
- 如何在在线环境中评估智能体?
- **如何在线环境中持续提升智能体能力?**

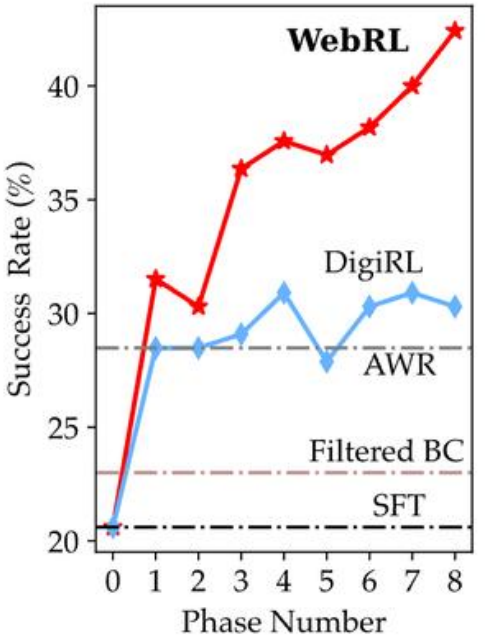
挑战：在线环境中如何持续提升智能体能力？

- **训练任务和数据不足**
  - 在线网页环境通常只提供有限的测试集用于评估。这种缺乏预定义训练任务和训练数据的情况显著阻碍了在这些环境中对智能体的有效训练。
- **反馈信号的稀疏性**
  - 在线网页环境中反馈信号具有严重稀疏性。我们很难有效的找到一个通用的自动化方法，判断一个任务是成功了还是失败了。同时，模型的初始策略通常很差，难以完成任务以获得正反馈信号。
- **在线学习中的策略分布漂移**
  - 由于缺乏预定义的训练集，在线探索成为必要，而这不可避免地导致智能体策略的分布漂移。这一现象可能会引发灾难性遗忘，并导致智能体的能力水平随时间下降。

WebRL：自进化在线课程强化学习方法

- **自我进化**：智能体不断与环境互动，收集实时轨迹数据。这种交互由的课程学习策略引导，动态地根据智能体当前的能力水平量身定制生成任务，从而有效缓解训练任务不足和增加获得正反馈的可能性
- **基于结果的奖励模型**：通过训练一个具有泛化性的奖励模型，来自动来评估任务的成功情况
- **KL 散度约束的策略更新**：防止在课程学习过程中出现剧烈的策略偏移
- **改进的经验重放缓冲区**：保留先前的知识，以降低灾难性遗忘的风险。这些技术使得智能体能够逐步改进，逐步处理更复杂的任务。





WebRL：自我进化

- 随着训练的进行，不会做的任务逐渐变得会做，而且难度逐渐提升

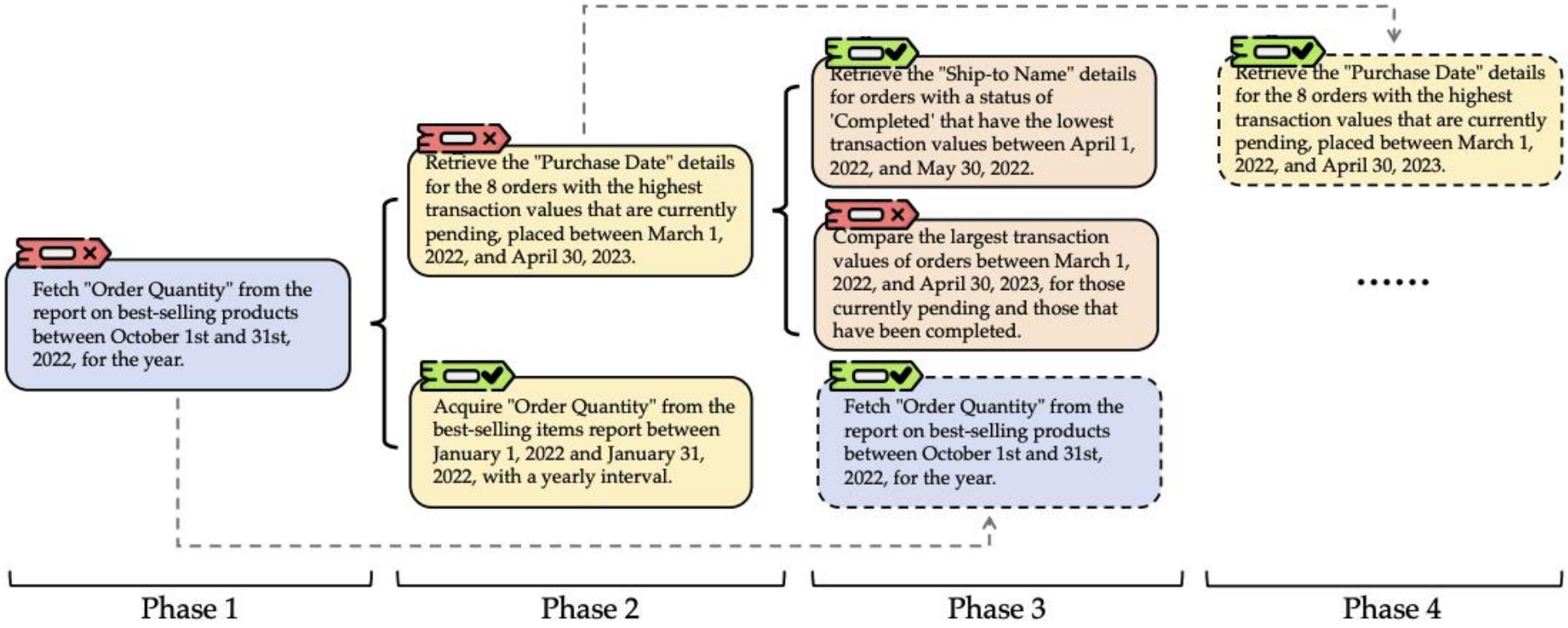


Figure 8: Examples of instructions generated in different phases under self-evolving curriculum learning.

WebRL：结果监督奖励模型

- 在课程学习过程中，我们需要根据代理生成的轨迹来确定是否完成了相应的指令
- 由于缺乏来自环境的反馈，我们训练了一个 LLM 作为结果监督奖励模型 ORM，以自动进行任务成功评估
  - 0 表示失败，1 表示成功

Table 3: Evaluation on output-supervised methods (baselines adopted from (Pan et al., 2024)). Our ORM, without accessing proprietary GPT-4, performs the best among all.

	Our ORM (8B)	GPT-4	Captioner + GPT-4	GPT-4V
Test Dataset (%)	80.8	71.9	72.6	71.2
Rollout (%)	79.4	71.2	73.3	70.5

WebRL：KL 散度约束的策略更新

在课程学习的过程中，所学的知识可能会被遗忘。传统方法通常通过混合来自不同阶段的数据来缓解该问题。然而，在网络代理任务中，中间步骤不会收到直接的过程奖励，只有来自最终状态结果的微弱信号。后面的步骤出现错误很容易导致最终的失败，导致中间步骤的误判，难以重用。

我们设计了一个KL限制的策略更新算法，以更直接地解决策略分布漂移问题，其细节如下：

Actor

$$(1)\quad \max_{\pi_\theta} \mathbb{E}_{I\sim\rho(I),a_t\sim\pi_\theta(\cdot|s_t)}\left[\sum_{t=0}^{T}\left(r(s_t,a_t,I)+\beta\log\pi_{\text{ref}}(a_t|s_t,I)\right)+\beta\mathcal{H}(\pi_\theta)\right]$$

$$(2)\quad \pi^*(a_t|s_t,I)=e^{(Q^*(s_t,a_t,I)-V^*(s_t,I))/\beta}$$

$$(3)\quad Q^*(s_t,a_t,I)=\begin{cases} r(s_t,a_t,I)+\beta\log\pi_{\text{ref}}(a_t|s_t,I)+V^*(s_{t+1},I), & \text{if } s_{t+1} \text{ is not terminal}\\ r(s_t,a_t,I)+\beta\log\pi_{\text{ref}}(a_t|s_t,I), & \text{if } s_{t+1} \text{ is terminal}\end{cases}$$

$$(4)\quad \beta\log\frac{\pi^*(a_t|s_t,I)}{\pi_{\text{ref}}(a_t|s_t,I)}=r(s_t,a_t,I)$$

$$(5)\quad \beta\log\frac{\pi^*(a_t|s_t,I)}{\pi_{\text{ref}}(a_t|s_t,I)}=r(s_t,a_t,I)+V^*(s_{t+1},I)-V^*(s_t)=A^*(s_t,a_t,I)$$

$$(6)\quad \mathcal{L}(\pi_\theta)=\mathbb{E}_\nu\left[\left(\beta\log\frac{\pi_\theta(a|s,I)}{\pi_{\text{ref}}(a|s,I)}-A^*(s,a,I)\right)^2\right]$$

$$(7)\quad \nabla_\theta\mathcal{L}(\pi_\theta)=-2\beta\mathbb{E}_\nu\Big[\underbrace{\left(A^*(s,a,I)-\beta\log\frac{\pi_\theta(a|s,I)}{\pi_{\text{ref}}(a|s,I)}\right)}_{\text{update direction}}\underbrace{\nabla_\theta\log\pi_\theta(a|s,I)}_{\text{sft loss}}\Big]$$

KL divergence constraint

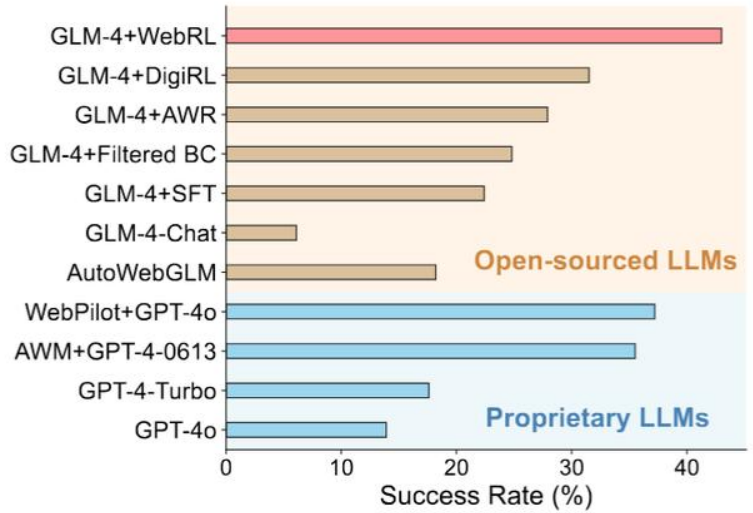
Critic

$$(8)\quad \mathcal{L}(V)=-\mathbb{E}_\nu\Big[r(s_T,a_T,I)\log V(s,a,I)+(1-r(s_T,a_T,I))\log(1-V(s,a,I))\Big]$$

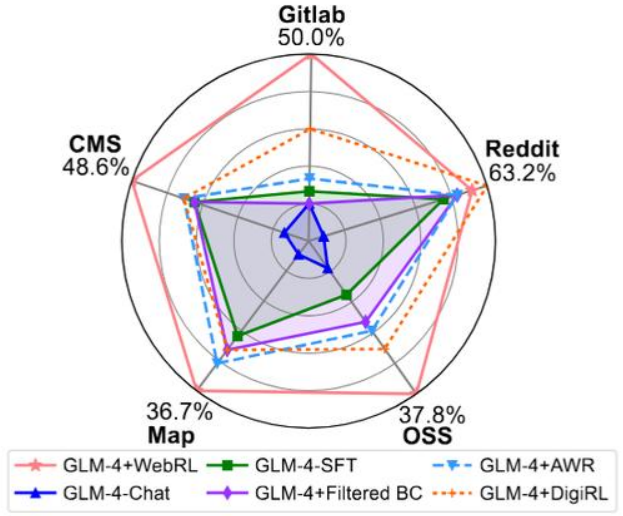
$$(9)\quad A(s_t,a_t,I)=\lambda\big(r(s_t,a_t,I)+V(s_{t+1},I)-V(s_t,I)\big)+(1-\lambda)\big(r(s_T,a_T,I)-V(s_t,I)\big)$$

WebRL：实验结果

- WebRL 在不同模型架构中效果一致，Llama3.1-8B 在 WebRL 训练下，平均准确率达到 42.4%； GLM-4-9B 也展现了相似的性能提升，达到了 43% 的平均准确率，证明了 WebRL 的鲁棒性和适应性。
- WebRL 在更大规模模型上仍具有有效性。Llama3.1-70B 在 WebRL 训练后的平均准确率达到了 49.1%，大大超过 GPT-4 的表现



(a) Performance comparison between proprietary LLMs and open-sourced LLMs on WebArena-Lite.



(b) Performance changes of GLM-4-9B trained with WEBRL and baseline methods.

WebRL：实验结果

- WebRL 与 DigiRL
  - 证明了通过自我进化的课程学习策略可以实现更加持续的性能提升
- WebRL 无经验回放 与 WebRL 无经验回放和 KL 散度控制更新
  - 证明了 KL 散度约束的策略更新算法在减轻知识遗忘上的有效性
- WebRL 与 WebRL 无经验回放
  - 证明了重放缓冲区在实现模型能力稳定提升上的有效性

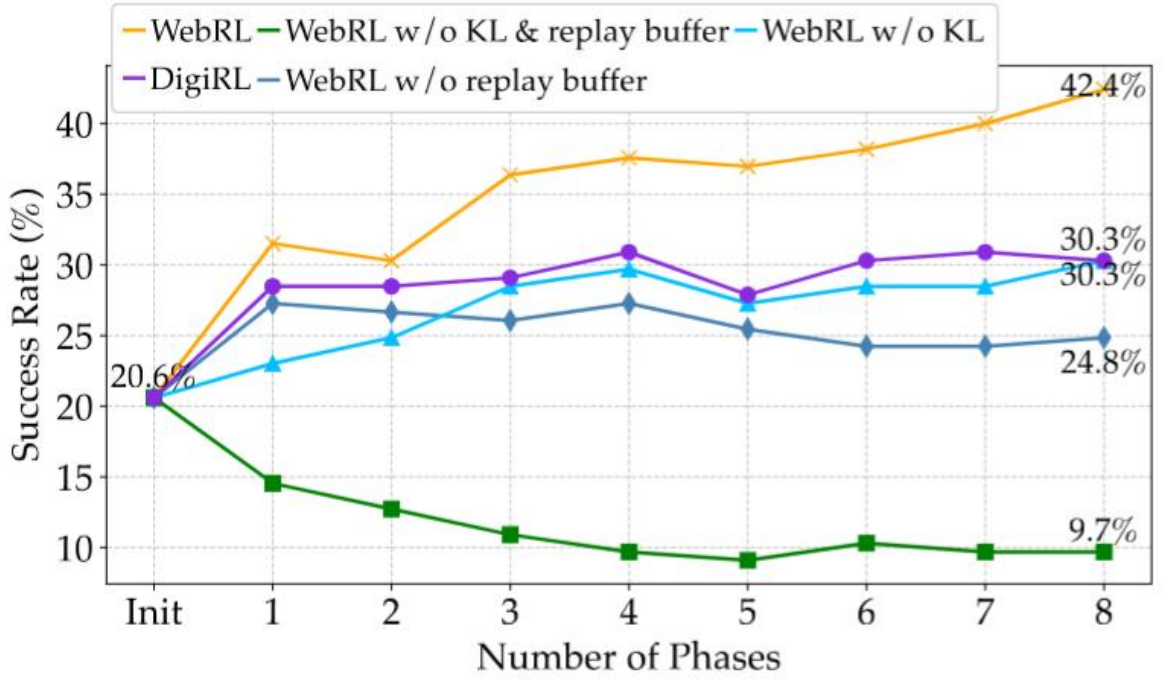


Figure 5: Ablation study of WEBRL on replay buffer, KL-constrained policy update and curriculum strategy.

AutoGLM：面向 GUI 的自主基础智能体

AutoGLM：面向 GUI 的自主基础智能体

开通内测白名单

下载 APK

AutoGLM: More Real-Speed Recording