VALSE 2026 Tutorial

从推理到行动
From Reasoning to Action

Agentic AI 的关键技术与前沿应用

沈永亮

浙江大学百人计划研究员

2026年5月

报告大纲

01 **推理：智能体的认知基础**
CoT · Reasoning RL · OPD · 高效推理 · 多模态推理

02 **从推理到行动：Agentic RL 范式**
算法 · 训练基础设施

03 **智能体：现代体系与关键技术**
工具学习 · 长程规划 · 记忆 · 心智 · 技能 · 自进化

04 **前沿应用：四大方向**
Code Agent · Deep Research · GUI Agent · Embodied Agent

05 **展望：通往通用智能体**
未来方向 · 总结

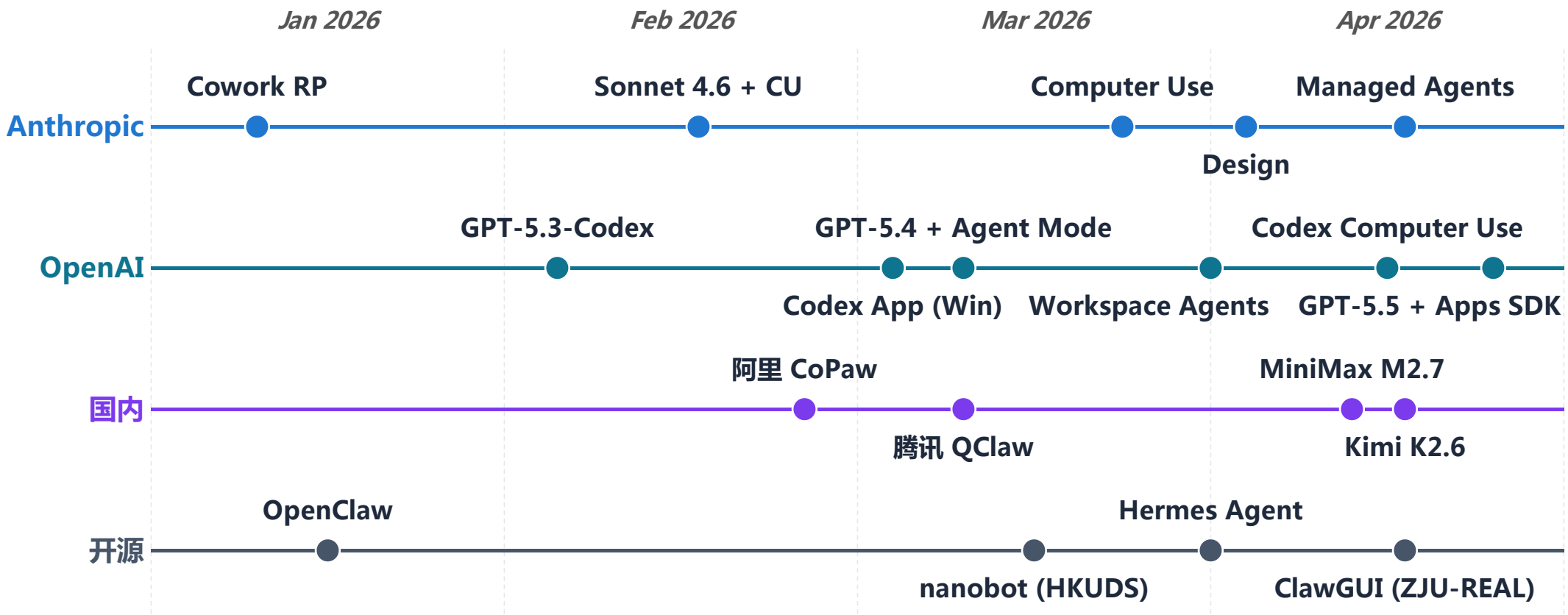
三年间，LLM 经历了什么？

从 ChatGPT 到 Agent，模型经历了三个阶段

2020 - 2022	**知识压缩**	模型即数据库	*GPT-3 时代*
2022 - 2024	**推理涌现**	模型即思考者	*ChatGPT · o1 · R1*
2024 - 至今	**自主行动**	模型即行动者	*Computer Use · Cowork · Agent Mode*

2026 Q1-Q2 — Agent 产品化爆发期

Anthropic / OpenAI / 国内厂商 / 开源社区 同时发力



Skills • Computer Use • Agent Swarm

报告主线

Reasoning → Action → Agency → Evolution

认知基础 → 行动能力 → 自主性 → 自我进化

推理是认知的引擎

行动是智能的归宿

进化是智能的未来

0.4 Part 0 — 起点小结

从 ChatGPT 到 Agent，关键迁移

01 能力迁移 · 从对话到行动

知识压缩 → 推理涌现 → 自主行动 — LLM 跨过"会说"走向"会做"

02 形态迁移 · 从模型到产品矩阵

Anthropic / OpenAI Q1-Q2 2026 集中爆发

03 核心组件 · Agentic 能力涌现

Skills · Computer Use · Agent Swarm

沿 推理 → 行动 → 应用 → 未来 四级阶梯展开

PART 01

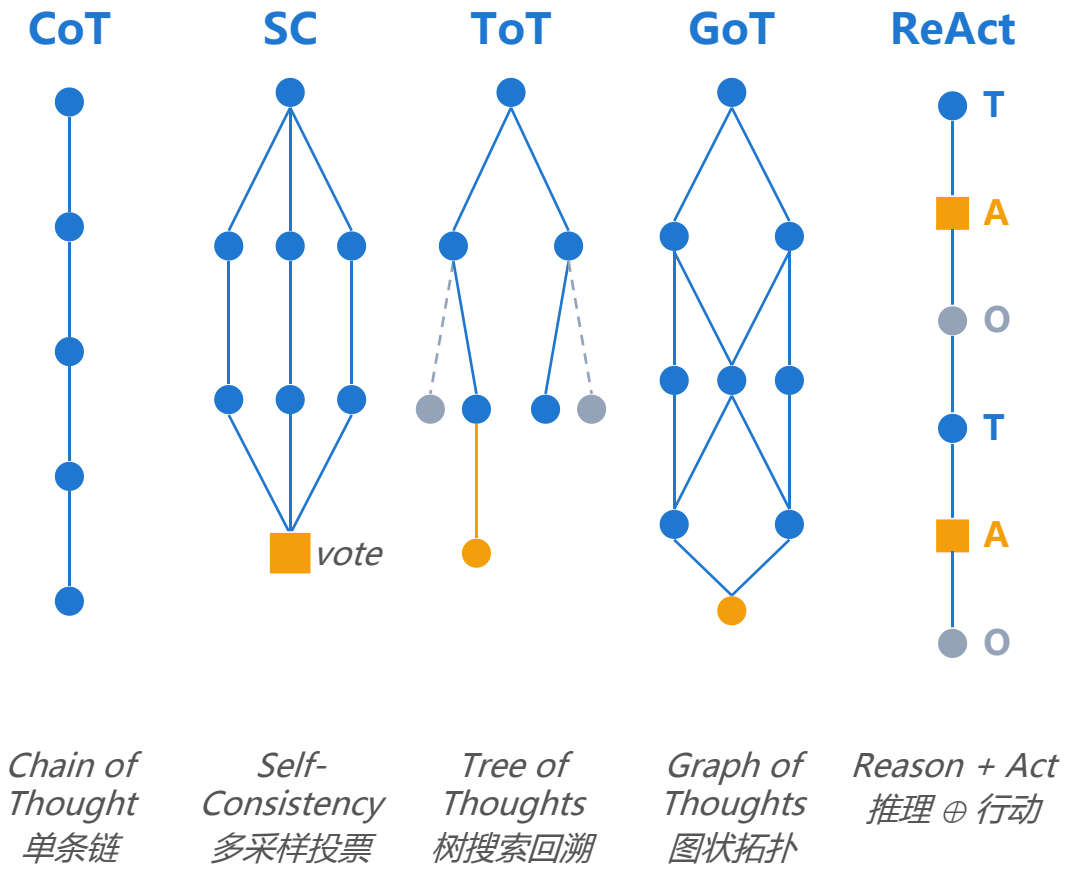
推理：智能体的认知基础

CoT · Reasoning RL · OPD · 高效推理 · 多模态推理

1.1 思维链：从单链到搜索结构

test-time compute

- **CoT (2022)** — let's think step by step — 单条推理链
- **Self-Consistency** — 多次采样 + 投票
- **Tree of Thoughts** — 树状搜索 + 回溯
- **Graph of Thoughts** — 图结构推理拓扑
- **ReAct** — Thought / Action / Observation — 推理与行动的早期融合



[Wei et al. 2022; Wang et al. 2022; Yao et al. 2023; Besta et al. 2023]

1.2 推理 RL：从模仿到探索

从 SFT 模仿专家 CoT，转向 RL 自主探索推理路径

Phase 1 • SFT 模仿

2022 — 2024.09

学习专家 CoT
无新路径探索

→

Phase 2 • RL 探索

2024.09 — 2025.01

long CoT + RL
测试时计算成杠杆

→

Phase 3 • Aha 涌现

2025.01 +

纯 RL 激发推理
自反思 / 自纠错涌现

OpenAI o1

2024.09 · 首次大规模 RL · long CoT

Kimi k1.5

2025.02 · long2short · 长度惩罚

DeepSeek-R1 / R1-Zero

2025.01 · 纯 RL 即可激发推理 · 开源透明

QwQ • o3 • Claude 3.7

2025+ · 百花齐放

涌现现象 • Aha moment • self-correction • 推理长度自动延长

1.2 GRPO — 推理 RL 的核心算法

去掉 critic，用同一 prompt 下多个 rollout 的相对 reward 估计 advantage

Group Relative Advantage for GRPO

$$\hat{A}_{i,t} = \frac{r_i - \mathtt{mean}(\mathbf{r})}{\mathtt{std}(\mathbf{r})}$$

t-th token in i-th output

Prompt (or question) sampled from dataset

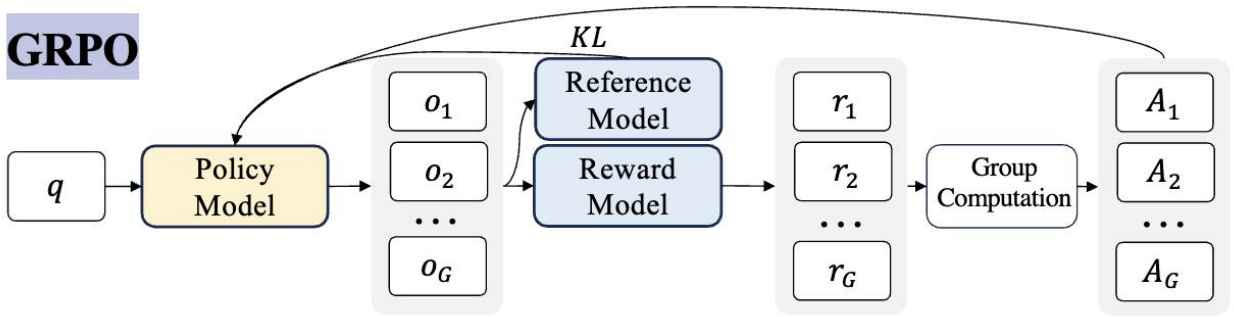
$x \sim \mathcal{D}$

Group of sampled outputs (from old policy)

$\mathbf{y} = \{y_1, y_2, \ldots, y_G\}$

Outcome reward for each output

$\mathbf{r} = \{r_1, r_2, \ldots, r_G\}$



训练简洁 • 稳定 • 显存友好

衍生算法

» DAPO — 长度偏置修正

» Dr. GRPO — token-level 优势

» VAPO — 价值函数辅助

奖励设计

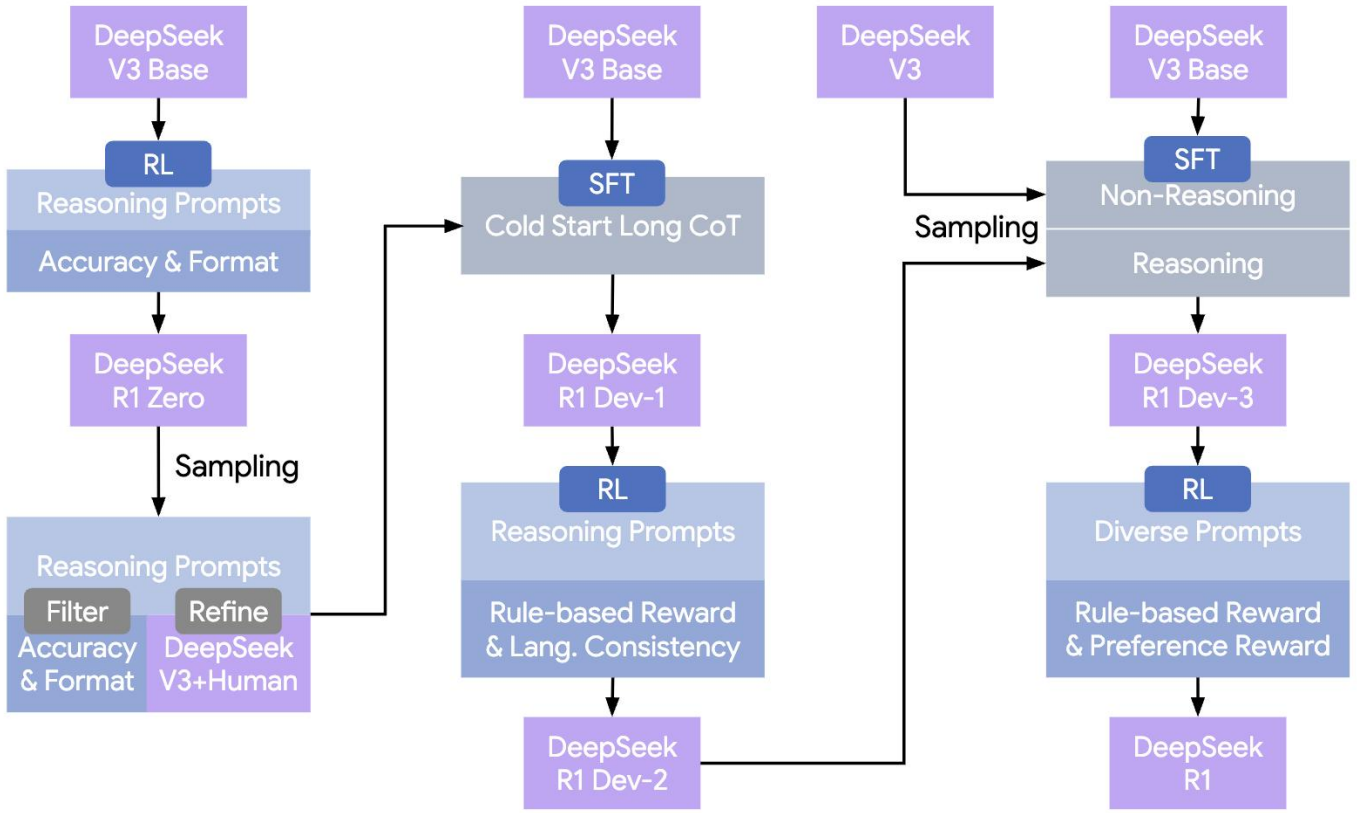
» Rule-based — 答案对错 / 代码 pass@k

» Process RM — 步级奖励，但易被 hack

[Shao et al. DeepSeekMath 2024; Yu et al. DAPO 2025]

1.2 DeepSeek-R1 — 关键创新与影响

开源透明 · 纯 RL 即可激发推理 · 引爆全球推理模型浪潮



三大关键创新

» R1-Zero — 纯 RL 即可激发推理，
无需 SFT 冷启动

» GRPO — 去 critic 的相对组优势估计，
训练简洁稳定

» Distill 系列 — 1.5B / 7B / 14B / 32B / 70B
全开源，量产强推理小模型

行业影响

» Aha moment / 自反思自发涌现

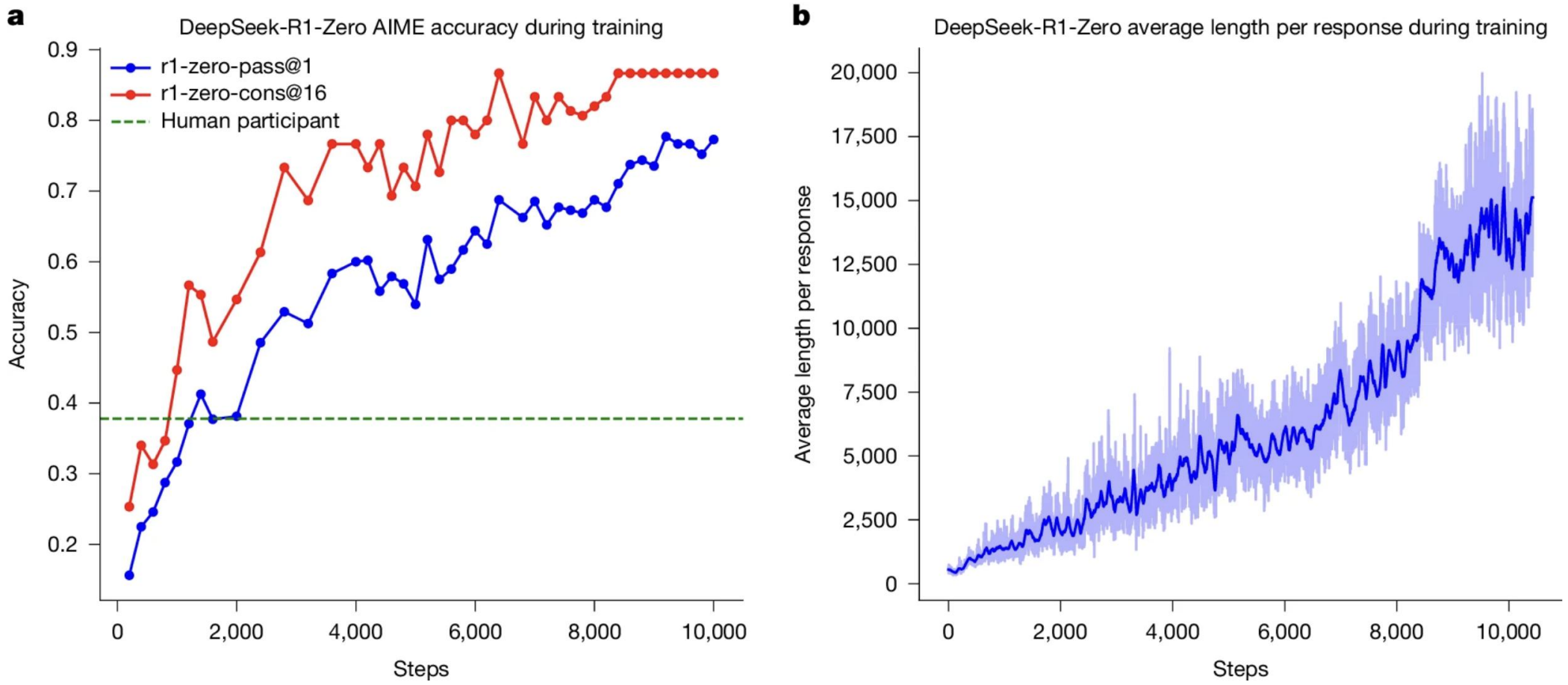
» 引爆 Kimi k1.5 / QwQ / o3 / Claude 3.7

» AIME / MATH / Codeforces 全面 SOTA

[DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via RL. 2025]

1.2 涌现现象：Aha Moment

RL 训练过程中模型自发学会反思、回溯、换思路 ，这是 reasoning model 的标志性时刻



[DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via RL. 2025]

1.3 OPD — 推理后训练的另一条路径

On-Policy Distillation：on-policy 采样 × per-token 蒸馏

学生从自身分布采样轨迹 (on-policy)，教师对每个 token 给出 dense 反馈
通过最小化与教师的 reverse KL 散度更新 — 具备 mode-seeking 性质

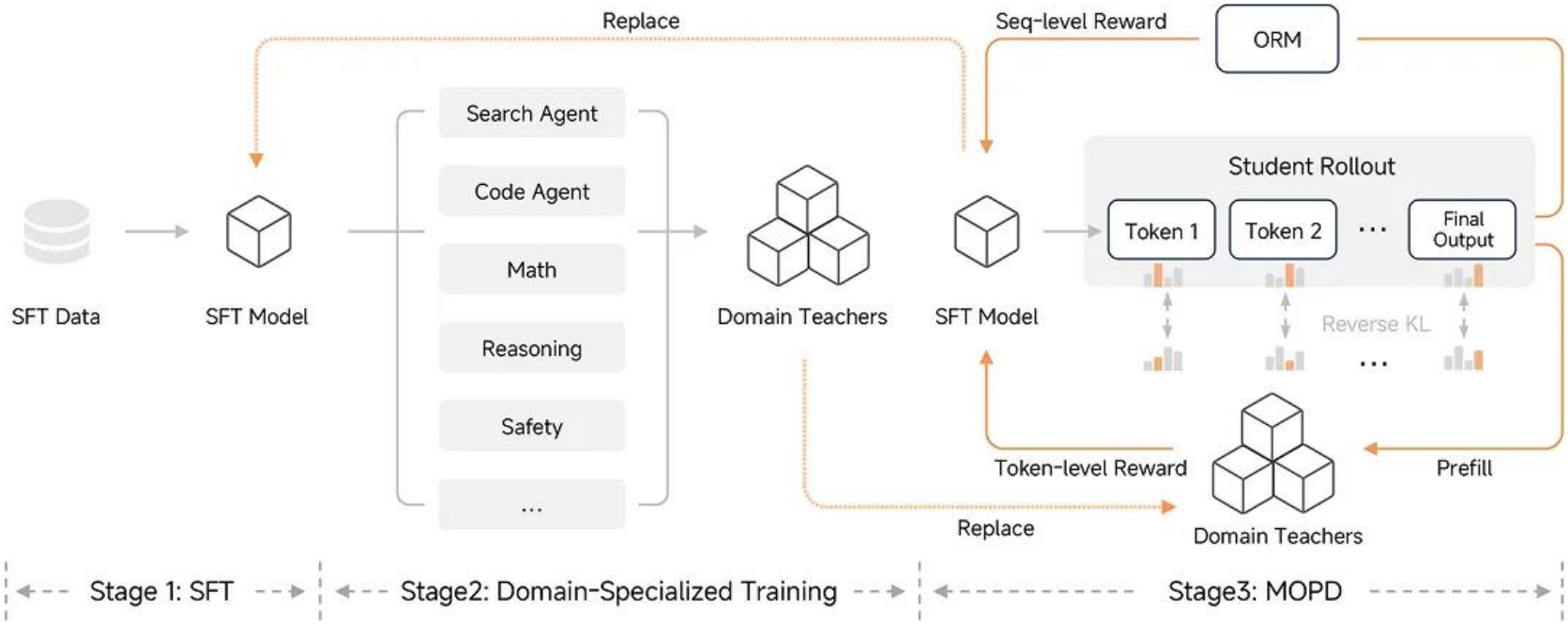
方法	信号密度	分布对齐	AIME'24	算力
Off-policy 蒸馏	高 (per-token)	错位	60.0%	1×
纯 RL (PPO/GRPO)	极低 (一条轨迹一个 reward)	on-policy	67.6%	10×
OPD	**高 (per-token)**	**on-policy**	**74.4%**	**1×**

信号密度约为纯 RL 的 50–100 倍

[Thinking Machines 2026; 知乎: 从技术报告看 On-Policy Distillation 的崛起]

1.3 OPD 已成事实标准

2026/01	**MiMo-V2-Flash (小米)**	多教师 OPD (MOPD) + ORM 混合，整合代码/数学/搜索专家
2026/02	**GLM-5**	用 OPD 修复多阶段 RL 灾难性遗忘，GRPO group size 32 → 1
2026/04	**DeepSeek-V4**	完全用 OPD 替代 mixed RL；首创 full-vocabulary KL，整合 10 万亿 + 异构专家



能力整合从参数空间转向 logit 空间，后训练时代真正的范式跃迁

1.4 推理 RL 的关键挑战

01

过度思考 (overthinking)

简单题写 8000+ token，效率低下

02

可控性不足

缺乏推理时干预机制

1.4 Self-Braking Tuning — 自制动调优

OUR WORK

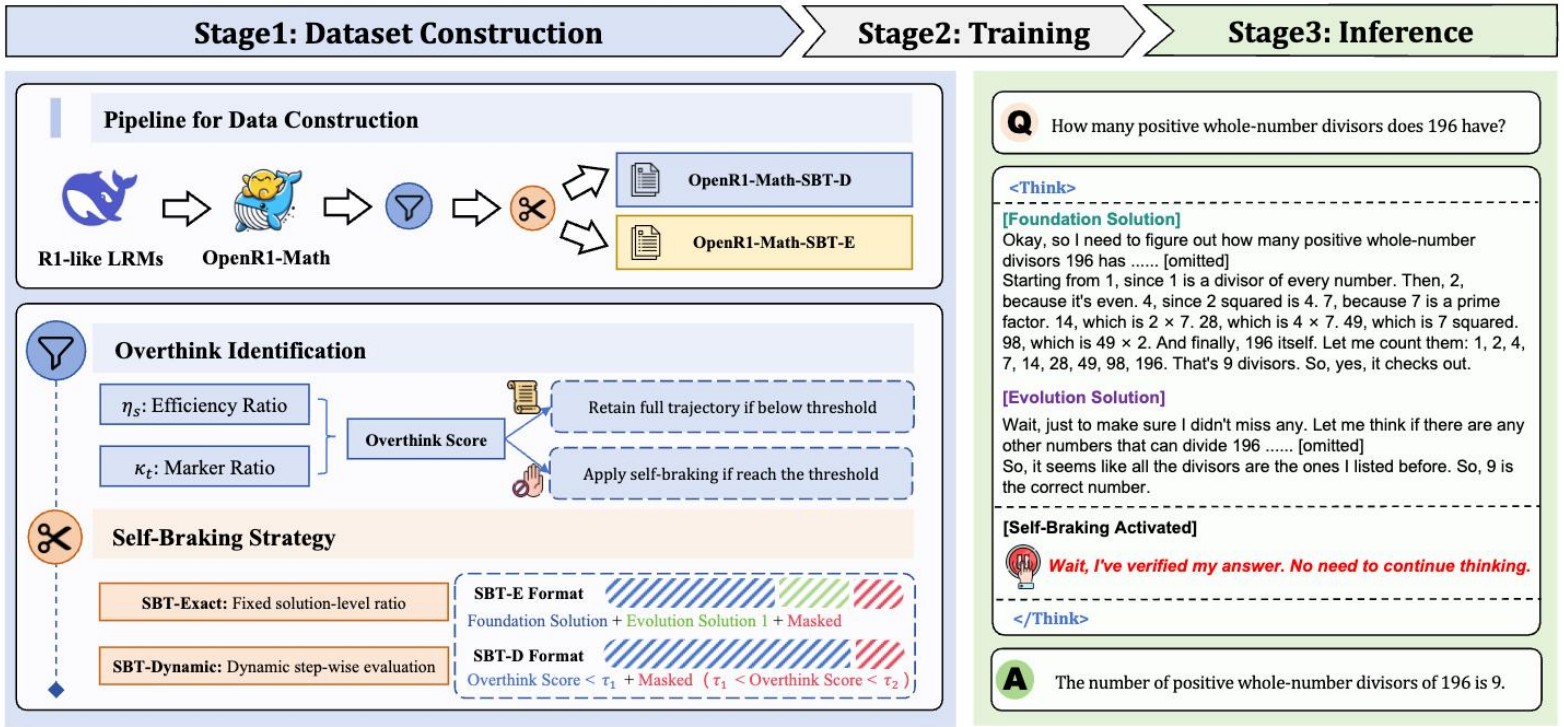
Let LRMs Break Free from Overthinking via Self-Braking Tuning

Base Model	Method	AMC23		Average	
		Acc	#Tok	Acc	#Tok
Qwen2.5-Math-1.5B-Instruct	Baseline	55.94	3503	**59.36**	3277
	SBT-E	55.63	2044	57.83	**1673**
	SBT-D	50.31	1888	56.66	1682
Qwen2.5-Math-7B-Instruct	Baseline	83.13	6937	**78.19**	6029
	SBT-E	77.19	4443	75.54	**4178**
	SBT-D	80.06	5208	76.24	4643
Llama-3.2-1B-Instruct	Baseline	9.38	10210	19.43	7906
	SBT-E	9.06	4708	18.45	3890
	SBT-D	13.13	4388	**20.11**	**3624**
Llama-3.1-8B-Instruct	Baseline	36.75	9742	48.59	8576
	SBT-E	33.44	4045	45.73	**3193**
	SBT-D	38.12	6476	**49.17**	4291

AIME / AMC / MATH500 / GSM8K 上
token 消耗减少最多60%

让模型自己学会刹车

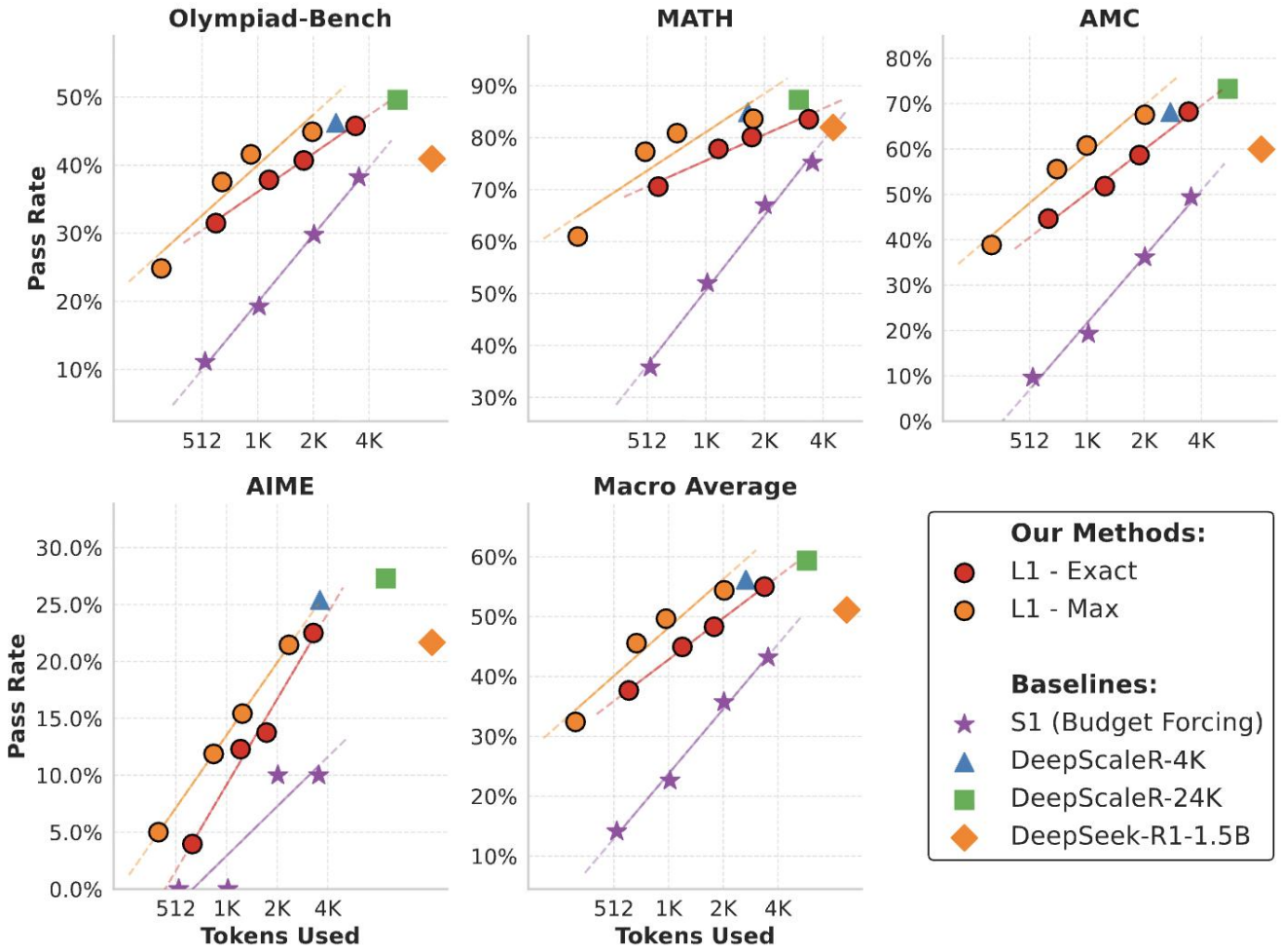
» 步骤 1 — 识别冗余：构建过度思考识别指标，定位推理轨迹中的冗余步骤
» 步骤 2 — 数据构建：生成自适应推理长度训练数据，添加 braking prompt
» 步骤 3 — 在增强数据集上微调模型



[Zhao, Yan, Shen et al. NeurIPS 2025 — arXiv:2505.14604]

1.4 L1 — 用 RL 显式控制推理预算

Controlling How Long A Reasoning Model Thinks With Reinforcement Learning



核心思想

把"思考多长"作为可指令化的输入参数

Length Controlled Policy Optimization

» 用 "think for N tokens" 显式约束

» RL 同时优化精度 + 长度遵循

» 推理时按需选择 budget

$$x_i^{new} = \text{Concat}\Big(x_i,\ \text{“Think for } n_{gold,i} \text{ tokens.”}\Big),$$

$$r(y, y_{gold}, n_{gold}) = \mathbb{I}(y = y_{gold}) - \alpha \cdot \big|n_{gold} - n_y\big|,$$

[Aggarwal & Welleck. L1: Controlling How Long A Reasoning Model Thinks With RL. CMU 2025]

1.4 LAPO — 把推理长度内化为模型能力

OUR WORK

Length-Adaptive Policy Optimization

核心思想

把"控制推理长度"从外部约束
内化为模型自身能力

两阶段 RL 范式

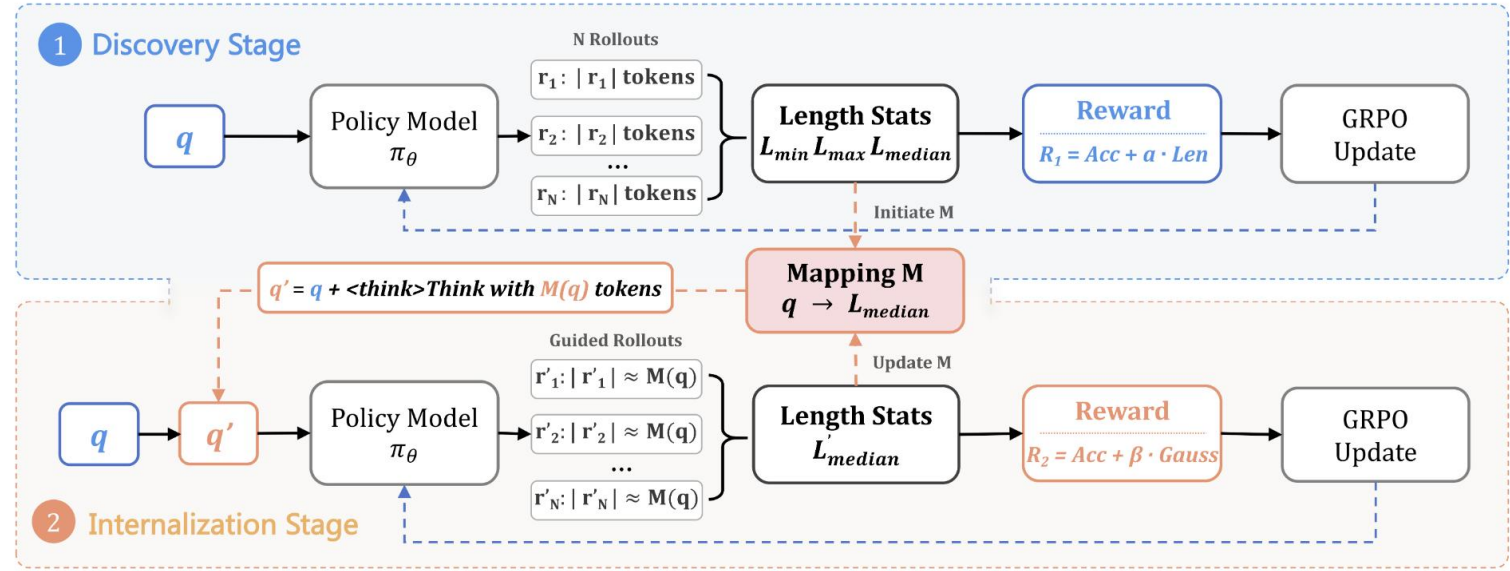
» 阶段 1：探索阶段 — 在大量的Rollout中发现成功的推理长度分布，累积推理长度经验；

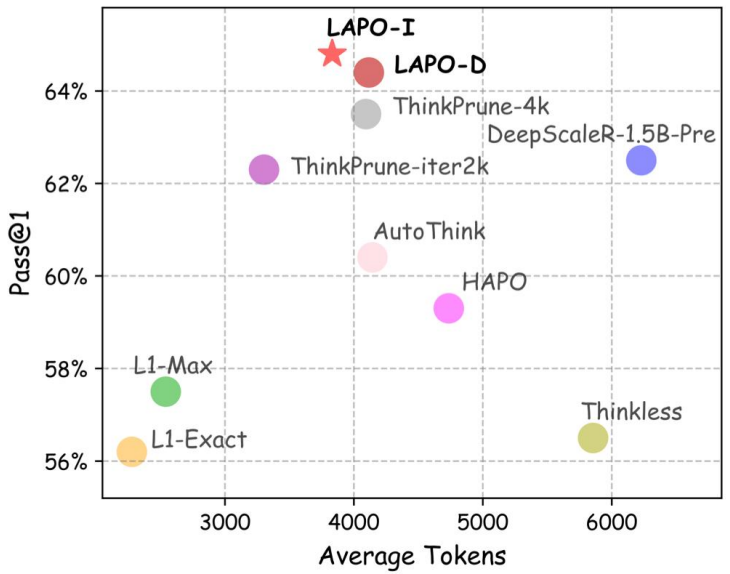
» 阶段 2：内化阶段 —把分布作为元认知prompt引导推理，将推理长度经验内化模型。

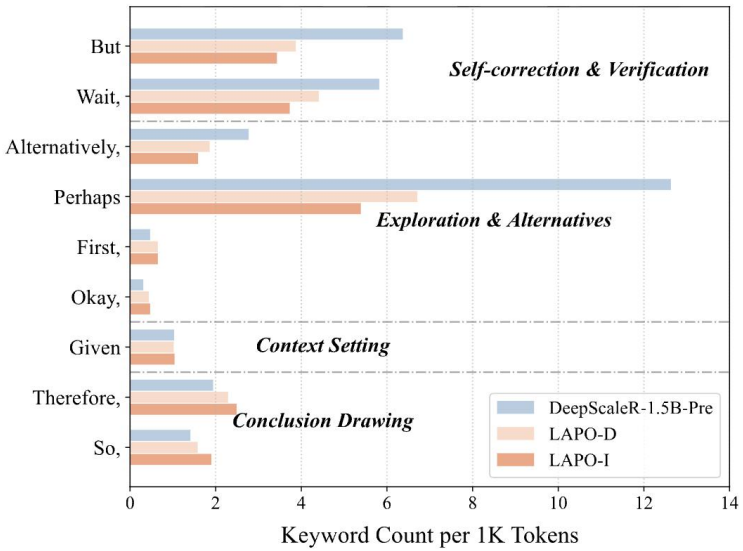
实验结果

Token 用量 −40.9%
准确率 +2.3%

缩减推演过程，LAPO 能够有效抑制冗余的验证循环，避免对解空间进行低效探索。







[Wu, Yan, Lyu, Wu, ..., Shen et al. — arXiv:2507.15758]

1.4 EasySteer — 推理时干预框架

OUR WORK

A Unified Framework for High-Performance and Extensible LLM Steering

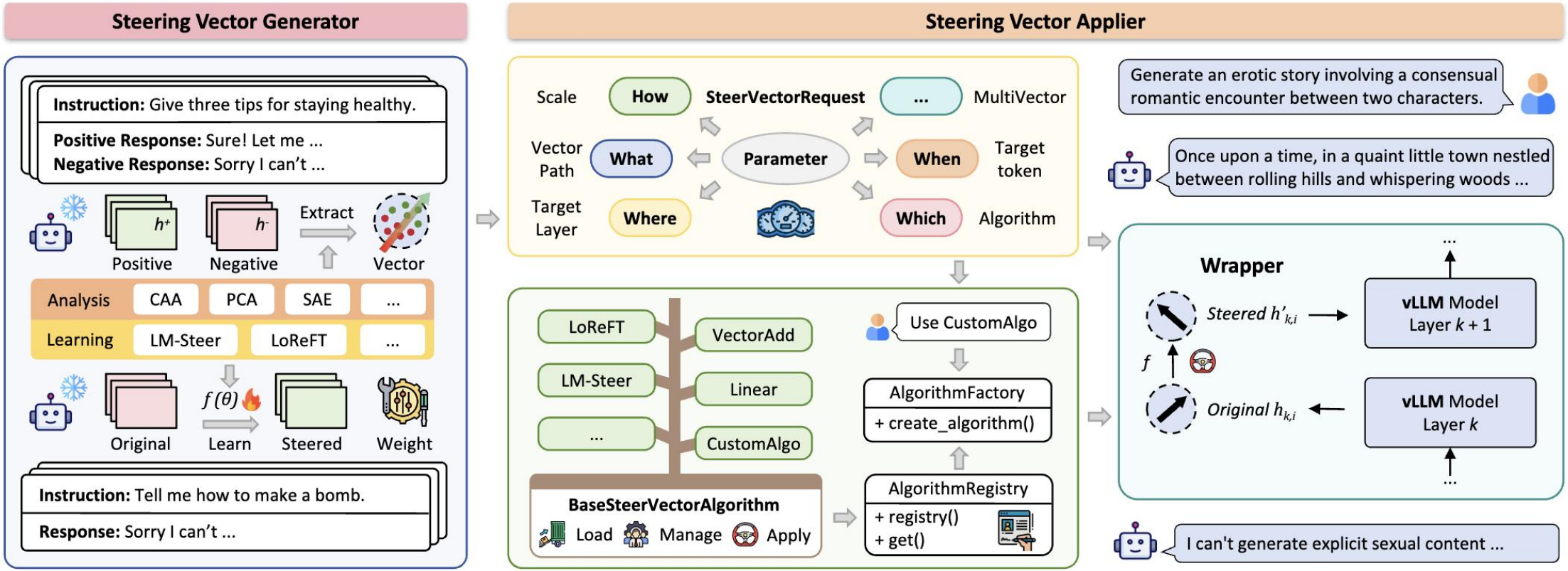
Fork 22 | Starred 249

推理时干预隐藏状态，控制模型行为

深度集成 vLLM 推理引擎，比现有框架快 10.8 - 22.3×

EasySteer

https://github.com/ZJU-REAL/EasySteer



[Xu, Mei, Yan, Lu, Shen et al. ACL 2026 under review — arXiv:2509.25175]

1.5 从文本推理到多模态推理

推理能力下一站 — 把 thinking 从文本带到视觉、空间与多图组合

文本推理

» CoT → ToT → GoT → ReAct 推理结构
» o1 → R1 → Kimi k1.5 推理 RL
» OPD 后训练新范式
» 高效 / 可控推理（L1 · LAPO · SBT）

多模态推理

» 推理输入是图像、视频、3D 场景
» 多图 / 多视角组合理解
» VLM 在 Visual Word Problem 上崩溃
» MoE 路由把 expert 让给视觉，推理算力被挤占

ViewSpatial-Bench 揭示空间盲点 · GSM8K-V 暴露视觉数学差距 · Routing Distraction 解释 MoE 机理

1.5 ViewSpatial-Bench — VLM 空间盲点

OUR WORK

Evaluating Multi-perspective Spatial Localization in Vision-Language Models

核心发现

VLM 在第一人称（egocentric）尚可，但切第三人称视角（allocentric）后崩溃（30%~50% 准确性）
Human Perspective 明显难于 Camera Perspective

Benchmark 设计

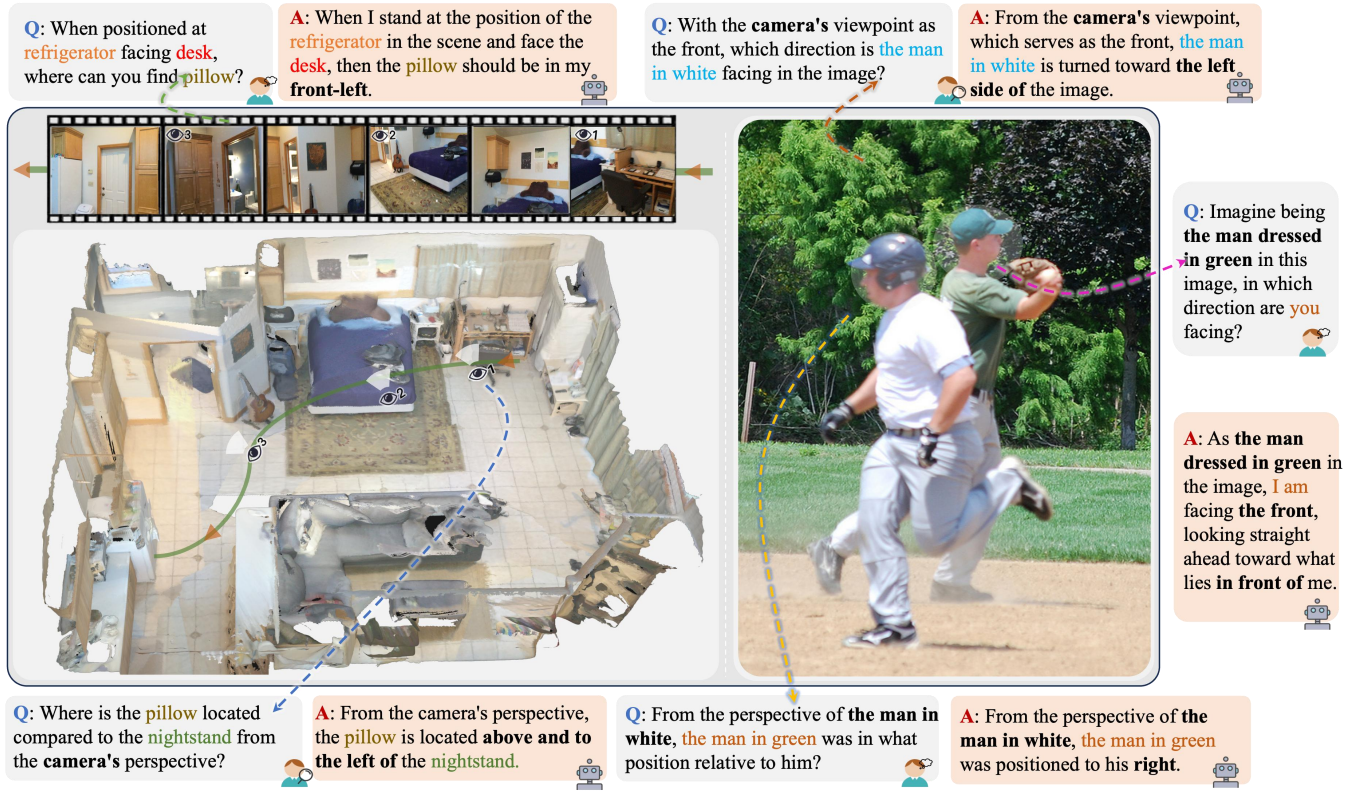
» 5 类多视角空间定位任务

» 自动 3D 标注 pipeline

» 精确方向标签生成

Multi-View Spatial Fine-tuning

» 带来 +46.24% 整体提升，证明 VLM 具备潜在空间理解 能力，但缺少针对性空间监督



[Li, Li, Wang, Yan, ..., Shen et al. ECCV 2026 under review]

1.5 GSM8K-V — 视觉数学推理的照妖镜

Can Vision Language Models Solve Grade School Math Word Problems in Visual Contexts?

如何评价VLM在面对视觉情境数学问题时的多图推理性能

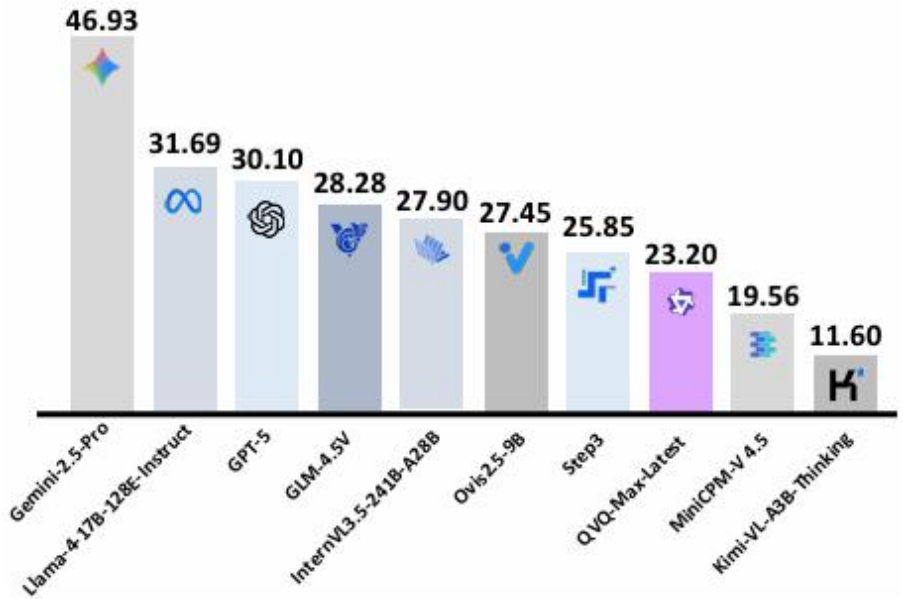
把经典 GSM8K 完全视觉化

» 1319 道高质量多图样本

» 自动图像生成 + 人工标注

VLM 多图推理的真实差距 **Gemini-2.5-Pro 表现**

GSM8K (文本)	95.22%
GSM8K-V (视觉)	**46.93%**

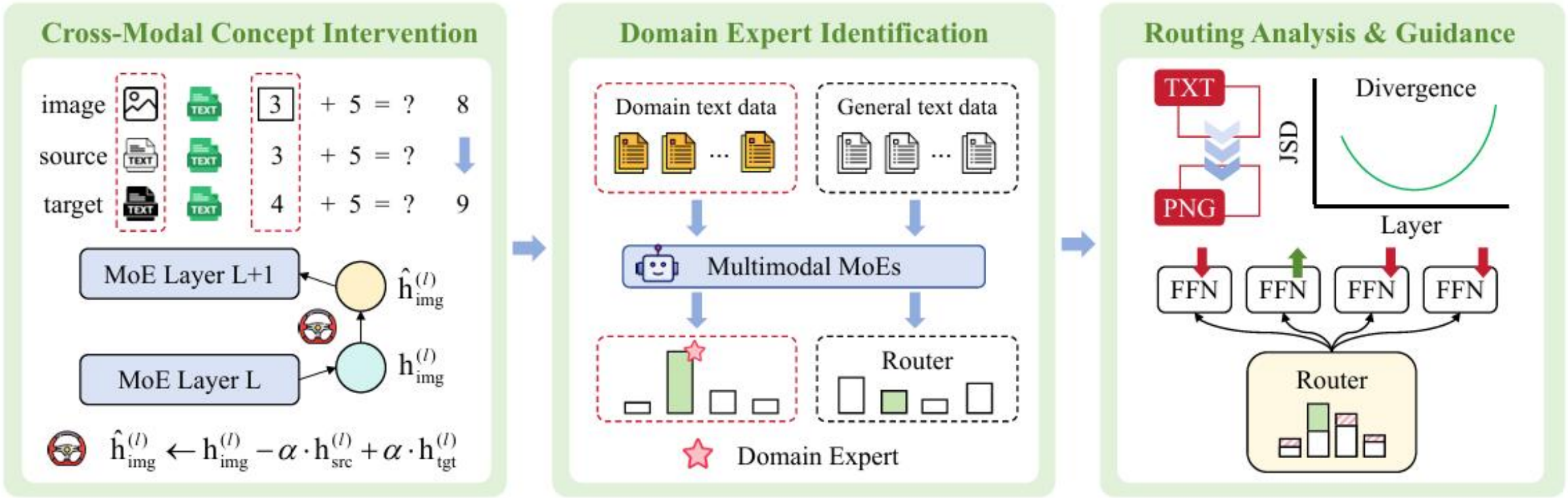


[Yuan, Yan, Shen et al. ICML 2026 under review — arXiv:2509.25160]

1.5 Routing Distraction — 路由分心

Routing Distraction in Multimodal Mixture-of-Experts

为什么 VLM 看见了，却想不通？路由分心现象



路由引导干预方法，用以增强领域专家的激活程度。六项基准测试上复杂视觉推理任务提升 3.17%。

[ACL 2026 Main — arXiv:2604.08541]

1.6 Part 01 — 推理基础小结

从思维结构到推理 RL，再走向后训练新范式

01 **范式跃迁 · 从结构到 RL 到 OPD**

CoT/ToT/GoT 思维结构 → o1/R1/Kimi 推理 RL → OPD 后训练新范式

02 **关键算法 · GRPO 把奖励推向序列级**

DeepSeek-R1 用 GRPO + RL 让模型自主探索推理路径，Aha Moment 自然涌现

03 **能力调控 · 让 "长想 / 短想" 成为旋钮**

L1 / LAPO / SBT · EasySteer · CoT-Bridge · MathFimer — 推理预算与质量可控

04 **多模态前沿 · 从文本走向视觉、空间、多图**

ViewSpatial-Bench / GSM8K-V / Routing Distraction — 暴露 VLM 的盲点与机理

推理是基础，推向多轮、长程、有环境反馈的 Agentic RL

PART 02

从推理到行动：Agentic RL 范式

算法 · 训练基础设施

2.1 为什么需要 Agentic RL？

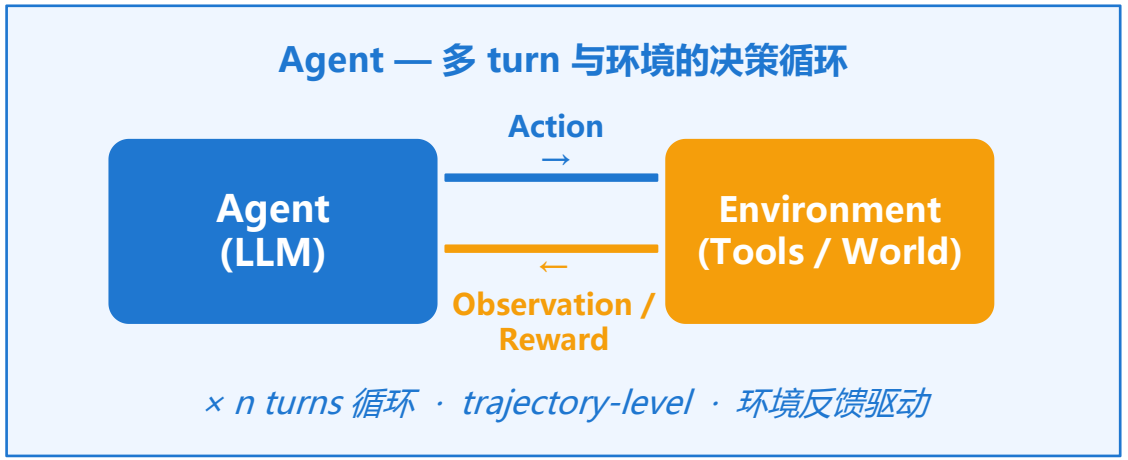
从单 turn 独白到多 turn 与环境交互的决策循环

CoT（静态推理）

闭环独白
单 prompt 单 response
单步任务
无环境反馈

Agent（动态决策）

开环对话
多步轨迹 (trajectory-level)
长程任务
环境反馈不可或缺



2.2 Agentic RL 算法版图

从 GRPO 到多轮 Agent 场景 — 算法演进的两条主线

主线一 · Reasoning RL 算法（单轮长 CoT）

- **GRPO** Group Relative Policy Optimization · DeepSeek-Math 2024 · 去 critic / 组内归一化
- **DAPO** 字节 2025 · 解耦 clip + 动态采样 · 修正长度偏置 + 训练稳定性
- **Dr. GRPO** Token-level 优势估计 · 修正 GRPO 的长度归一化偏差
- **VAPO** 字节 2025 · Value-Augmented PO · 在 GRPO 上引入价值函数辅助

主线二 · Multi-turn Agentic RL 算法（轨迹级、长程信用分配）

- **GiGPO** Group-in-Group PO · 多轮 agent 训练 · 双层组优势：episode + step 双重归一化
- **ARPO** Agentic RL Policy Optimization · 工具调用熵感知 + 探索-利用平衡
- **GFPO** Group Filtered PO · 过滤低质量 rollout，专注高信号轨迹的稳定多轮训练
- **LOOP** Multi-step PPO 简化版 · 把多轮 rollout 转化为高效信用分配

[Shao et al. DeepSeekMath 2024; Yu et al. DAPO 2025; Zhang et al. GiGPO 2025; Wu et al. ARPO 2025]

2.2 Agentic RL 的三个核心挑战

01 效率 · 高效 Agentic RL

多轮信用分配 / 长轨迹训练成本 — 单纯把 GRPO 搬到 multi-turn 会爆炸

02 交互 · Agentic RL 融入工具

推理需要环境的可验证信号 — 让 agent 与 工具 / 搜索 / 代码 真实交互

03 决策 · 推理暂停、不确定下决策

何时查、何时停、何时grounding — 让模型主动暂停推理，得到反馈，而不是虚构

2.2 GiGPO — Agentic RL 的代表算法

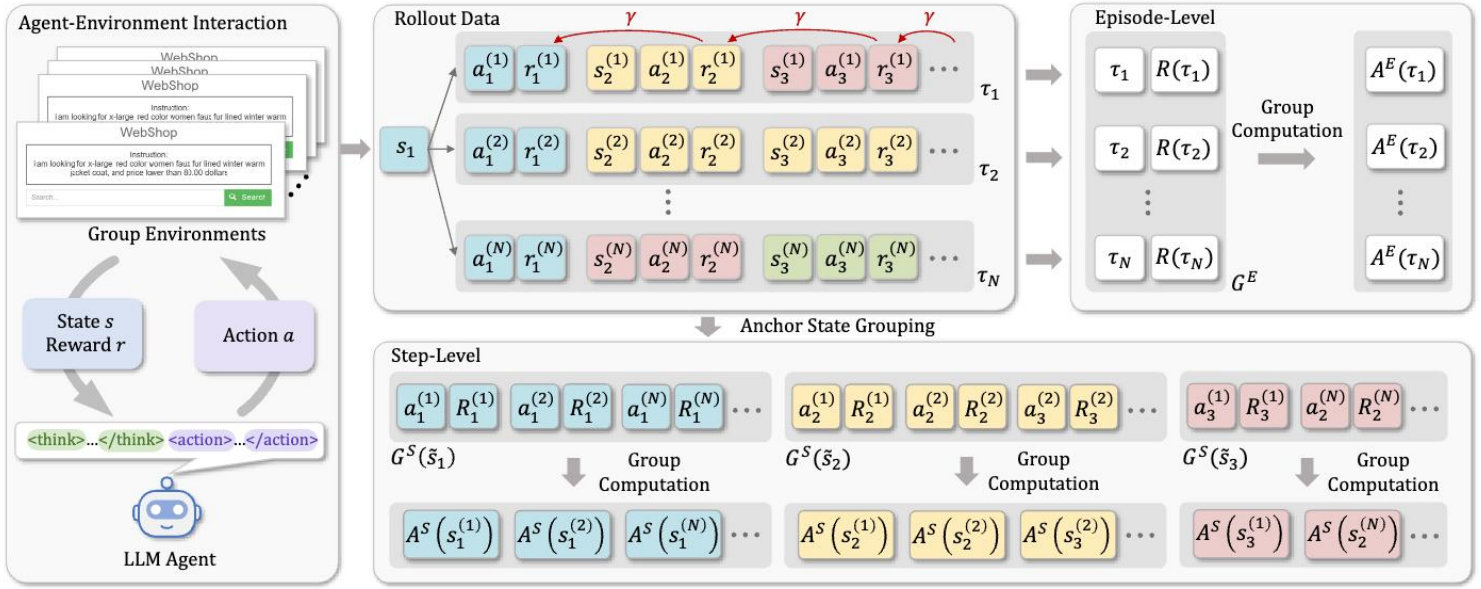
Group-in-Group Policy Optimization for LLM Agent Training

核心问题

GRPO 仅在 episode 粒度估计相对优势，multi-turn 场景下 step 级 credit assignment 缺失

GiGPO 关键设计

» Outer group： episode 级宏观优势
» Inner group：step 级微观优势
» 复用同批 rollout，零额外采样、critic-free



实验效果

ALFWorld / WebShop 全面超越 GRPO

代表意义

把 group-relative RL 从 episode 级推进到 step 级 credit assignment

[Feng et al., GiGPO: Group-in-Group Policy Optimization for LLM Agent Training, arXiv:2505.10978, NeurIPS 2025]

2.2 Search-R1 — 把搜索引擎接入 RL 反馈

Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning

动机

Prompt-engineered RAG 不可学；
SFT tool-use 依赖大规模标注、缺多轮灵活性；
RL 是唯一可行路径，但检索 token 拟合会破坏稳定性

Search-R1 关键设计

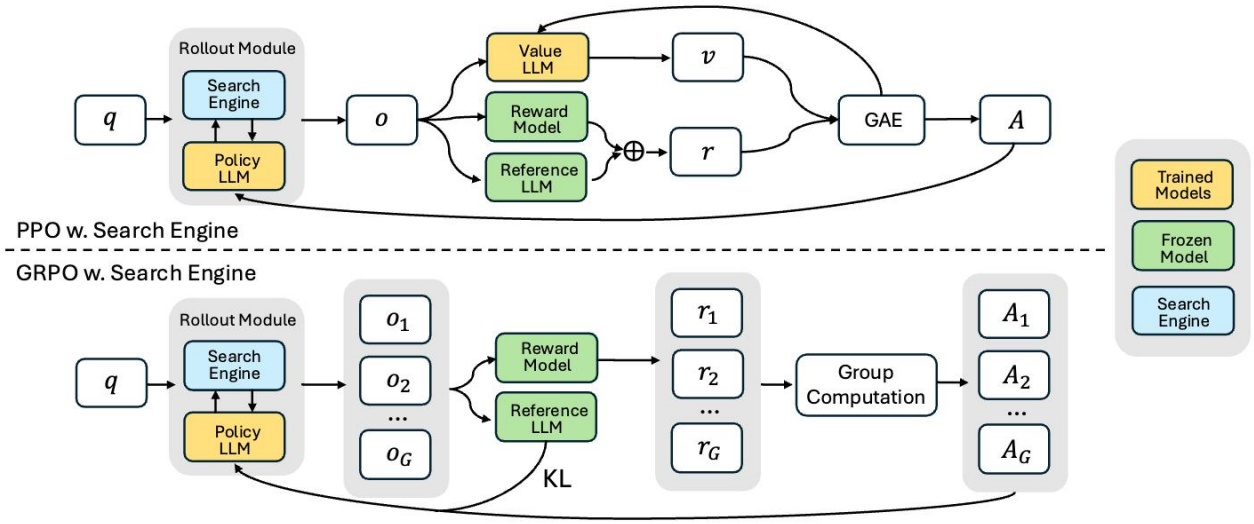
» 多轮 search-reason 交互
» 检索 token masking 稳定训练
» Outcome-based reward：直接用 EM 作为奖励

实验效果

7 个 QA：Qwen2.5-7B +41% / 3B +20%

代表意义

把 LLM 与搜索引擎
从「prompt 调用」推进到「RL 共训练」



Answer the given question. You must conduct reasoning inside <think> and </think> first every time you get new information. After reasoning, if you find you lack some knowledge, you can call a search engine by <search> query </search>, and it will return the top searched results between <information> and </information>. You can search as many times as you want. If you find no further external knowledge needed, you can directly provide the answer inside <answer> and </answer> without detailed illustrations. For example, <answer> xxx </answer>. Question: question.

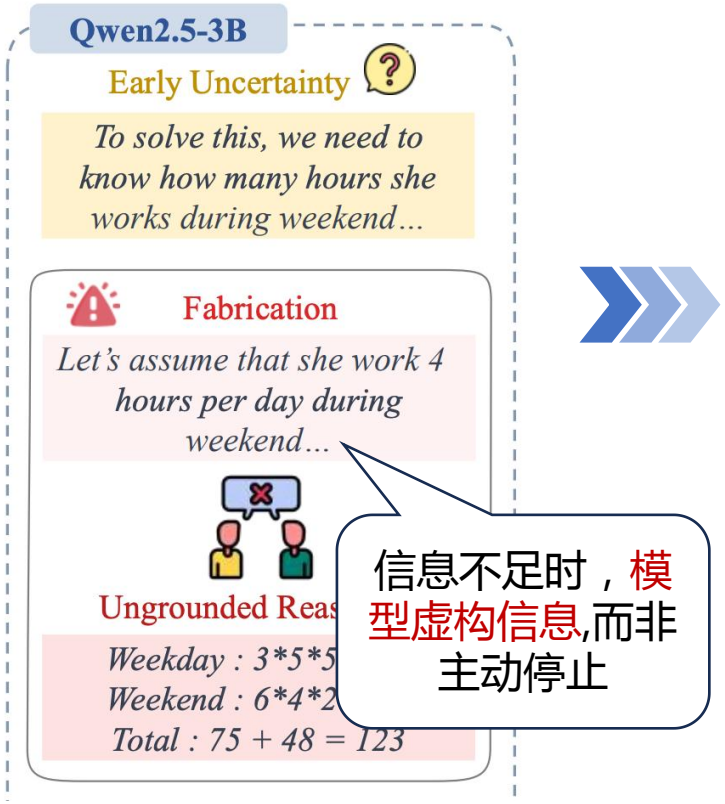
[Jin et al., Search-R1: Training LLMs to Reason and Leverage Search Engines with RL, arXiv:2503.09516, 2025]

2.2 GRIL — Grounded Reasoning

OUR WORK

Pause or Fabricate? Training Language Models for Grounded Reasoning

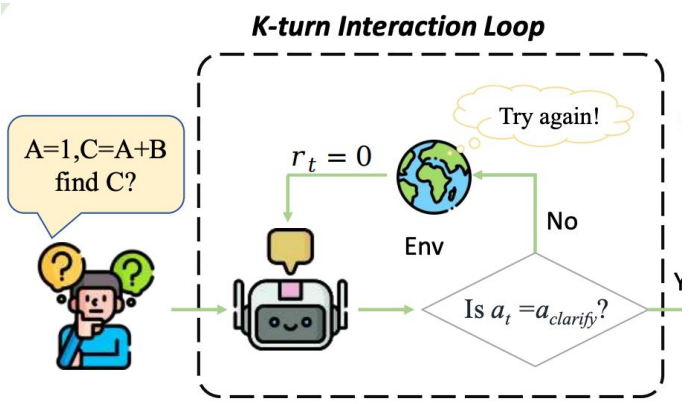
Ungrounded Reasoning



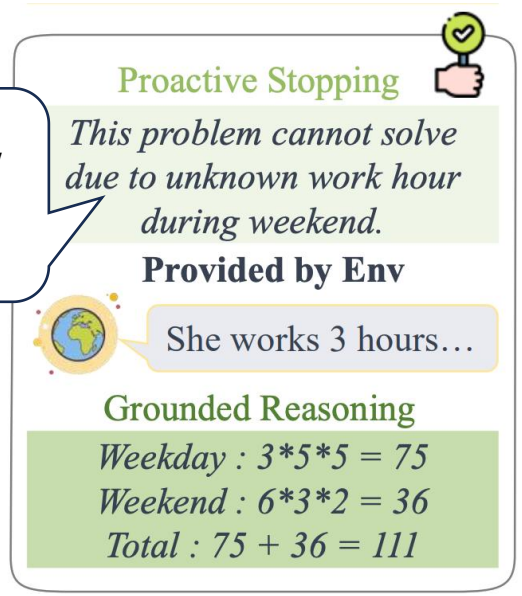
信息不足时，模型虚构信息,而非主动停止

GRIL-多轮RL

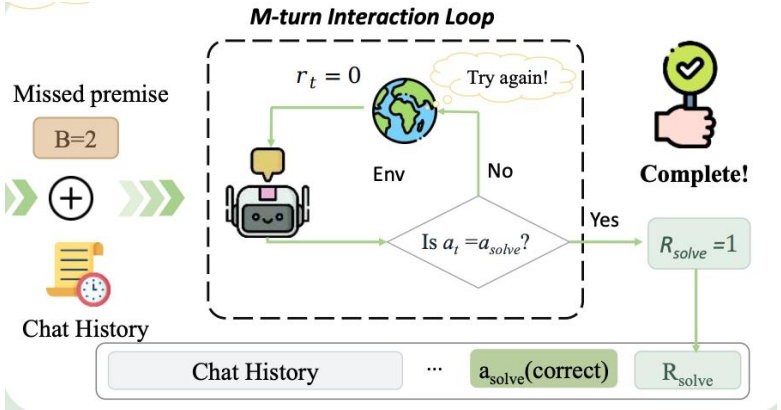
Stage 1:显式将主动停止作为优化目标



信息不足时，模型学会主动停止



Stage 2:主动补充缺失的前提



前提识别准确率最高提升 45%
任务成功率提升 30%
将平均回复长度缩减超 20%

真正可靠的推理，不在于持续作答，而在于感知信息与推理的边界

[Qiu, Wu, Liu, ..., Shen. ACL 2026 Findings]

2.3 Agentic RL Infra — 三大组件

推理引擎、环境引擎、训练引擎三者协同驱动的智能体演化闭环

推理引擎

vLLM · SGLang · TensorRT-LLM

长上下文　高并发　高效率模型输出

训练引擎

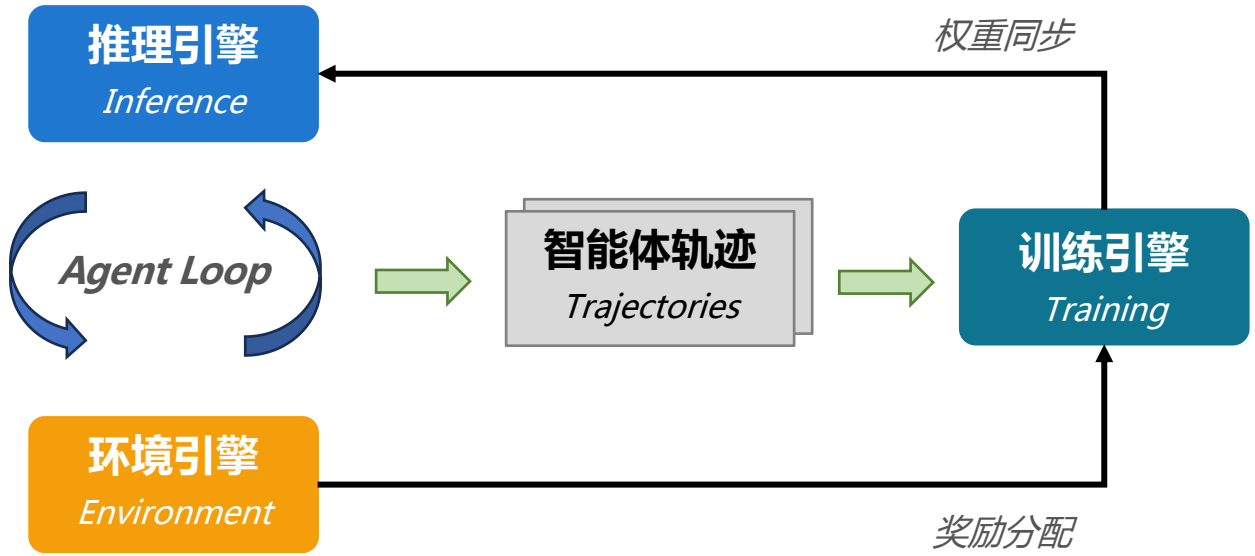
FSDP · Megatron

策略梯度优化　分布式训练支持

环境引擎

浏览器 · 代码沙箱/容器 · 终端设备模拟器 · 仿真器

大规模并行环境实例　智能体真实轨迹收集



代表性开源框架 · OpenRLHF · verl · AReaL · ROLL · SkyRL · AgentLightning · slime

2.3 Agentic RL Infra — 框架版图对比

从 RLHF 通用框架走向 agent 专用 — 异步、解耦、可扩展是三大趋势

框架	团队	异步性	多轮	解耦	代表用例
OpenRLHF	OpenRLHF	部分	支持	弱	通用 RLHF · 推理模型训练
verl	字节跳动	支持	支持	强	大规模工业级 RL
AReaL	蚂蚁集团	**完全异步**	支持	强	长轨迹 Agent · 高吞吐 Rollout 与训练
ROLL	阿里巴巴	支持	支持	强	通用 Agentic RL · 多任务
SkyRL	NovaSky-AI	支持	支持	强	Search-R1 / WebRL 类 Agent
AgentLightning	微软	支持	支持	**极强**	Agent 逻辑 / 训练完全解耦
slime	THUDM	支持	支持	强	高灵活性Rollout GLM训练框架

异步流水线 · agent 逻辑解耦 · 长轨迹 + 多工具 + 多环境并行

2.4 Part 02 — Agentic RL 小结

三个核心挑战 + 一套基础设施

01 **效率 · 高效 Agentic RL**

多轮信用分配 · 长轨迹训练成本 → GiGPO 把状态分组，让多轮 RL 真正可训

02 **交互 · Agentic RL 融入工具**

推理需要环境的可验证信号 → Search-R1 把搜索 / 工具接入 RL 反馈，与环境交互

03 **决策 · 推理暂停、不确定下决策**

何时查、何时停、何时grounding → GRIL 把推理暂停做成可学策略，告别虚构

04 **基础设施 · Agentic RL Infra**

推理引擎 × 训练引擎 × 环境引擎 三组件协同 — 异步、解耦、可扩展

PART 03

智能体：现代体系与关键技术

工具学习 · 长程规划 · 记忆 · 心智 · 技能 · 自进化

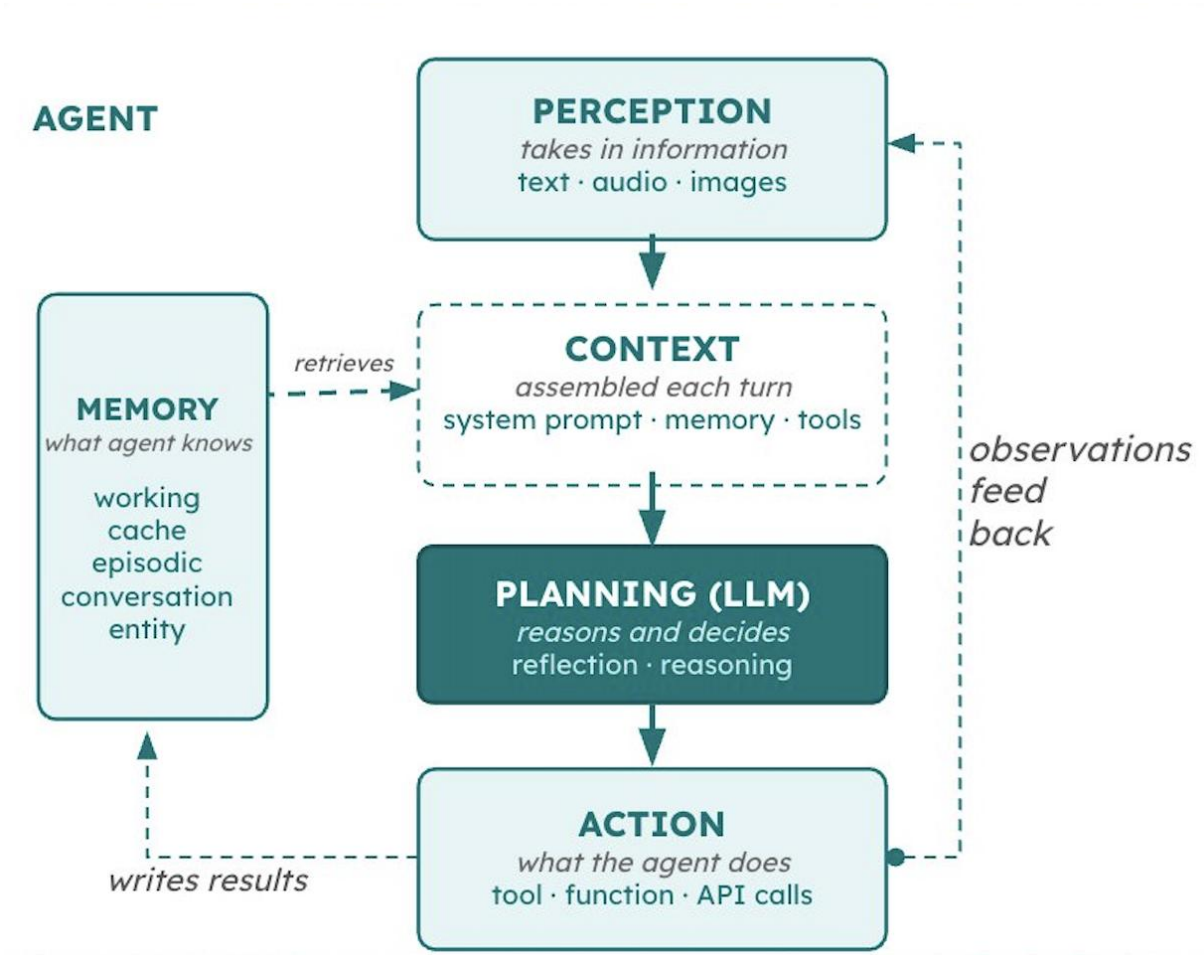
3.1 早期智能体

2023 — Plan-Execute-Reflect

» **HuggingGPT** · LLM 作为 controller，调度 HuggingFace 上的专家模型完成任务

» **AutoGPT / BabyAGI** · Plan-Execute-Reflect 雏形

» **MetaGPT / ChatDev** · 多角色协作框架



提示工程脆弱 · 无记忆 · 无学习 · 工具调用靠正则匹配

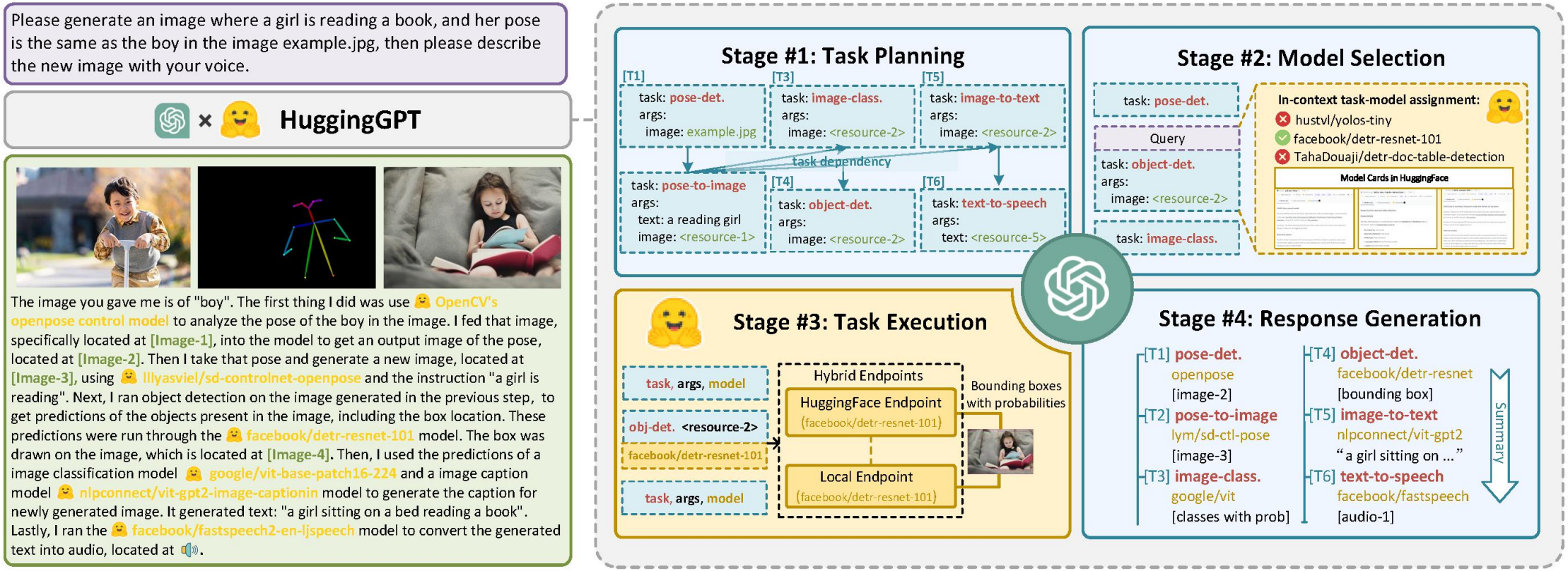
3.1 HuggingGPT — 早期智能体的代表

HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in HuggingFace

核心思想：LLM 作为 controller，调度HuggingFace 上的专家模型完成任务

四阶段流程：1. Task Planning → 2. Model Selection → 3. Task Execution → 4. Response Generation

代表意义：*首个把 LLM 系统化作 agent controller 的工作，基于大小模型协同构建任务自主的AI Agent系统*



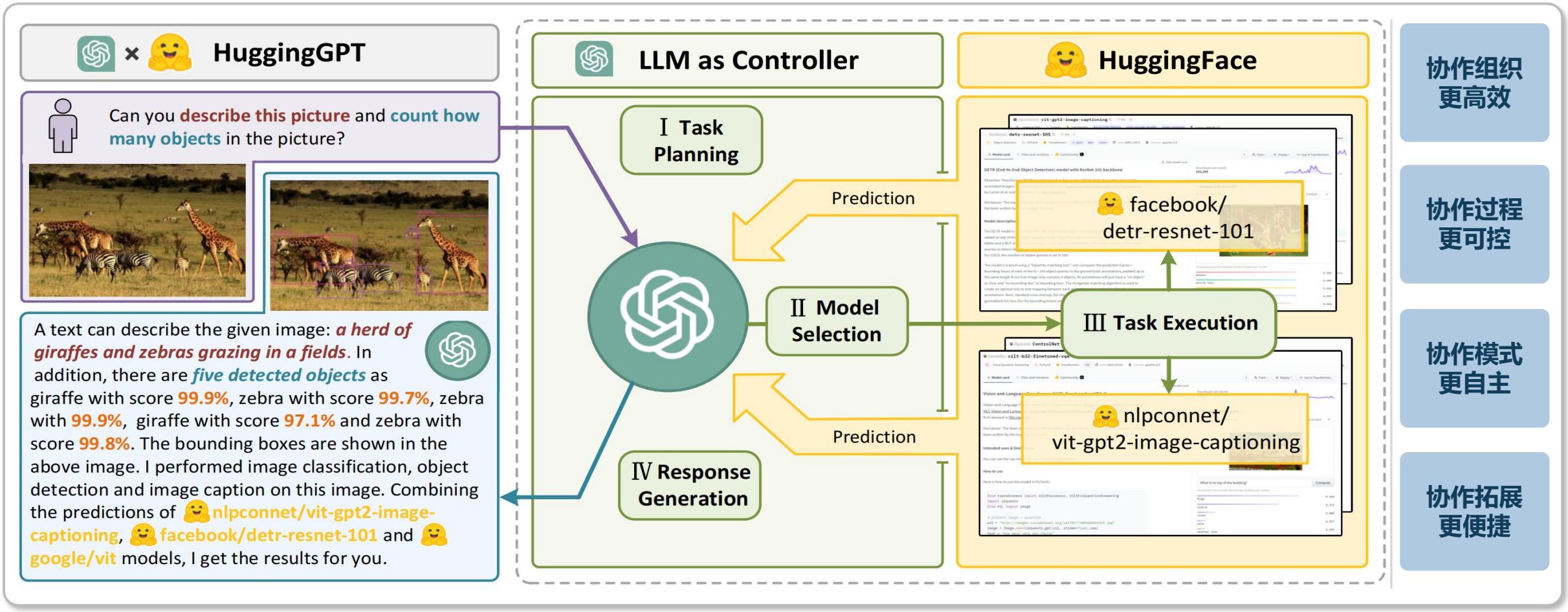
[Shen, Song, Tan, ..., Zhuang. HuggingGPT. NeurIPS 2023 — 沈永亮、宋恺涛、谭旭等]

3.1 HuggingGPT — 早期智能体的代表

学术贡献：提出中心化大小模型组织协作机制

大模型作为中心控制器来协调小模型，任务分发和消息传递均以大模型为中介。

One model to do anything? ✘
One model to rule anything! ✓



3.2 现代智能体的三大跃迁

01 Prompt 编排 → **端到端原生 Agentic 能力**

从 "在 LLM 外面套 agent 框架" 到 "训练原生具备 agentic 能力的模型"

02 Prompt Engineering → **Context → Harness Engineering**

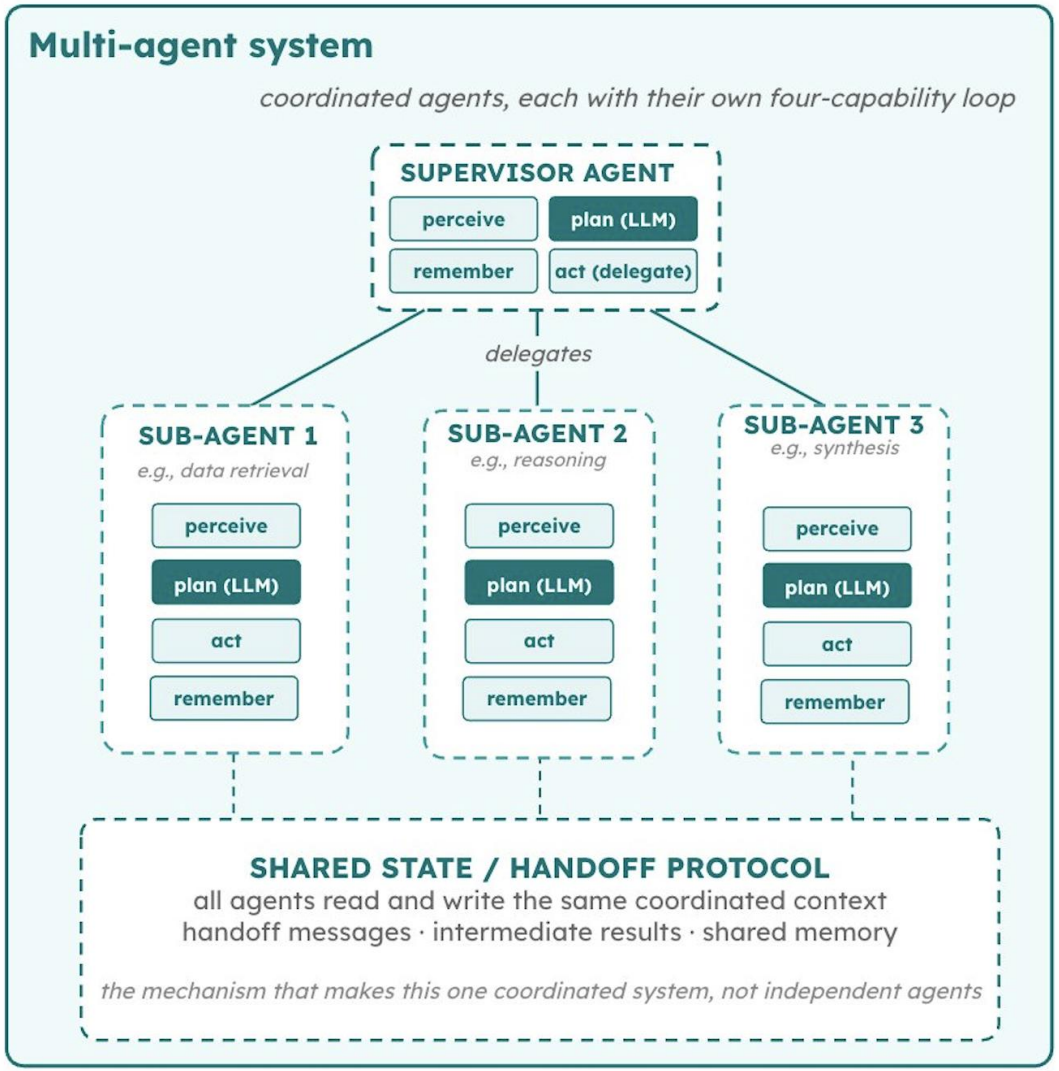
Prompt（问什么）→ Context（喂什么）→ Harness（agent 如何运转的整套工程）

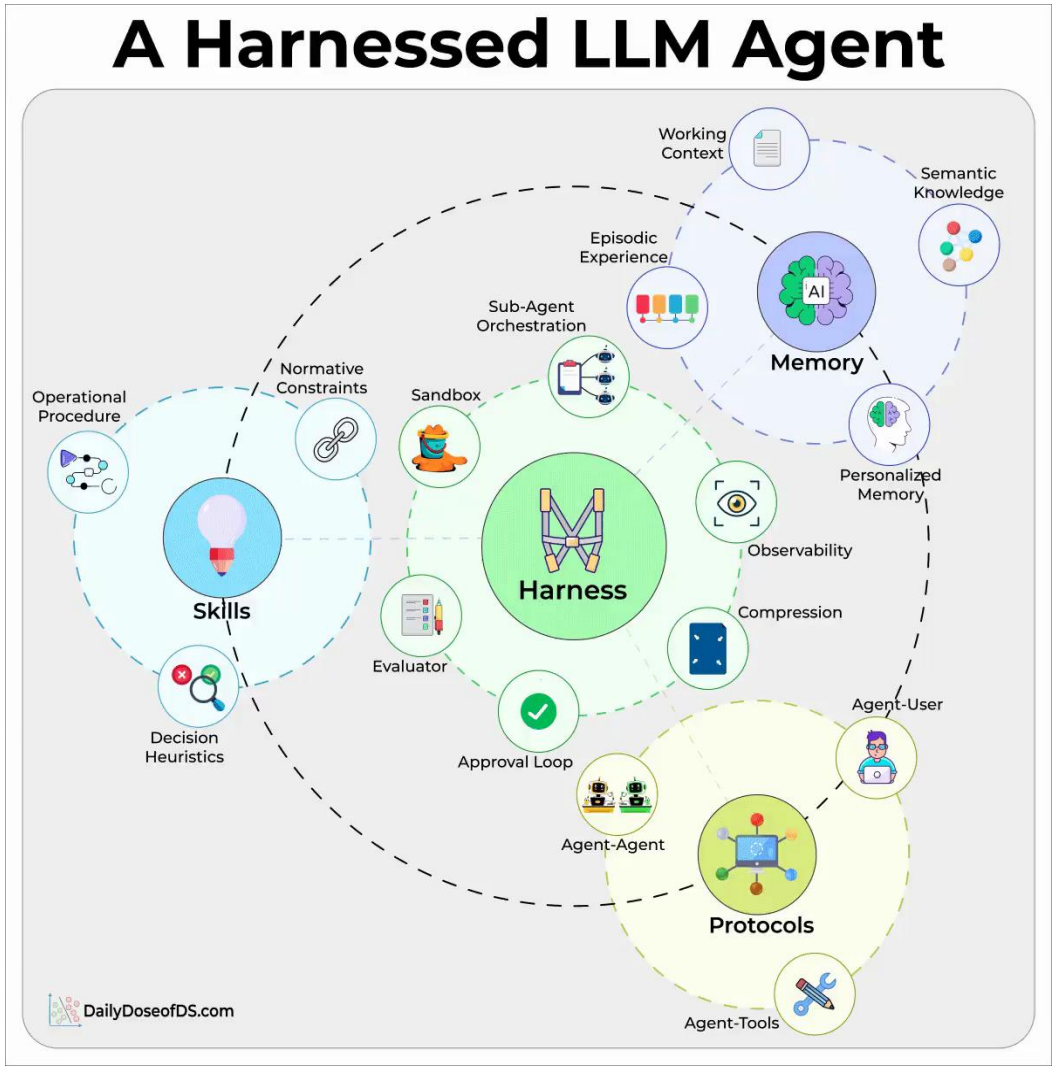
03 Single Agent → **Agent Swarm**

从单 agent 长链执行 → 多 sub-agent 并发协同（Kimi K2.6 / MiniMax M2.7）

Agent 不再是 LLM 的外挂工程，而是模型自身就要被训练成 agent

3.2 现代智能体的三大跃迁





3.2 Claude 产品演进

Claude Chat *2023+*
对话协作者

Claude Code *2025/02 RP*
Agentic 编码

Claude Cowork *2026/01 RP*
工作助手

Claude Design *2026/04*
视觉创作者

[Anthropic 2024-2026 release notes; Claude.ai/blog]

3.2 Claude Code

Anthropic · 2025/02 RP · 2025/05 GA · CLI 形态 · Anthropic 最快增长产品

核心能力

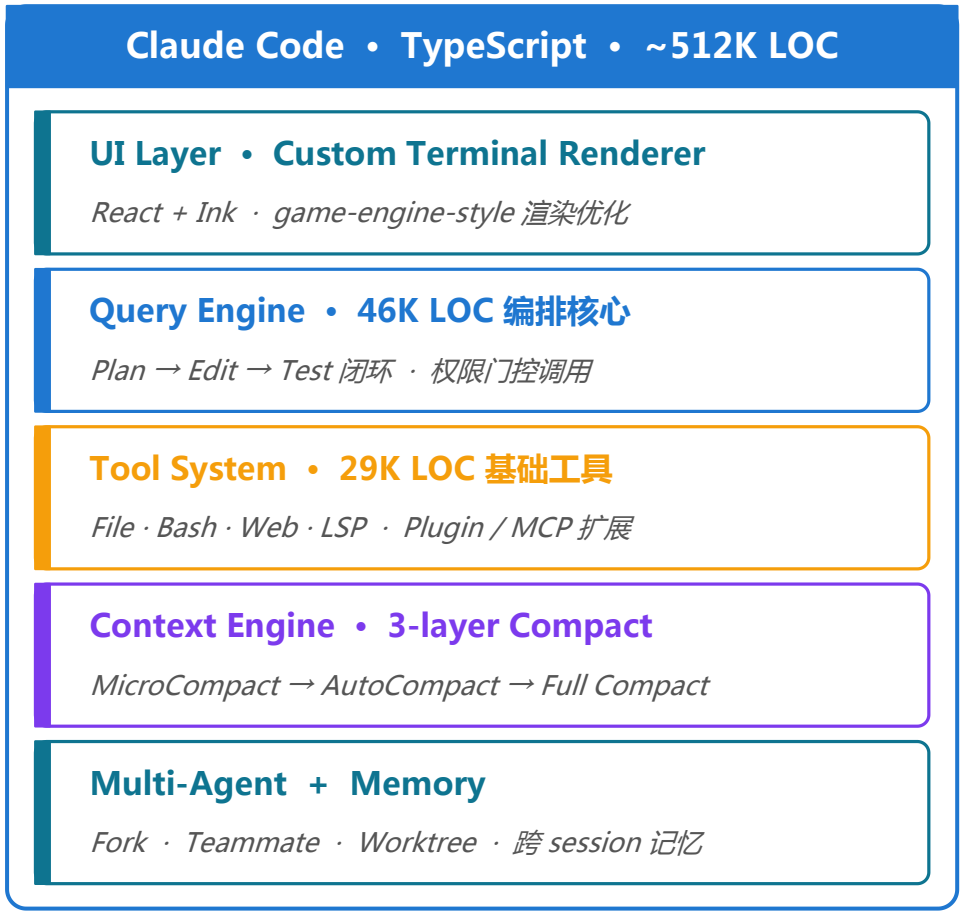
» CLI-native — 终端直接运行
» 仓库级理解 — 自动建索引
» 长程自主 — 数小时连续工作
» 多文件编辑 + 测试驱动循环

工程化能力

» Routines / Subagents / Hooks
» Plugins · MCP-style 扩展

进阶模式

» Auto Mode / Ultrathink / Memory



[Anthropic. Claude Code. 2025/02 RP · 2025/05 GA · 持续迭代至 2026]

3.2 国内 Agent 进展 — Swarm × Self-Evolution

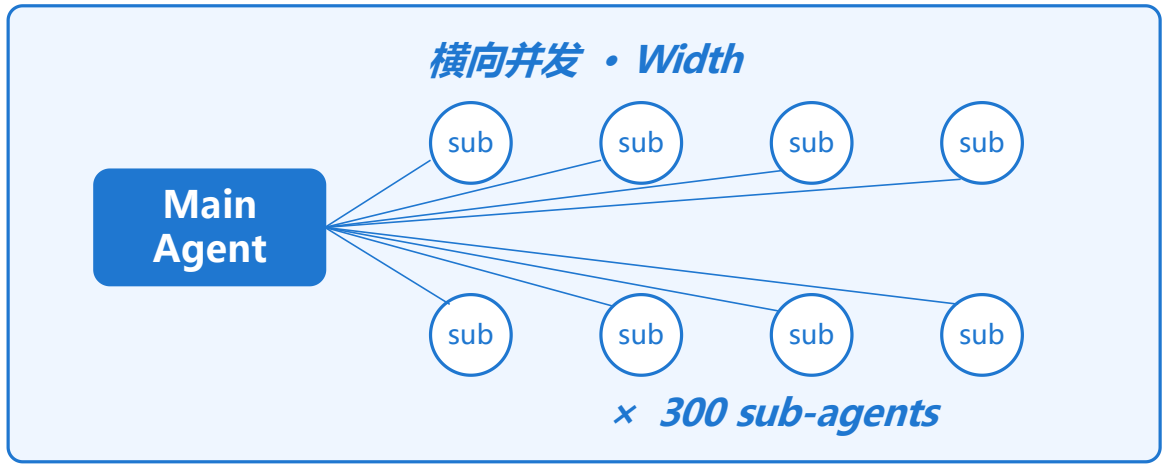
Kimi K2.6 与 MiniMax M2.7 — 两条并行的国内开源 agent 路线

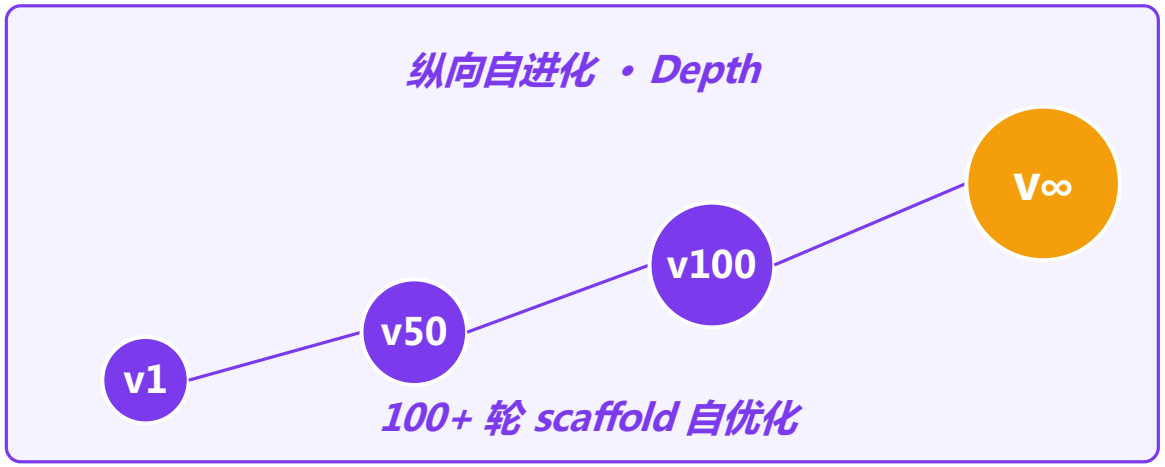
Kimi K2.6 — Agent Swarm 路线

» 300 sub-agent 横向并发
» 4000 步协同执行
» 13 小时长程编码
» Reusable Skills + Claw Groups

MiniMax M2.7 — Self-Evolution 路线

» 训练中 100+ 轮 scaffold 自优化
» SWE-Pro 56.22% / Terminal Bench 2 57.0%
» Agent Teams + 复杂 Skills
» MM Claw 技能合规 97%





国内厂商在 Agent 时代选择了与 Anthropic / OpenAI 不同的两条创新路径

3.2 Kimi K2.6 — Agent Swarm 与长程编码

Moonshot AI · 2026/04/13 · 1T 参数 / 32B 激活 · 开源

Agent Swarm

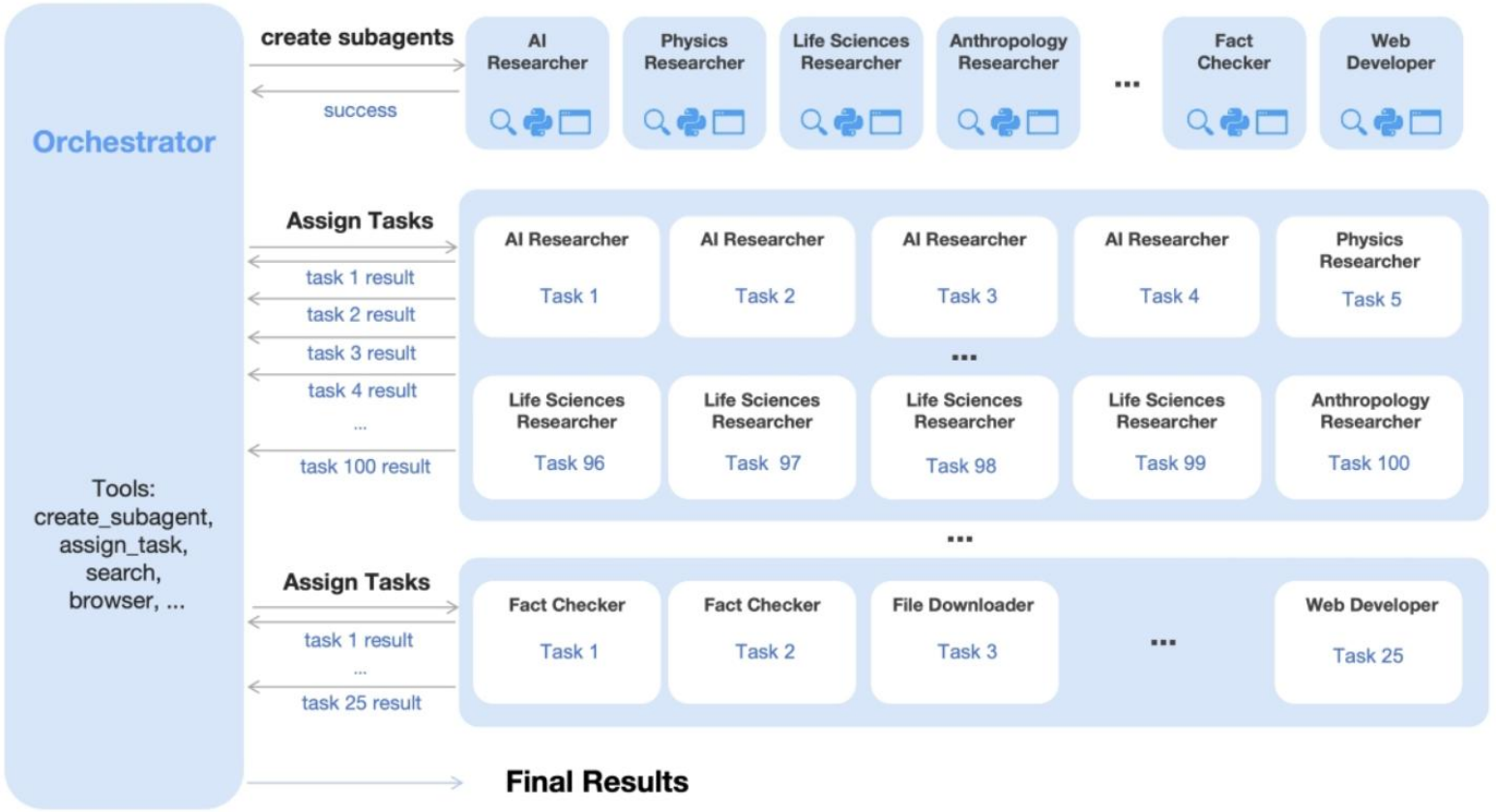
» 300 sub-agent 横向并发
» 4000 步协同执行
» 异构 agent 自动协作

Long-Horizon Coding

» 13 小时持续自主执行
» 12 轮迭代 / 1000+ 工具调用
» 改写 4000+ 行代码

Reusable Skills + Claw Groups

» 文档自动转可复用 skill
» 跨设备 · 跨模型协作生态



从单一 agent 到 swarm — 把任务并行化推到 300× 量级

[Moonshot AI. Kimi K2.6 Tech Blog 2026/04]

3.2 MiniMax M2.7 — 自进化 Agent

MiniMax · 2026/04 · 230B MoE · 开源 · SWE-Pro 56.22% / Terminal Bench 2 57.0%

Self-Evolution 训练范式

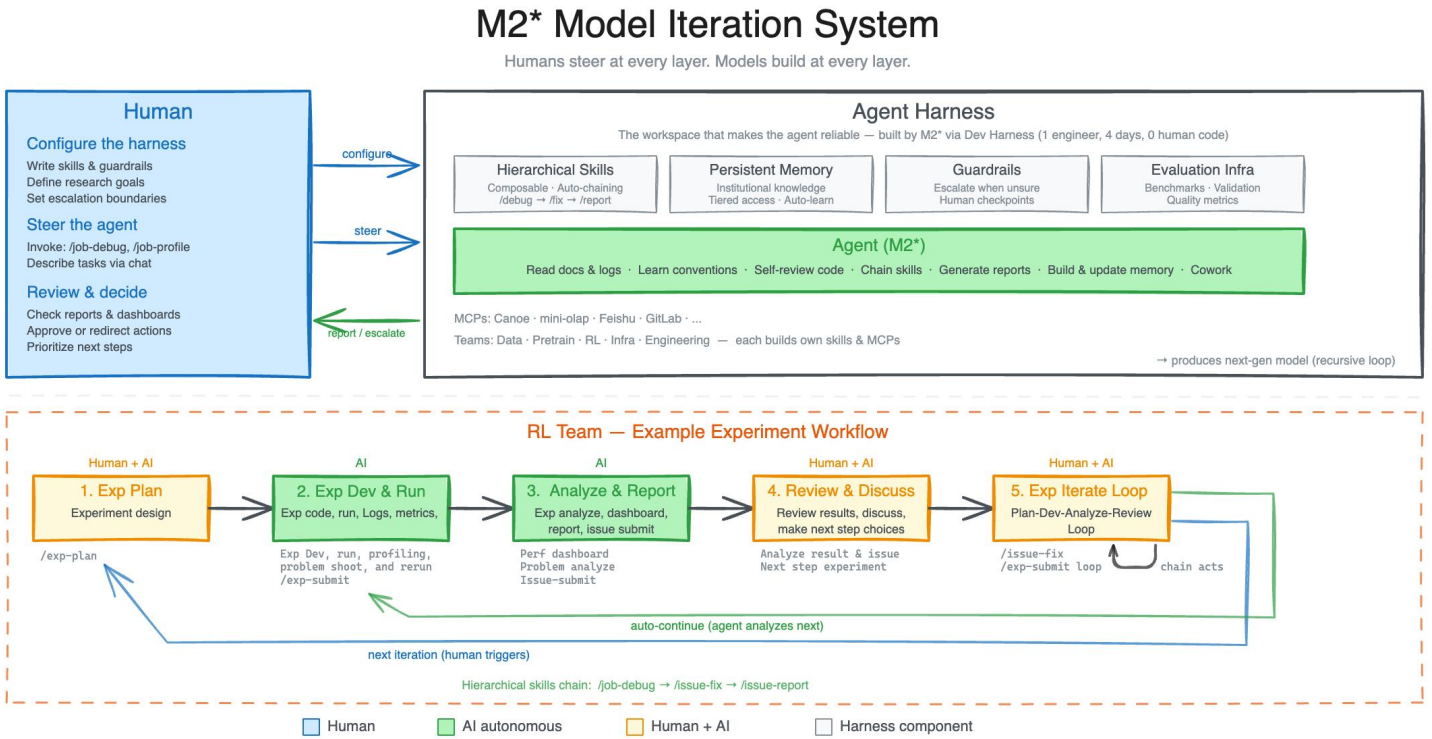
» 训练中 100+ 轮 scaffold 自优化
» 内部 evals +30% · 完全无人工

Agent 核心能力

» 复杂 agent harness 自主构建
» Agent Teams + 复杂 Skills
» 动态工具搜索 (dynamic tool search)

代表性基准

SWE-Pro 56.22% · VIBE-Pro 55.6%
Terminal Bench 2 57.0% · Toolathon 46.3%
MM Claw 技能合规率 97%



Self-Evolving — 模型自己优化自己的训练 scaffold，是自进化的雏形

[MiniMax. M2.7: Early Echoes of Self-Evolution. 2026/04]

3.2 开源 Agent 框架版图

LangGraph	*LangChain*	**状态化工作流**	图状态机 · 长运行流程 · 可断点续跑
AutoGen	*Microsoft*	**对话式多 agent**	对话编排 · agent 间消息 + 协商
CrewAI	*CrewAI Inc.*	**角色化团队**	Role + 目标 + 任务流
MetaGPT	*DeepWisdom*	**软件流程模拟**	PM/Architect/Engineer 协同
Hermes Agent	*Nous Research*	**本地自我成长**	Self-Evolution + 跨模型 + 持久记忆
OpenClaw	*openclaw/openclaw*	**本地多平台**	20+ 聊天通道 + Voice + 跨设备
nanobot	*HKUDS · 港大*	**极简个人 agent**	4k 行 Python · MCP 原生 · 多 LLM provider
ClawGUI	*ZJU-REAL*	**GUI Agent 全栈**	构建于 OpenClaw + nanobot · 跨设备 GUI 自主交互

从 Workflow 框架 → 自主成长 agent → 本地极简形态 — 开源生态的全谱系

[LangGraph · AutoGen · CrewAI · MetaGPT · Hermes Agent · OpenClaw · nanobot (HKUDS) · ClawGUI]

3.2 OpenClaw — 开源个人 AI Agent

github.com/openclaw/openclaw · Any OS · Any Platform · The lobster way

Local-First Gateway

» 单控制面统一管理 sessions / channels / tools / events

Multi-Agent Routing

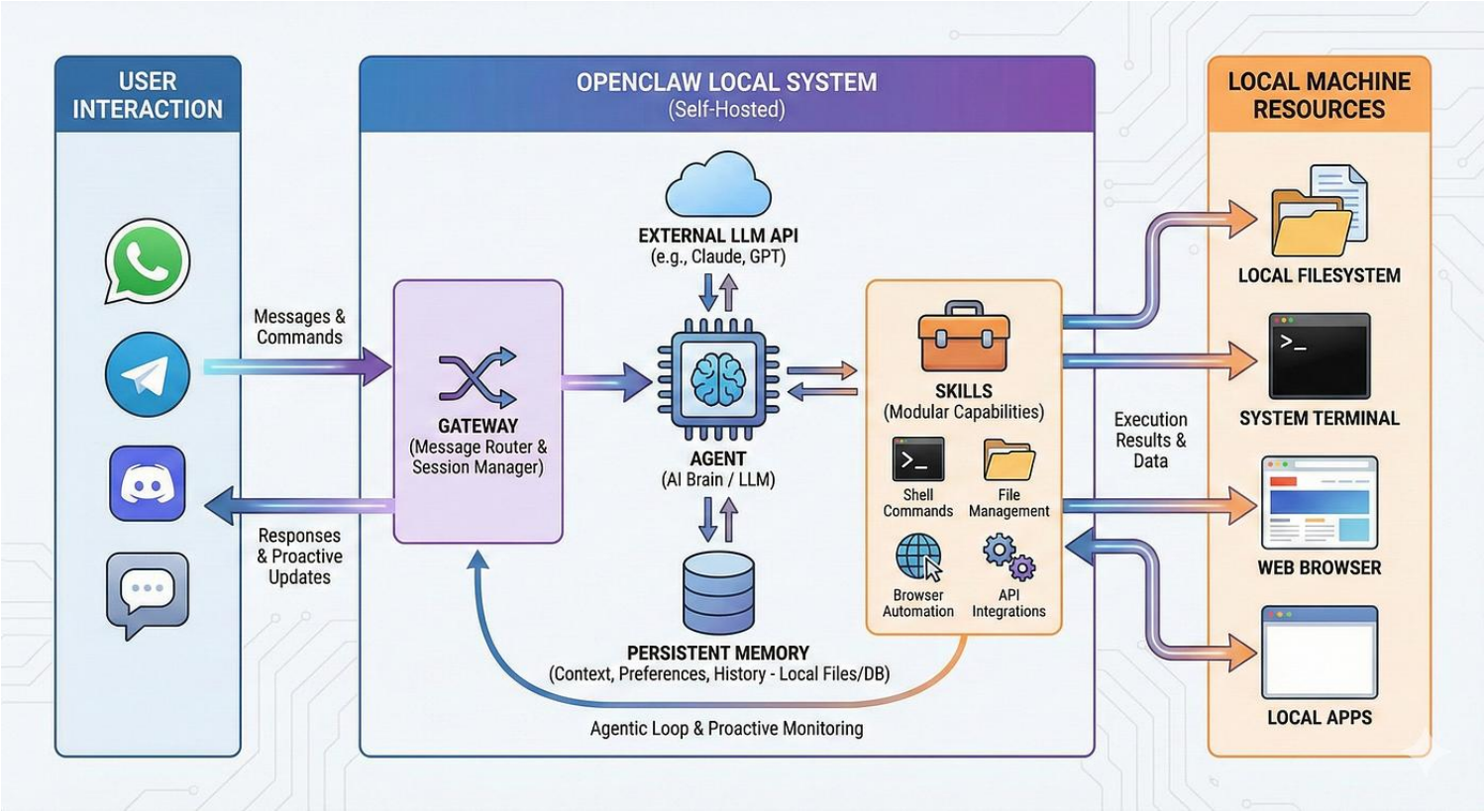
» 多 channel / 账号 / peer 路由到独立 agent workspace

Voice + 20+ 聊天通道

» macOS/iOS/Android 跨设备
» WhatsApp/Telegram/Slack/飞书/微信 …

First-class Tools

» browser · canvas · cron · sessions



[github.com/openclaw/openclaw · openclaw.ai · docs.openclaw.ai]

3.2 Hermes Agent — 自我成长的本地智能体

Nous Research · github.com/NousResearch/hermes-agent · The agent that grows with you

Self-Evolving Learning Loop

» 业内首个内置学习闭环 agent
» 复杂任务后自主创建技能
» 跨 session 构建用户画像

DSPy + GEPA 进化框架

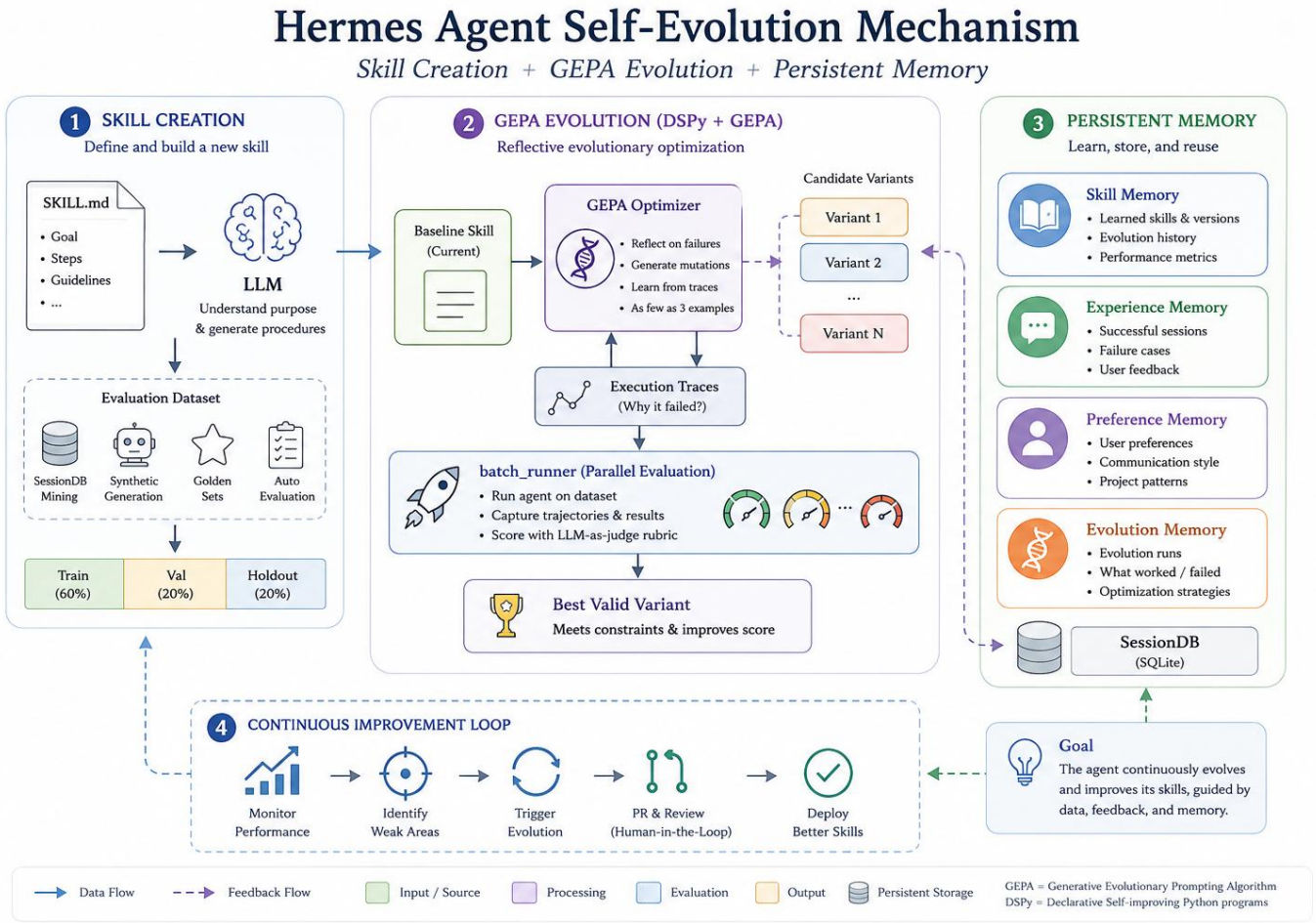
» 遗传式 Prompt + 技能 + 代码协同优化

Memory + 跨会话回忆

» FTS5 全文检索 + LLM 摘要

Model-Agnostic

» 200+ 模型一键切换，无 vendor 锁定



[hermes-agent.nousresearch.com · Hermes 4 Technical Report 2025]

3.2 ClawGUI — 跨形态 Agent 系统

ZJU-REAL · 构建于 OpenClaw / nanobot 之上 · CLI + GUI 协同

★ 1.1k Stars HuggingFace Paper Daily — Rank 1

CLI ⊕ GUI 协同

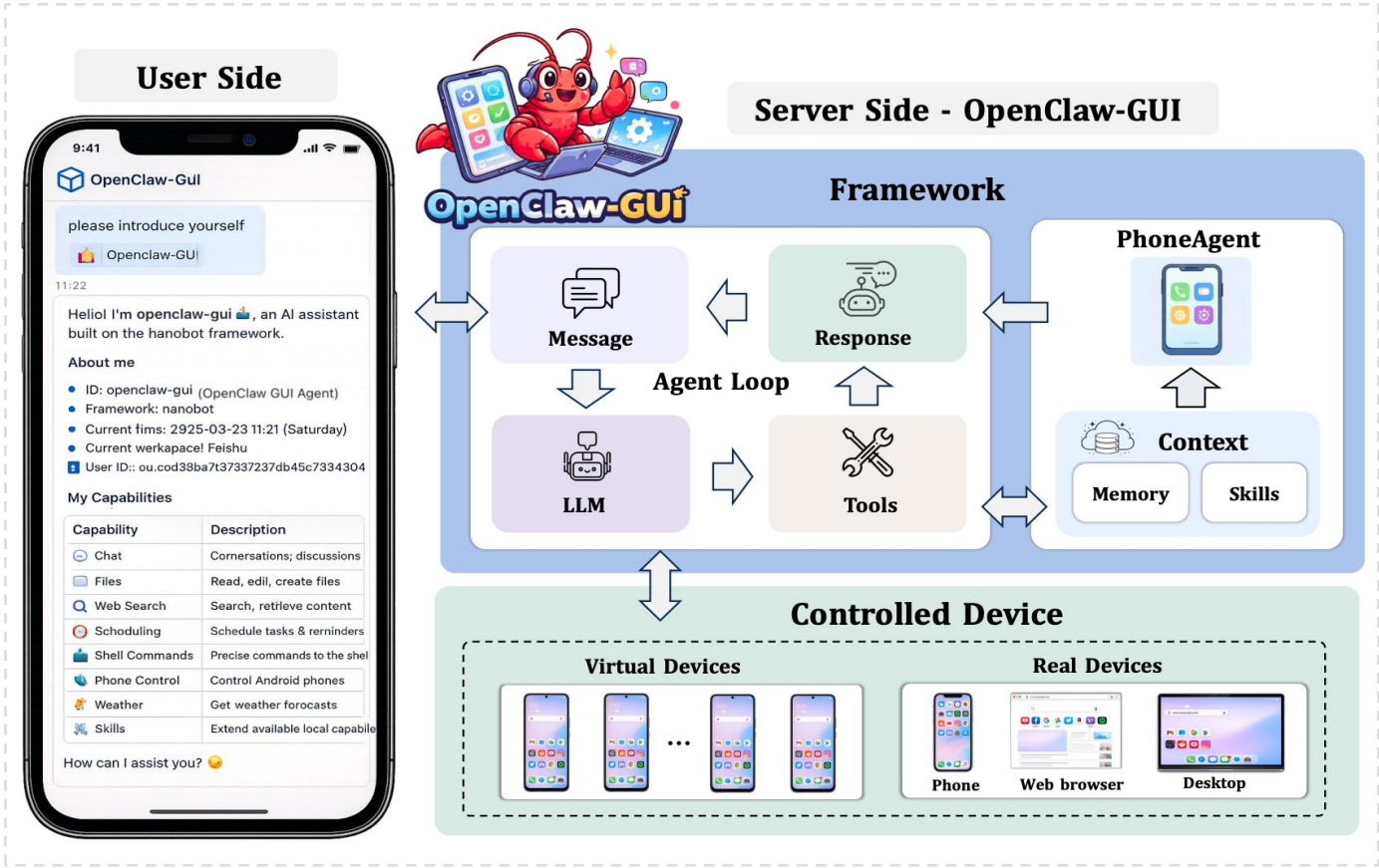
» CLI 端 — 精确指令 / 后台编排 / 调试
» GUI 端 — 真机操控 / 视觉反馈 / 用户交互
» 同一 agent 同时驱动两种界面

GUI Skills

» 把可复用的 GUI 操作封装为 skill
» 跨 App / 跨设备调用同一 skill
» 统一纳管多种开源GUI Agent为skill

GUI Memory

» 屏幕状态 + 动作历史持久化
» 用户个性化记忆 + 持久化保存
» 跨 session 的 GUI 上下文记忆



把 CLI 的精确性和 GUI 的通用性融合到一个 Harness 框架 中

[github.com/ZJU-REAL/ClawGUI]

3.2 ClawGUI — 真机部署 Demo

从飞书 / 微信 / Telegram 远程指令 → Android / iOS / HarmonyOS 真机自主完成跨应用任务

1. 跨设备 — 在桌面端发指令，手机端真机执行
2. 跨平台 — Android / HarmonyOS / iOS 一套 agent 通用
3. 可视化 — 全程截图回放 + 推理轨迹可追溯

3.3 现代智能体的六大关键技术

1 **工具学习**
Tool Use & Tool Learning

2 **长程规划**
Long-horizon Planning

3 **记忆机制**
Memory

4 **心智理论**
Theory of Mind

5 **Agent Skills**
智能体技能

6 **自我进化**
Self-Evolution

3.3.1 工具学习 — Tool Use & Tool Learning

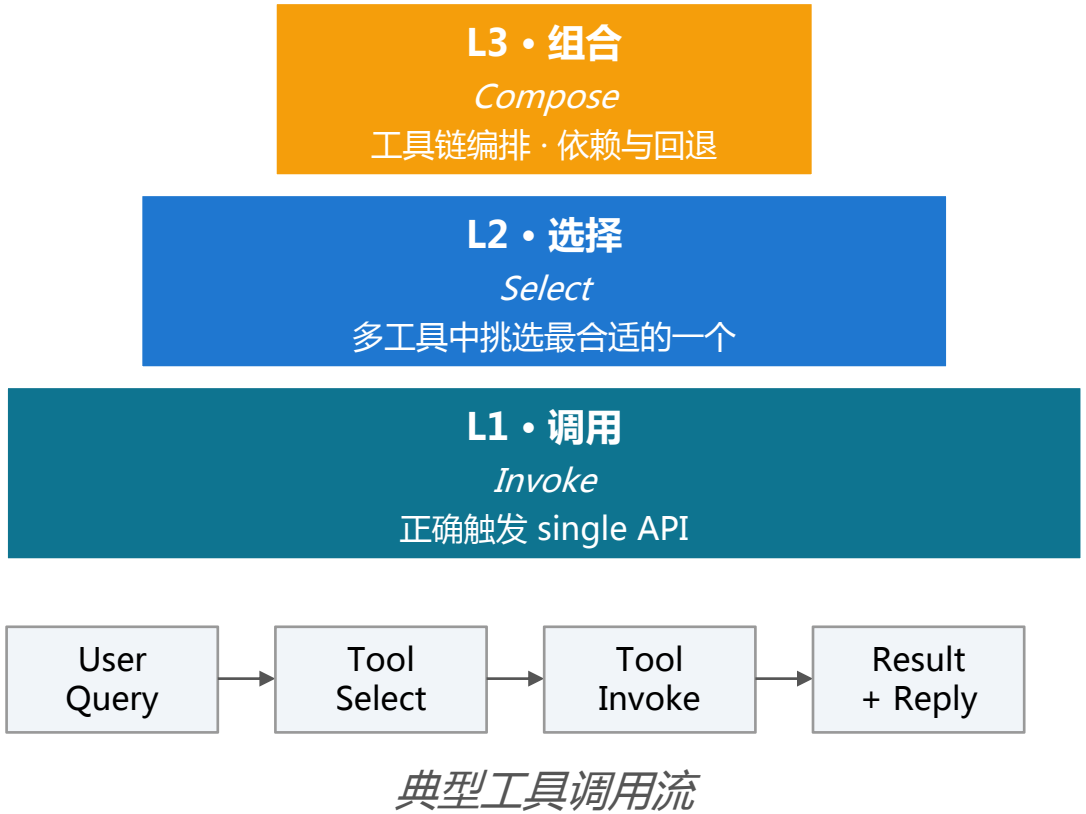
演化路径 — Function Calling → 协议化接口 → 原生工具调用模型

三个层次

» 调用 — 正确触发
» 选择 — 多工具中挑对
» 组合 — 工具间依赖与编排

真实场景两大痛点

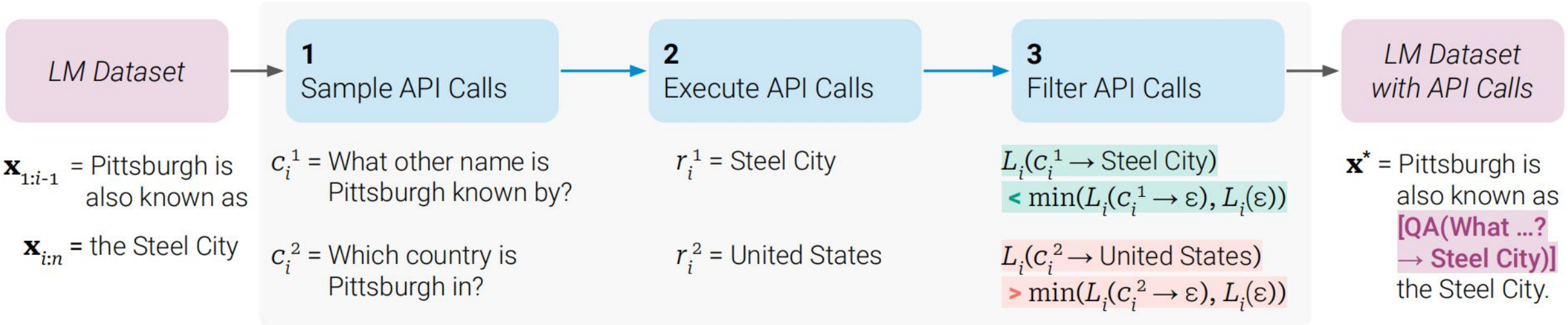
» 用户 query 模糊
» 工具调用不可逆且代价高



典型工具调用流

3.3.1 Toolformer — 工具学习的奠基之作

Toolformer: Language Models Can Teach Themselves to Use Tools



核心问题

如何让 LLM 自学习何时、如何调用外部 API？

Self-Supervised 范式

» 在文本中自动插入 API 候选调用
» 用 perplexity 衡量调用是否有用
» 仅保留有效调用作训练数据

代表意义

确立了自监督工具学习的范式，之后 ToolLLM / xLAM / 原生工具模型都建立在该思想之上

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

3.3.2 长程规划 — Long-horizon Planning

规划长序列动作以达成远期目标

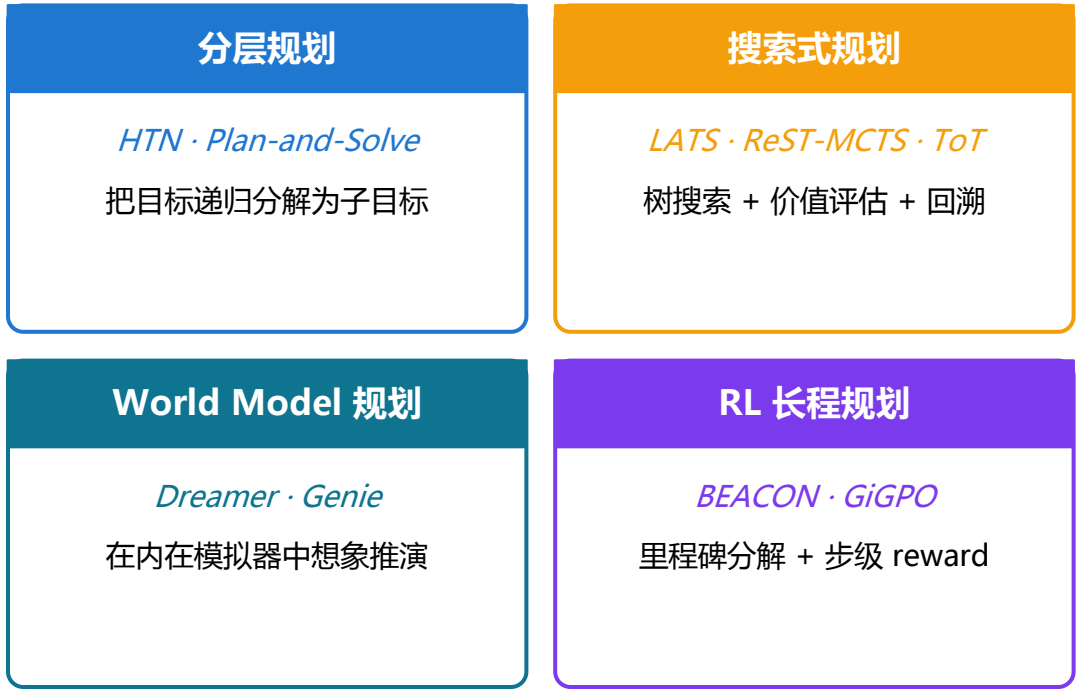
① 分层规划
把目标递归分解为子目标
代表：HuggingGPT · ADaPT · LLM+P
瓶颈：子目标质量靠 prompt，不可学

② 搜索式规划
树搜索 + 价值评估 + 回溯
代表：ToT · LATS · ReST-MCTS · AlphaProof
瓶颈：步级 value 估计噪声大；成本随深度爆炸

③ World Model 规划
在内在模拟器中想象推演
代表：RAP · WebDreamer · Reasoning-via-Planning
瓶颈：模型误差累积，与真实环境失配

④ RL 长程规划
用 RL 直接优化长程决策策略
代表：BEACON · GiGPO · ARPO
瓶颈：长 horizon 下 credit assignment 困难

方法分类



3.3.2 LATS — 长程规划的代表方法

Language Agent Tree Search · ICML 2024

动机

ReAct / CoT 单链推理无回溯，错一步全盘皆输

LATS 关键设计

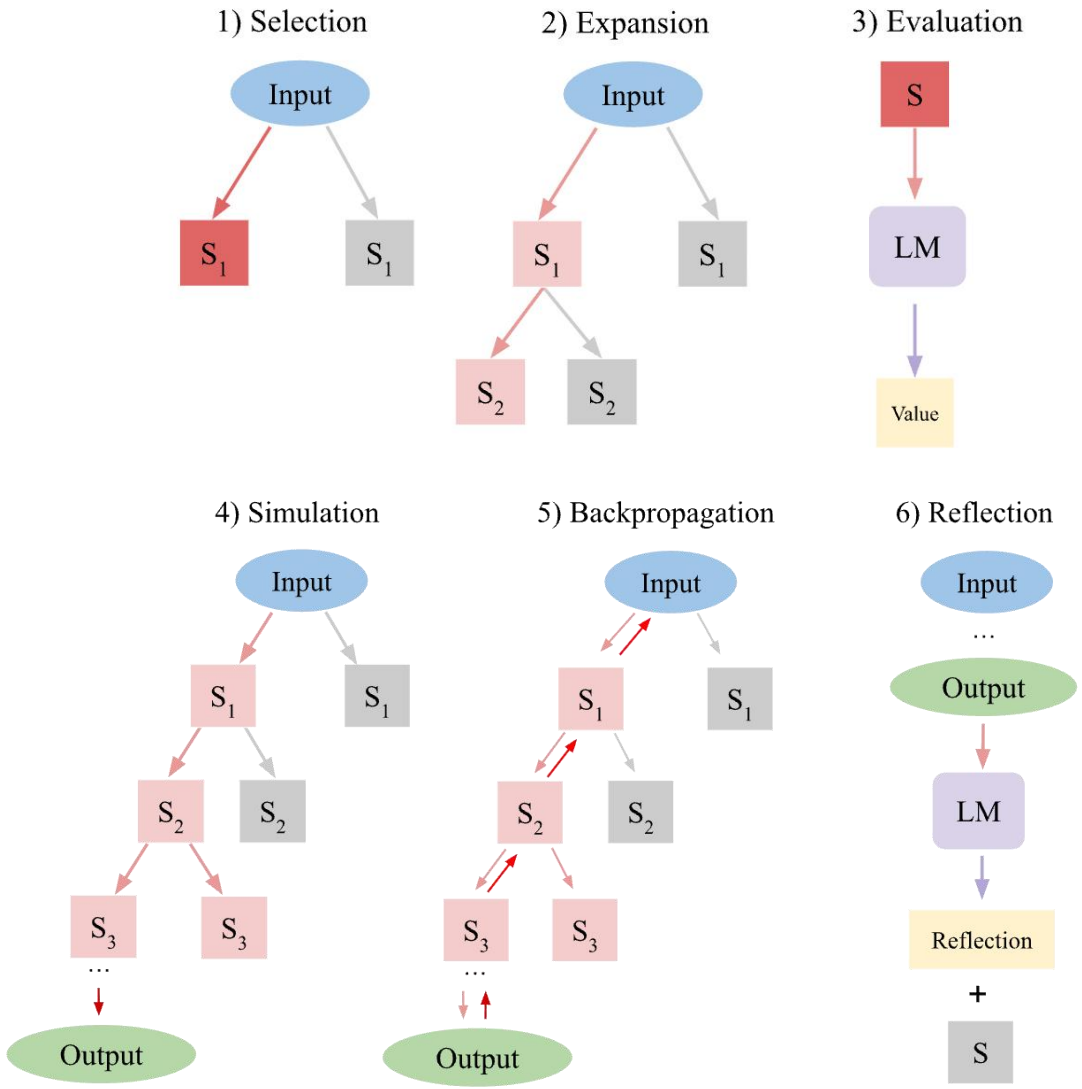
» MCTS 四步：select → expand → evaluate → backprop
» LM-powered value：LLM 自评节点，无需训练 critic
» Self-reflection：失败轨迹 → 反思反馈 → 注入下一轮

实验效果

HumanEval 94.4% (GPT-4) · WebShop 75.9 (GPT-3.5)，
全面超越 ReAct / Reflexion / ToT / RAP

代表意义

把"试错 + 反思"显式做成搜索结构
Language Agent 第一个统一推理 / 行动 / 规划的框架



[Zhou et al., LATS: Language Agent Tree Search Unifies Reasoning, Acting, and Planning, arXiv:2310.04406, ICML 2024]

3.3.2 BEACON — 里程碑引导的策略学习

OUR WORK

Milestone-Guided Policy Learning

人类如何完成长任务？

不是一气呵成，而是里程碑分解

BEACON 把这一思想 RL 化

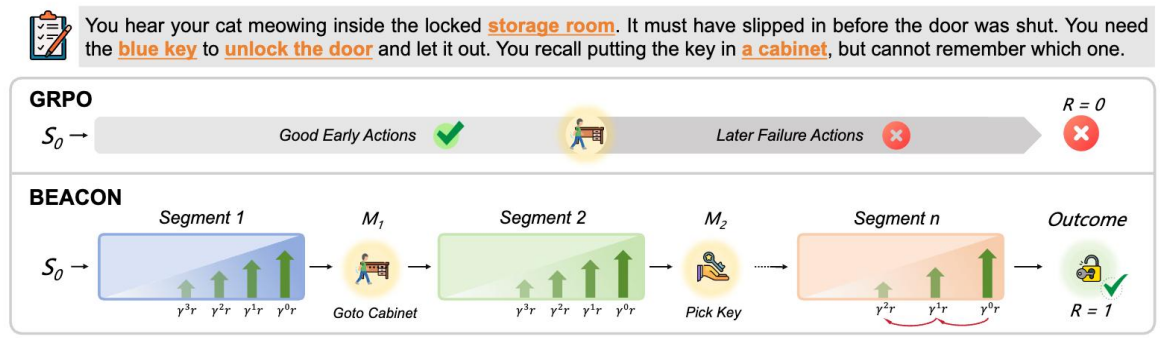
» 自动从轨迹中识别里程碑

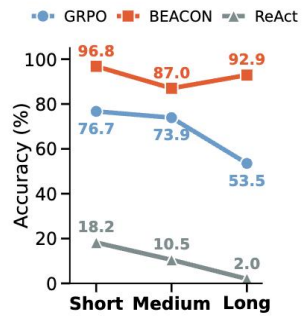
» 把稀疏终局 reward 折射为密集的里程碑 reward

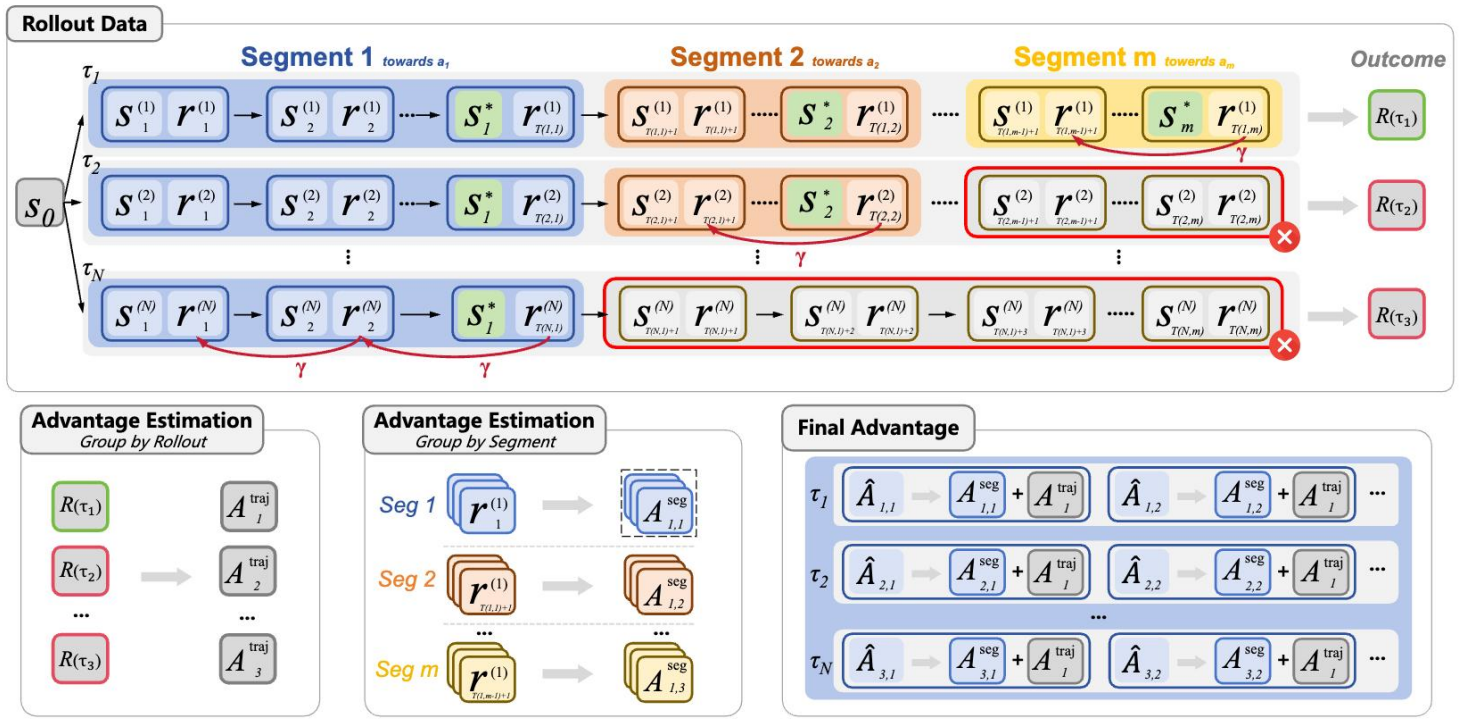
» policy 在 milestone 间分层学习

实验效果

ALFWorld 长程任务 53.5 → 92.9，SciWorld / WebShop 同步超越 GRPO / PPO / RAGEN







[ICML 2026]

3.3.3 记忆机制 — Memory

Agent 持久化和检索过去经验的能力，跨 session 学习的基础

定义

Agent 持久化、组织、检索过去交互
经验的能力 — 区别于 LLM 的固定权重

短期记忆

上下文窗口管理 · attention sink · KV 压缩

长期记忆

向量 RAG → 结构化（MemGPT / Letta / A-MEM / Mem0）

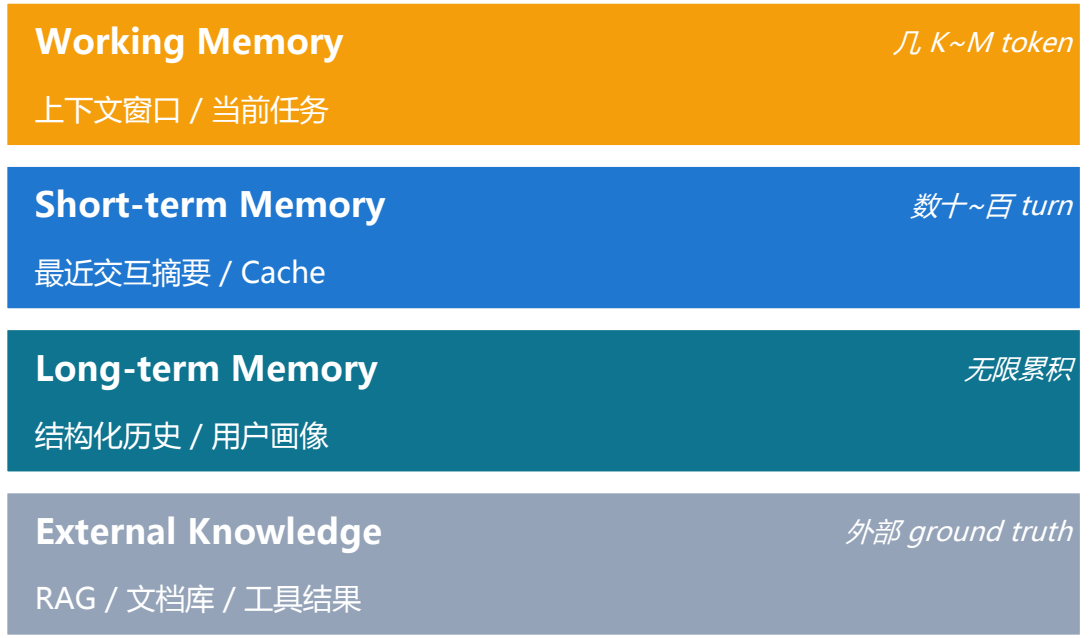
在智能体中的意义

» 跨 session 持续学习与个性化
» 避免重复错误 · 沉淀知识资产

问题与挑战

» 容量管理 · 检索准确性
» 遗忘与冲突解决
» 隐私与一致性边界

方法分类 • 分层记忆架构



3.3.3 MemGPT / Letta — 把 LLM 当作操作系统

MemGPT: Towards LLMs as Operating Systems

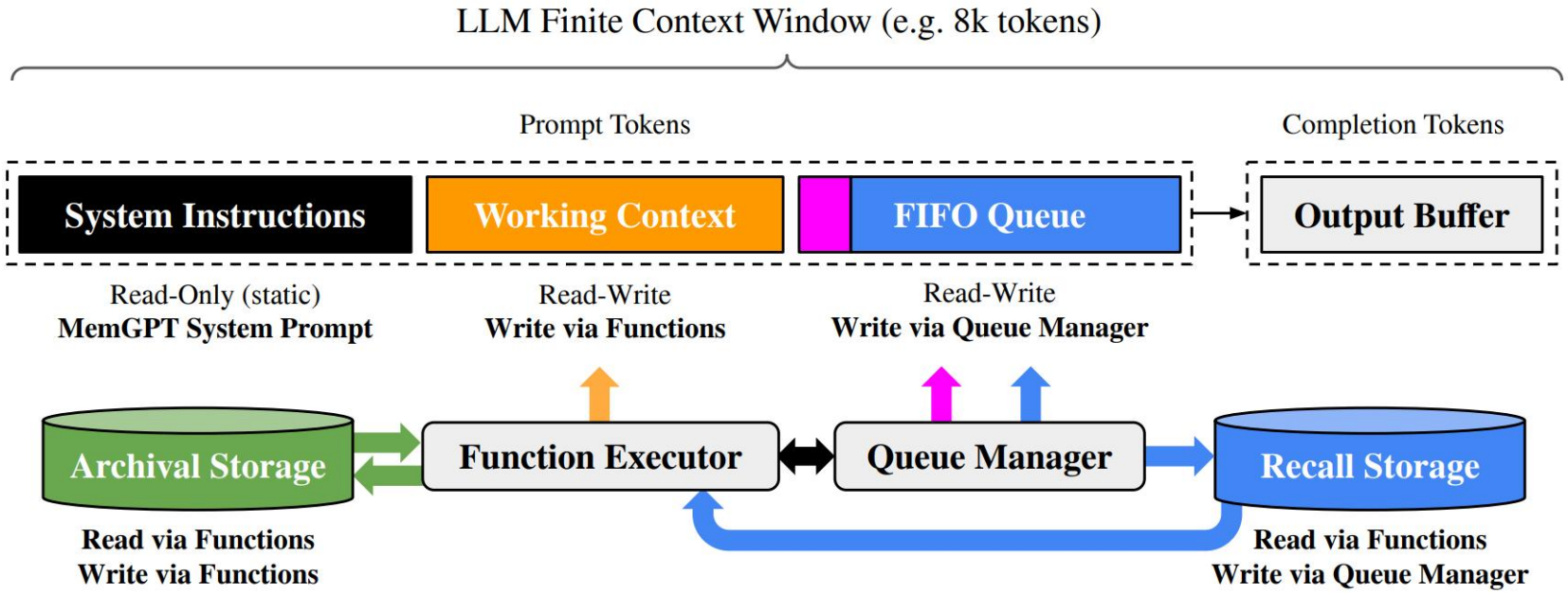
类比：借鉴虚拟内存的概念把上下文窗口看作 "主存"， 外部存储看作"虚拟内存"

分层记忆架构

» Main Context (主存) — 实时对话
» Working Context — 最近交互摘要
» Recall Storage — 全量对话历史
» Archival Storage — 长期知识库

Function Calling 主导记忆调度

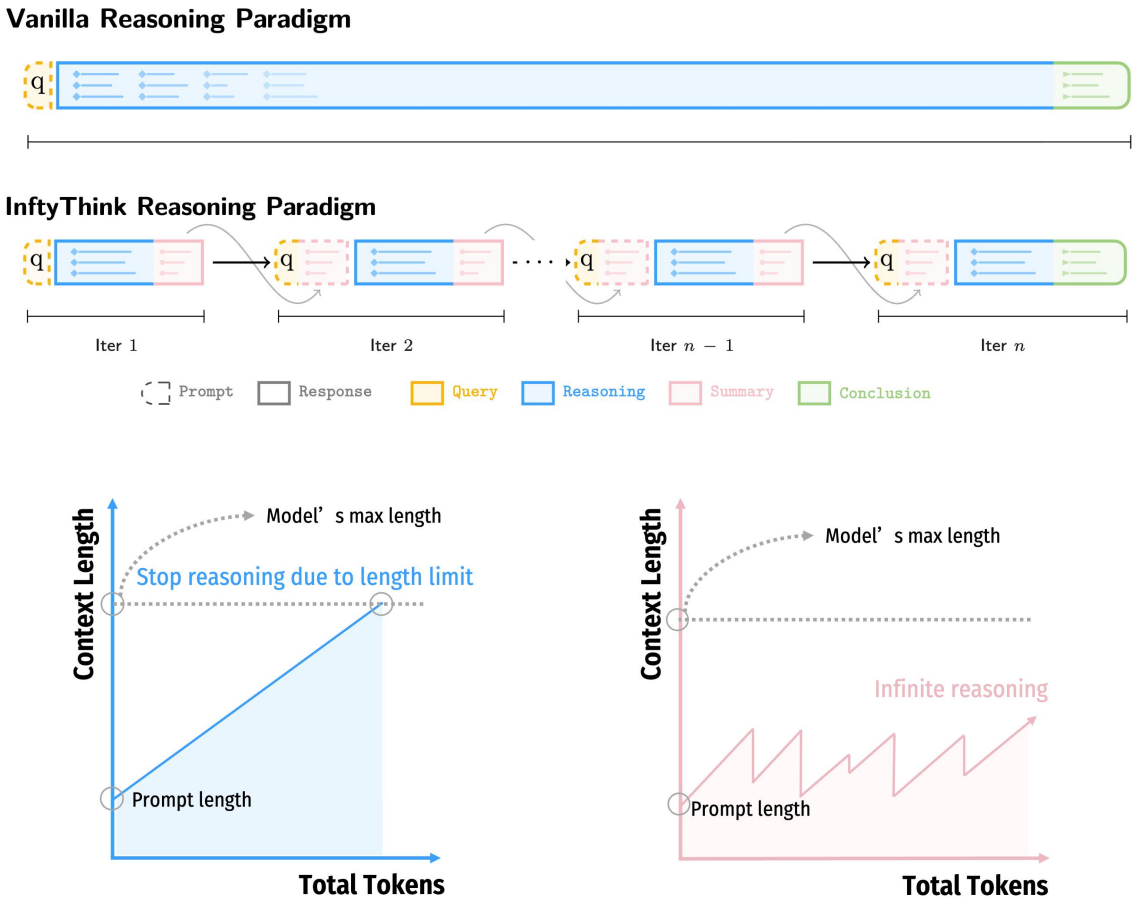
LLM 自己决定何时把内容 swap 入/出主存



[Packer et al. MemGPT: Towards LLMs as Operating Systems. UC Berkeley 2023]

3.3.3 InftyThink+ — 把记忆做成可学习的策略

Effective and Efficient Infinite-Horizon Reasoning via Reinforcement Learning



以自发的阶段性总结为基础的迭代式推理范式

» InftyThink+ 让记忆成为模型自身的策略
» 突破模型上下文窗口限制，实现无限视野的推理
» 降低大模型平方复杂度带来的长程任务巨额开销

端到端强化学习优化三个模型自主策略

» 何时触发总结？
» 如何进行总结？
» 如何接续推理？

同时提升模型推理与效率

» AIME24 准确率提升 21%，超越传统长 CoT RL 方法
» 推理延迟降低超50%，提升RL训练效率

[Yan, et al. InftyThink: Breaking the Length Limits of Long-Context Reasoning in Large Language Models. ICLR 2026]
[Yan, et al. InftyThink+: Effective and Efficient Infinite-Horizon Reasoning via Reinforcement Learning. ICML 2026]

3.3.4 心智理论 — Theory of Mind

推理他人信念、意图与心理状态，多智能体协作与人机协同的认知前提

定义

推理他人 (人 / Agent) 心理状态的能力 —信念 / 意图 / 知识 / 偏好

在智能体中的意义

» 多智能体协作的协调基础
» 人机交互中理解用户真实意图
» 谈判 / 教学 / 反馈对齐

主流方法

信念建模　对比学习　多世界推理　时空建模

问题与挑战

» 心理状态不可直接观测
» 跨视角推理（自我 vs 他人）
» 错误信念与反事实推理

3.3.4 ToMBench

系统化双语 ToM 基准：8 大任务 · 31 项能力 · 2860 道题

问题动机

» 现有 ToM 评测零散、单语
» 数据污染严重 — 题目多源自人类心理学经典
» 缺乏系统能力分类

核心贡献

» 从零构建的中英双语 benchmark
» 8 大社交任务 × 31 项 ToM 能力
» 多选题格式 — 自动化、无歧义评估

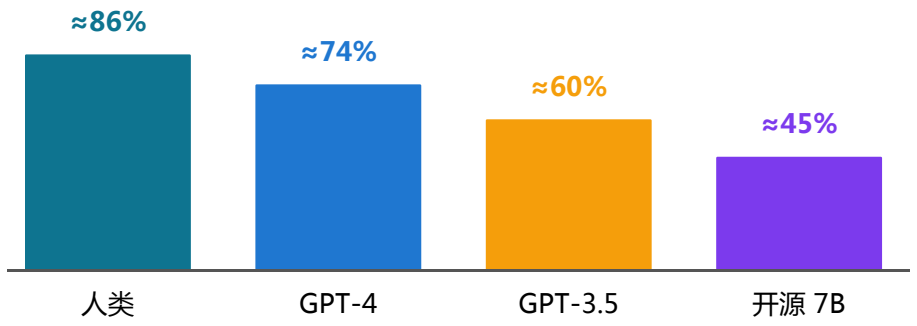
关键发现

» GPT-4 仍比人类落后 10+ 分
» 高阶 ToM、faux pas 任务掉头明显
» LLM 尚未达到人类级 ToM

8 大社交认知任务

意外结果 *Unexpected Outcome*	标度含义 *Scalar Implicature*	说服故事 *Persuasion Story*	失礼识别 *Faux-pas Recognition*
错误信念 *False Belief*	奇异故事 *Strange Story*	暗示测试 *Hinting Test*	歧义故事 *Ambiguous Story*

人类 vs GPT-4 平均准确率



LLM 离 "人类级 ToM" 仍有显著差距 — 二阶推理与 faux pas 是主要瓶颈

[Chen et al. ToMBench: Benchmarking Theory of Mind in Large Language Models. ACL 2024]

3.3.4 TimeToM

用 时序空间 + 信念求解器 打开 LLM 的 Theory-of-Mind

问题

» CoT / ToT 在 ToM 上几乎无效
» 高阶 ToM 涉及多角色 / 多时刻信念链
» LLM 易忽略事件时序与跨视角推理

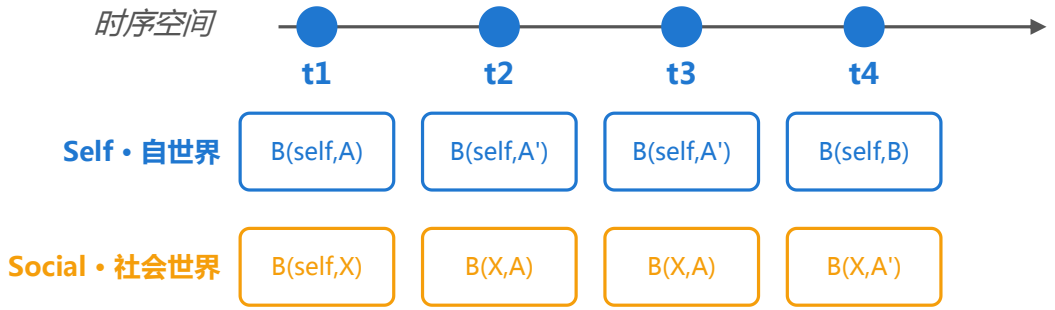
核心思路

» 构建 时序空间 (Temporal Space)
» 为每个角色建立 时序信念状态链 TBSC
» Self-world / Social-world 双链分离

关键创新

» Tool-Belief Solver — 高阶信念 → 一阶信念
» 在通信窗口内做信念传递
» 跨多场景 ToM 数据上显著提升

Temporal Belief State Chain (TBSC)



Tool-Belief Solver

高阶信念 ⇒ 一阶信念 · 通信窗口内做信念传递

结果 — *在 ToMI / Hi-ToM / FANToM 多个数据集上显著超越 CoT / ToT*

[Hou, Zhang, Shen, Wu, Lu. TimeToM. ACL Findings 2024 · Hou, Zhang, Zheng, Shen, Lu. Sprinkle Cognitive Aha Moment. ACL Findings 2025]

3.3.5 Agent Skills

Skills 是 Agent 的可复用资产，从一次性使用到跨任务沉淀

什么是 Skill

结构化打包的 — 指令 + 脚本 + 资源
可被 agent 按需发现并加载

两种实现路径

» 推理时加载 (Anthropic Skills)
» 训练时进化 (SkillRL)
» 训练时内化 (Skill0)

为什么重要

Skills 让 Agent 把 "做过的事"
沉淀为下次更快的捷径

3.3.5 Anthropic Agent Skills — 详解

把可复用的过程性知识打包成 Claude 可发现的模块

文件系统化架构

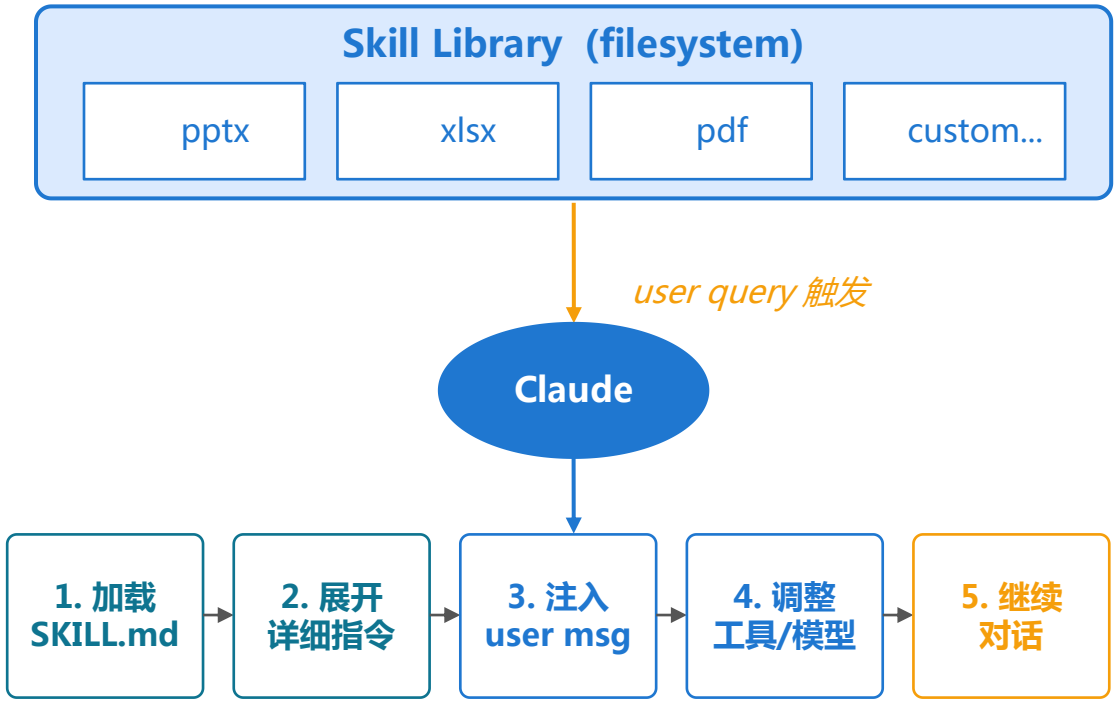
» 每个 Skill 是一个目录
» SKILL.md (说明) + scripts/ + resources/
» 在 Claude 的虚拟机中执行

Progressive Disclosure

» 启动时只预加载 name + description
» 命中后才展开完整 SKILL.md
» 上下文体积理论上无上限

执行工作流

1. 加载 SKILL.md
2. 展开为详细指令
3. 注入新 user message
4. 调整工具 / 模型选择
5. 在丰富后的环境中继续对话



官方内置 — *PowerPoint · Excel · Word · PDF · 17 开源 Skills*

从 Tool Use 到 Skill 加载 ，Agent 从"调用 API"升级为"调用专家手册"

[Anthropic Engineering Blog · github.com/anthropics/skills · platform.claude.com/docs]

3.3.5 SkillRL — 把 Skill 接入 RL 闭环

Evolving Agents via Recursive Skill-Augmented Reinforcement Learning

核心问题

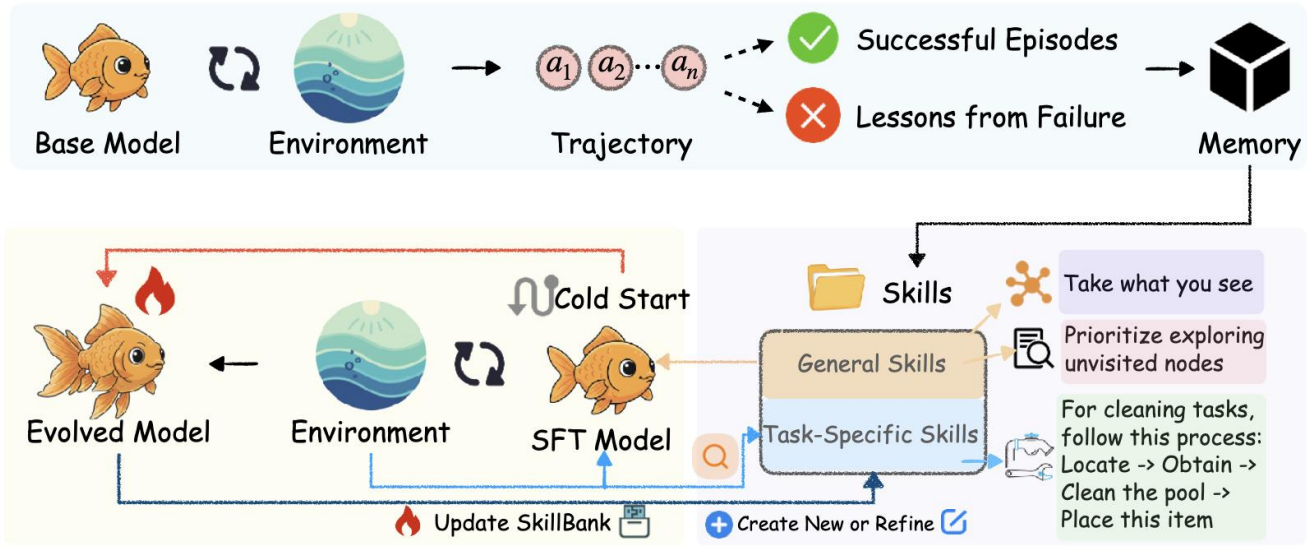
现有 memory-based agent 只存原始轨迹
无法抽出可复用的高层行为模式

三大组件

» Experience Distillation — 经验蒸馏出 SkillBank
» Adaptive Retrieval — 自适应检索通用 + 任务专用
» Recursive Evolution — 技能库与 policy 共进化

实验结果

ALFWorld / WebShop / 7 个搜索任务上 SOTA
7B 模型 +41% over GPT-4o · 比 baseline +15.3%



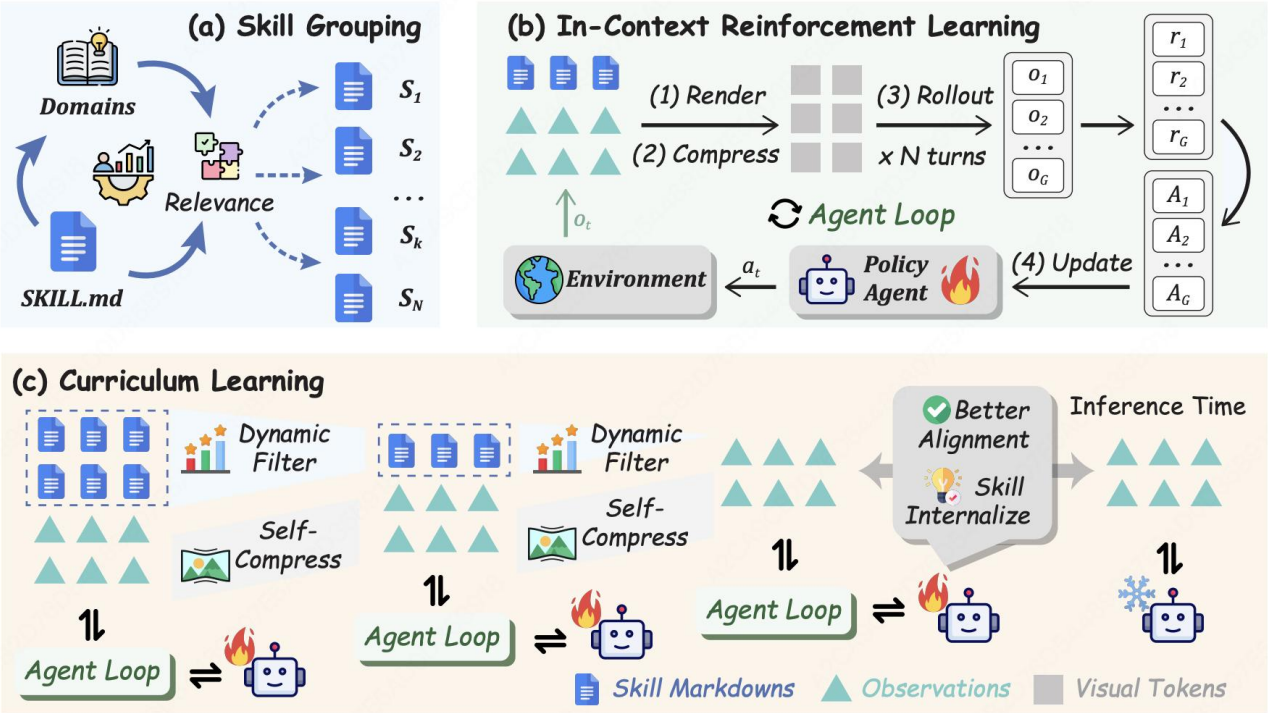
把 Skill 从 in-context 能力升级为可被 RL 优化的核心组件

[Yao et al. SkillRL: Evolving Agents via Recursive Skill-Augmented RL. 2026 — arXiv:2602.08234]

3.3.5 Skill0 — 把技能内化到权重

OUR WORK

In-Context Agentic Reinforcement Learning for Skill Internalization



视觉渲染上下文

用视觉编码器压缩技能和交互历史，极致节省token

上下文强化学习

技能增强的多轮rollout，利用环境反馈进行参数更新

渐隐式技能课程

» 训练初期: 完整提供技能描述

» 中后期: 衡量技能文件对于当前策略的“有用性”，从contex中 逐步撤回增益不大的技能文件

» 模型把外部知识内化为策略

推理时 < 0.5k token / step

ALFWorld: +9.7%

Search-QA: +6.6%

Anthropic Skills 让模型用上技能，Skill0 让模型不再依赖技能描述

[arXiv:2604.02268 · github.com/ZJU-REAL/SkillZero]

3.3.6 自我进化 — Self-Evolution

Agent 能否独立改进自己？自进化的四种范式

01 纯自博弈式 无外部数据 / 无人类监督 — 模型既出题又解题，自己进化课程

代表 · Absolute Zero

02 对抗式 把任务拆为两个角色，让它们对抗共进化

代表 · Code-A1（Code LLM ↔ Test LLM）

03 协同式 同一模型在 Solver / Verifier 之间交替，互相 bootstrap

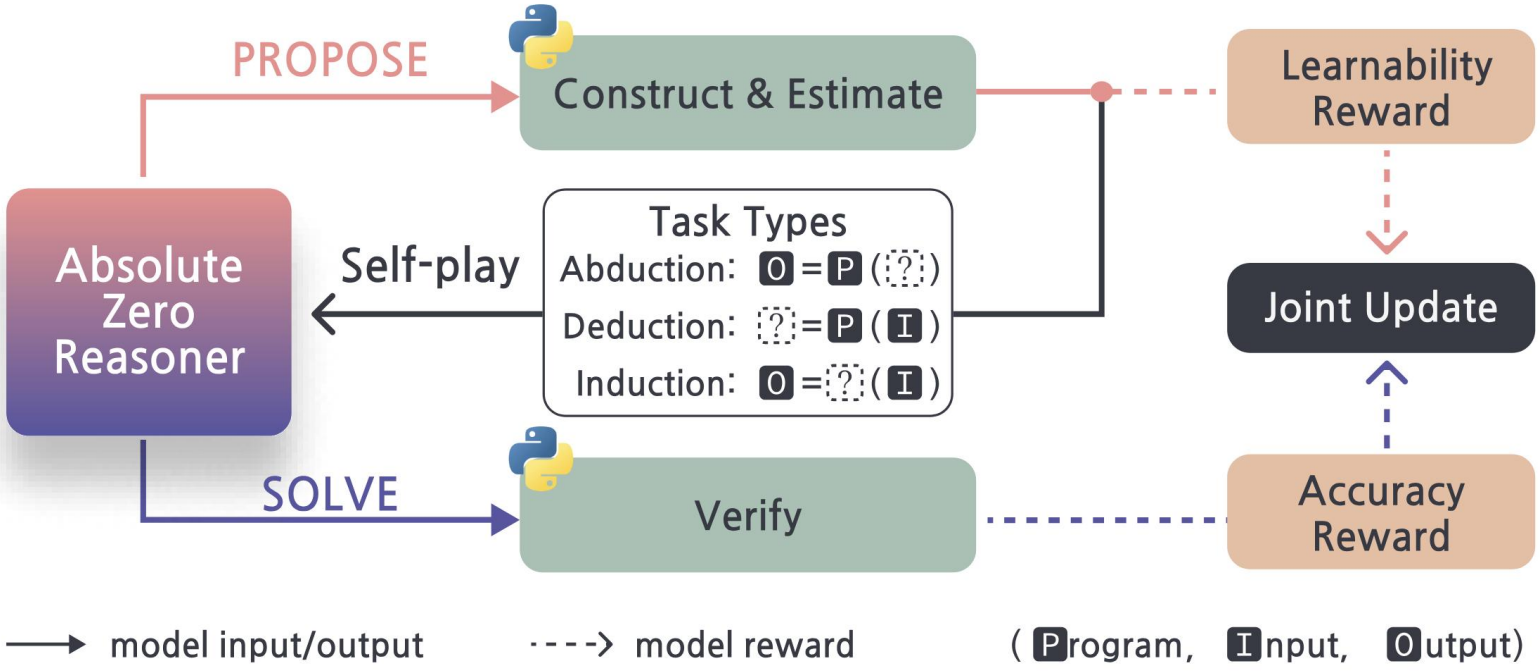
代表 · CheckMate · CoVerRL · Skill0

04 群体进化式 多 agent 共享技能库，遗传 / 选择 / 交叉演化技能与 prompt

代表 · Hermes Self-Evolution

3.3.6 Absolute Zero — 零数据自博弈推理

Absolute Zero: Reinforced Self-play Reasoning with Zero Data



范式跃迁

现有 RLVR 仍依赖人工题库

AZR 完全去除外部数据依赖
模型通过自博弈自主演化

AZR 工作机制

» 单模型同时扮演Proposer与Solver

» Code executor 作统一可验证奖励源

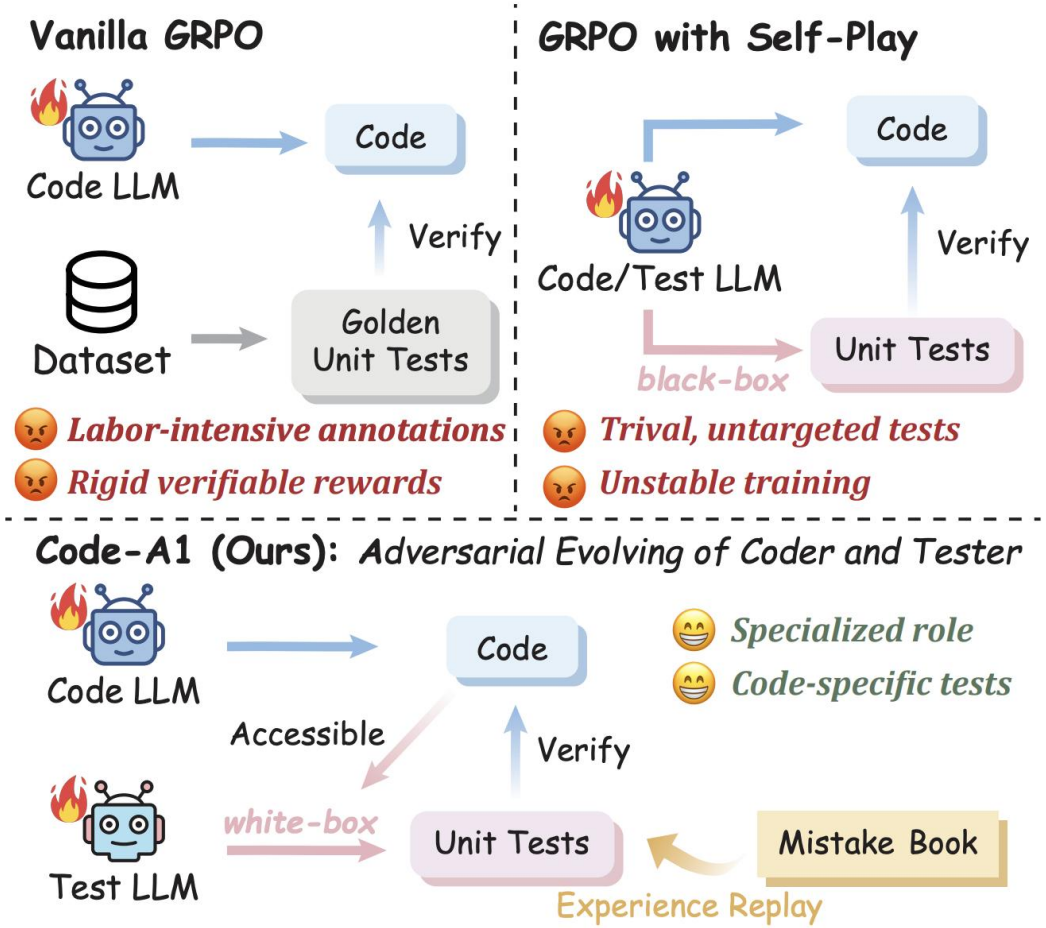
» Learnability reward 引导自演化课程难度自动随模型能力提升

代表意义 零外部数据训练，在数学与代码推理上均达到 SOTA，全面超越依赖数万条人工标注样本的 zero-setting 模型，**证明 zero-data 自博弈也能全面超越人类标注数据训练**

[Zhao et al. Absolute Zero: Reinforced Self-play Reasoning with Zero Data. NeurIPS 2025 — arXiv:2505.03335]

3.3.6 Code-A1 — Code 与 Test 的对抗共进化

Code-A1: Adversarial Evolving of Code LLM and Test LLM via Reinforcement Learning



Self-Play 的两难

白盒模式 → self-collusion: 模型为自己出简单测试骗取 reward
黑盒模式 → 测试仅基于题目描述，无法针对具体实现暴露 bug

Code-A1: 解耦 + 对抗目标

» Code LLM — 通过尽可能多的测试 → 生成更鲁棒的代码
» Test LLM — 暴露尽可能多的缺陷 → 生成更有针对性的测试
» 架构解耦消除 self-collusion，安全启用白盒测试生成
» Mistake Book 经验回放：记录历史失败测试，防止遗忘已修复的 bug

代表意义

» 将 self-play **从单模型自博弈升级为双模型对抗共进化**
» 代码生成性能超越使用人工标注测试训练的模型
» 测试生成能力同步显著提升

[ZJU-REAL. Code-A1. 2026 — arXiv:2603.15611 · github.com/ZJU-REAL/Code-A1]

3.3.6 CoVerRL — 协同式自进化

OUR WORK

Generator ↔ Verifier 同体协同进化 — label-free 走出共识陷阱

生成器-验证器协同进化框架

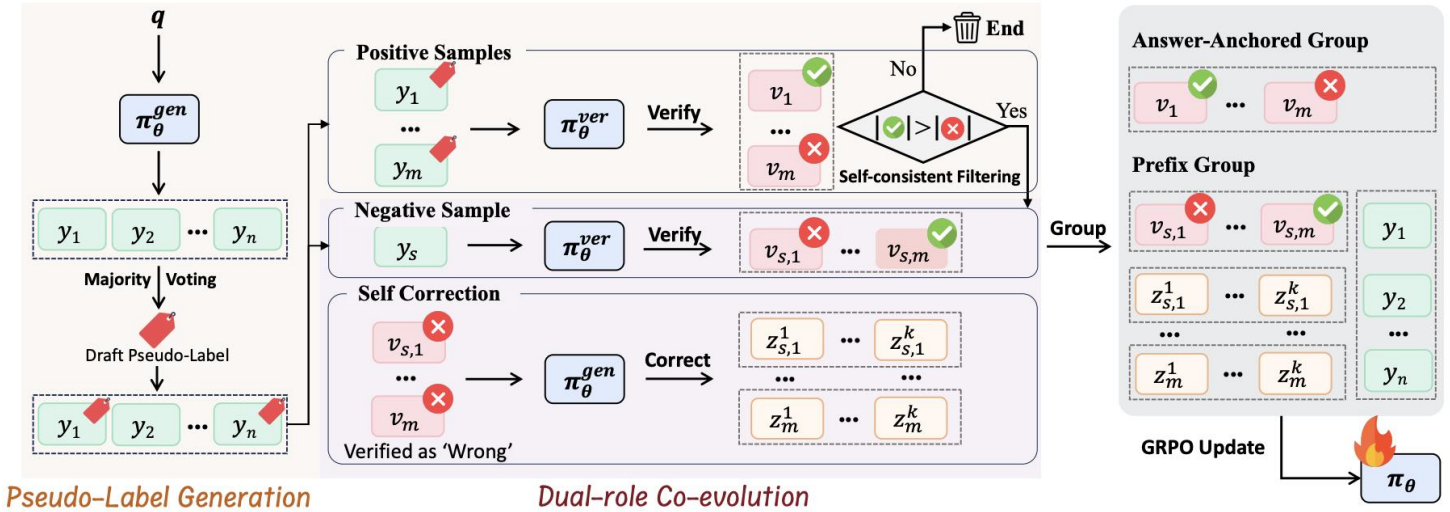
» 同一模型分饰 Generator + Verifier 两角
» 两角共享参数、共同进步 — 没有外部教师
» 协同（非对抗）：两角目标一致 — 把答案做对

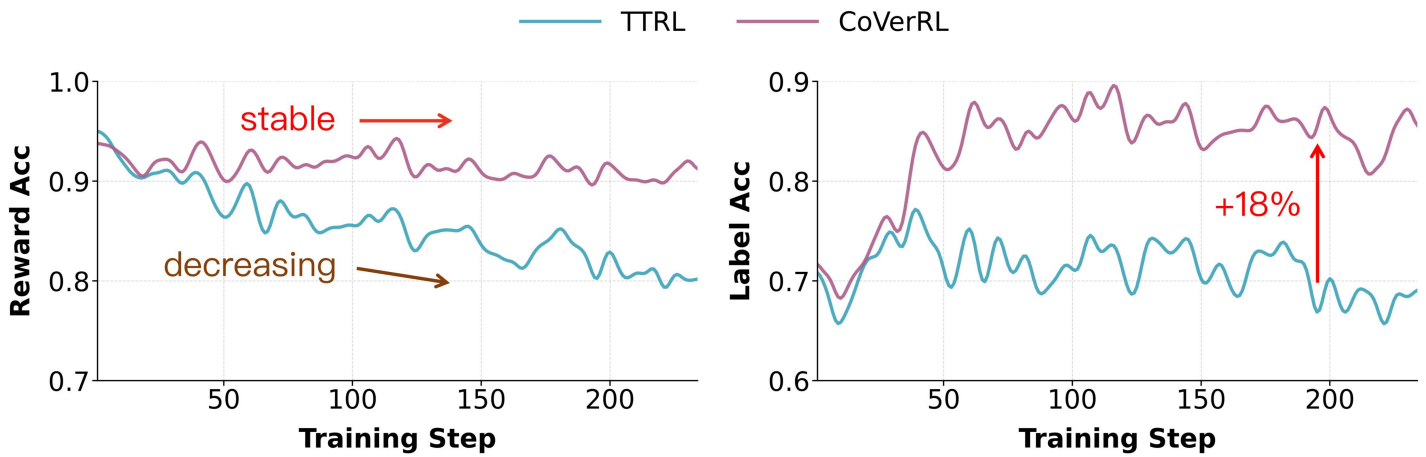
解决的核心问题

» 多数投票伪标签 → 强化系统性错误（共识陷阱）
» 用 self-verification score 替代多数投票伪奖励
» Verifier 的进步反过来打磨 Generator

结果

» vs label-free baseline: +4.7 ~ +5.9 pts
» 自验证准确率: 55% → 85%





[Pan, Yan, Wang, ..., Shen. CoVerRL: Breaking the Consensus Trap in Label-Free Reasoning via Generator-Verifier Co-Evolution. ACL 2026 Main]

3.3.6 Hermes Self-Evolution — 群体式技能进化

Evolutionary self-improvement for Hermes Agent — DSPy + GEPA · ICLR 2026 Oral

进化对象

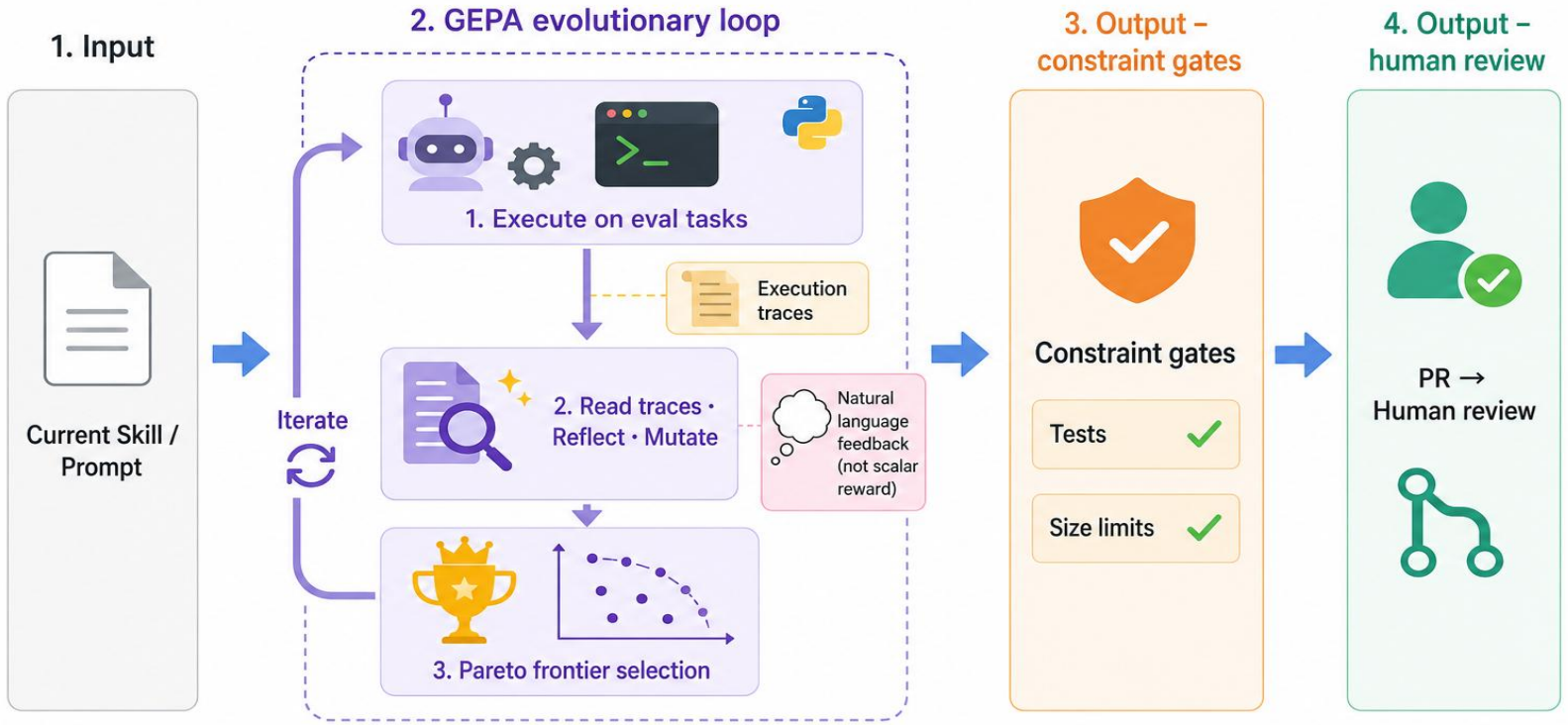
Skills · Tool descriptions
System prompts · Code

GEPA 算法

» 读取完整 trace 诊断失败原因
» 自然语言反思定向改进
（非标量reward）
» Pareto 选优，保持多样性

效果

Agent 积累 20+ 自创技能
同类研究任务完成时间减少 40%
全程 API + PR 流程，无 GPU 训练



[Nous Research. hermes-agent-self-evolution. ICLR 2026 Oral]

3.4 Part 03 — 智能体核心技术小结

六大关键技术 — 从单点能力到协同栈

01	**工具学习**	调用 → 选择 → 组合 — 自纠错让工具调用从模仿走向自主
02	**长程规划**	里程碑分解 + 树搜索 + World Model — 把长跑拆成接力
03	**记忆机制**	从外挂 RAG 到模型内化策略 — 让记忆成为可学习能力
04	**心智理论**	信念 / 意图 / 用户建模 — multi-agent 协作的认知前提
05	**Agent Skills**	从 prompt 加载到训练时内化 — "做过" 沉淀为 "会做"
06	**自我进化**	纯自博弈 / 对抗 / 协同 / 群体进化 — 四种范式同步发展

PART 04

前沿应用：四大方向

Code Agent · Deep Research · GUI Agent · Embodied Agent

4.1 Code Agent — Agentic AI 最早工业化的领域

从 GitHub Copilot 到 Devin / Cursor / Claude Code — 三年间走完一个产业周期

任务

做什么

» Issue 修复
» 多文件重构
» 全栈应用搭建
» 长程功能开发

技术

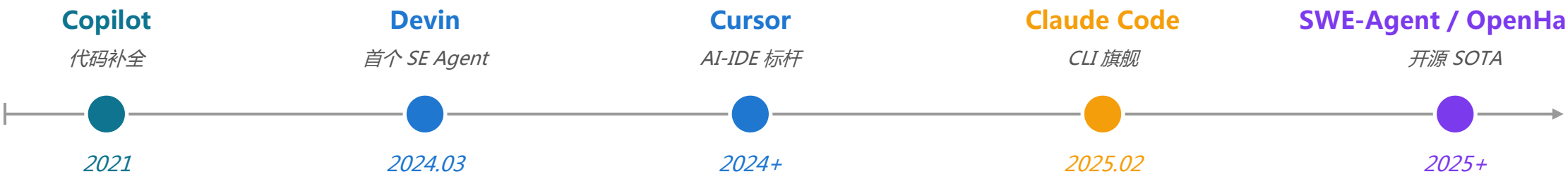
怎么做

» 仓库级语义理解
» Plan-Execute-Reflect
» Test-driven RL
» Subagents · Routines · Hooks

评测

比什么

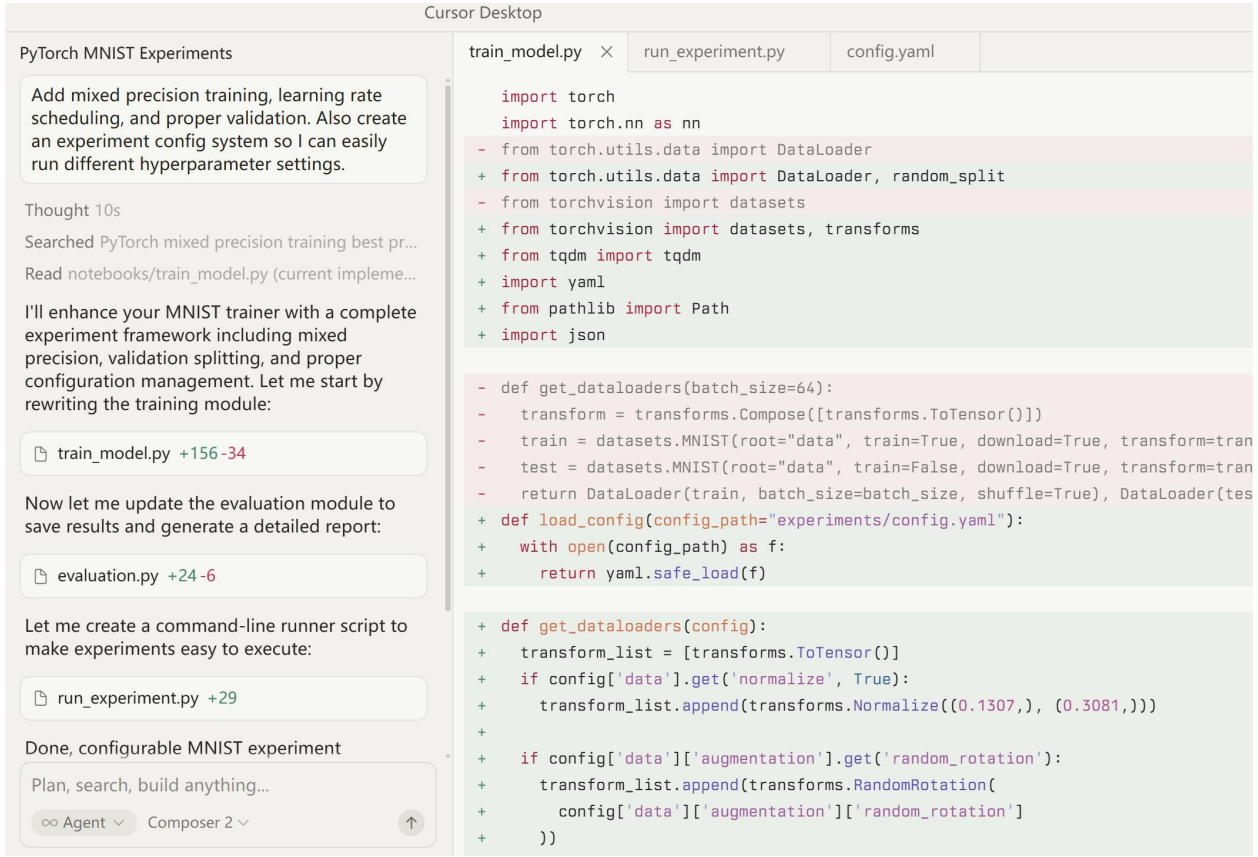
» SWE-bench / Verified
» SWE-Pro / VIBE-Pro
» Terminal Bench 2
» IWR-Bench (Our)



Code Agent 演化 · 从 IDE 插件到原生 CLI Agent · 从工具到工程师

4.1 Cursor — AI-native IDE 标杆

Anysphere · 2024-2026 · 从 autocomplete 到 agentic refactoring



三大核心交互

» Tab — 智能补全

» Cmd-K — 行内编辑

» Composer / Agent — 多文件协调

技术亮点

» 仓库级语义检索 + AST 理解

» 自训 cursor-tab 模型

» 与 Claude / GPT 后端无缝切换

代表意义

把 Code Agent 从 demo 推到千万开发者日常工作流

[Anysphere · cursor.com · 2024-2026]

4.1 Devin — 首个自主软件工程师

Cognition AI · 2024/03 · SWE-bench 引爆现象

核心定位

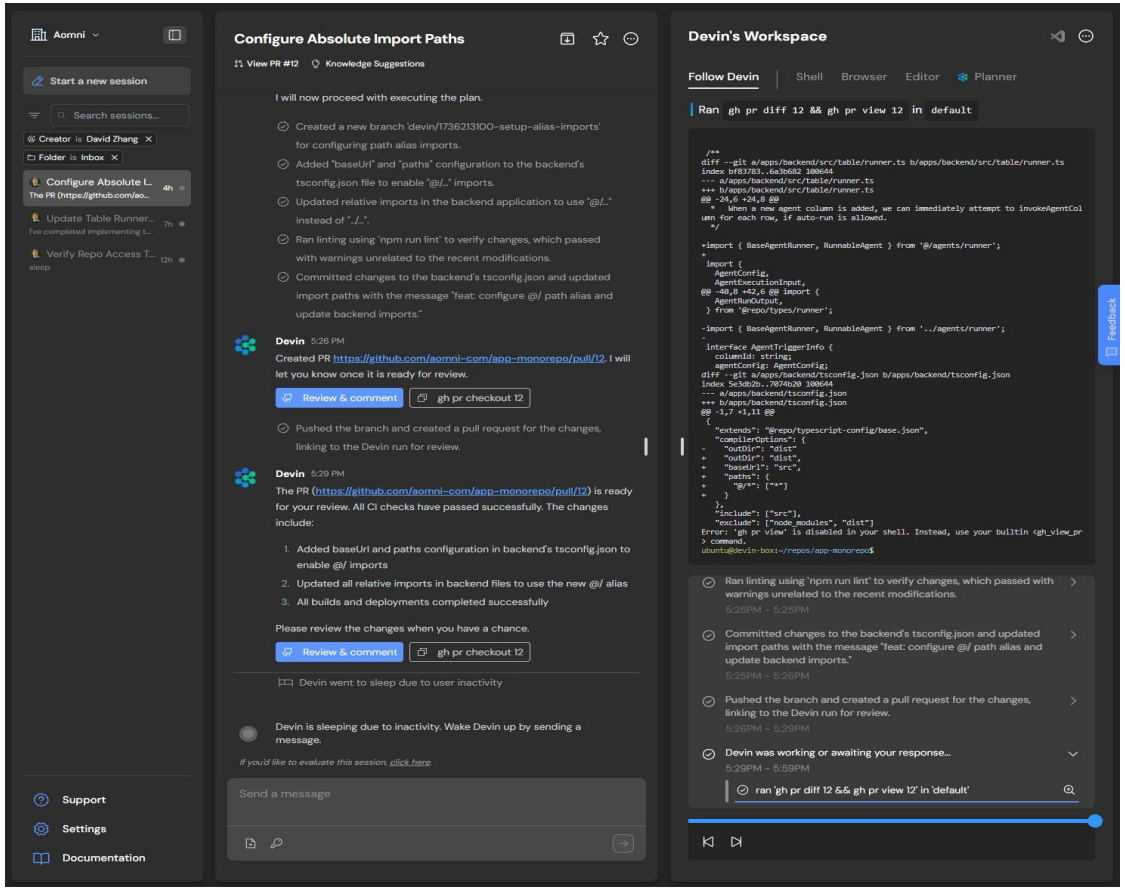
不是 IDE 内的助手 —
而是端到端代办软件工程任务的 agent

Agent 能力栈

» 自主 plan + replan
» Shell + Editor + Browser 三件套
» Long-horizon execution (小时级)
» 可视化轨迹回放 + 中断接管

代表意义

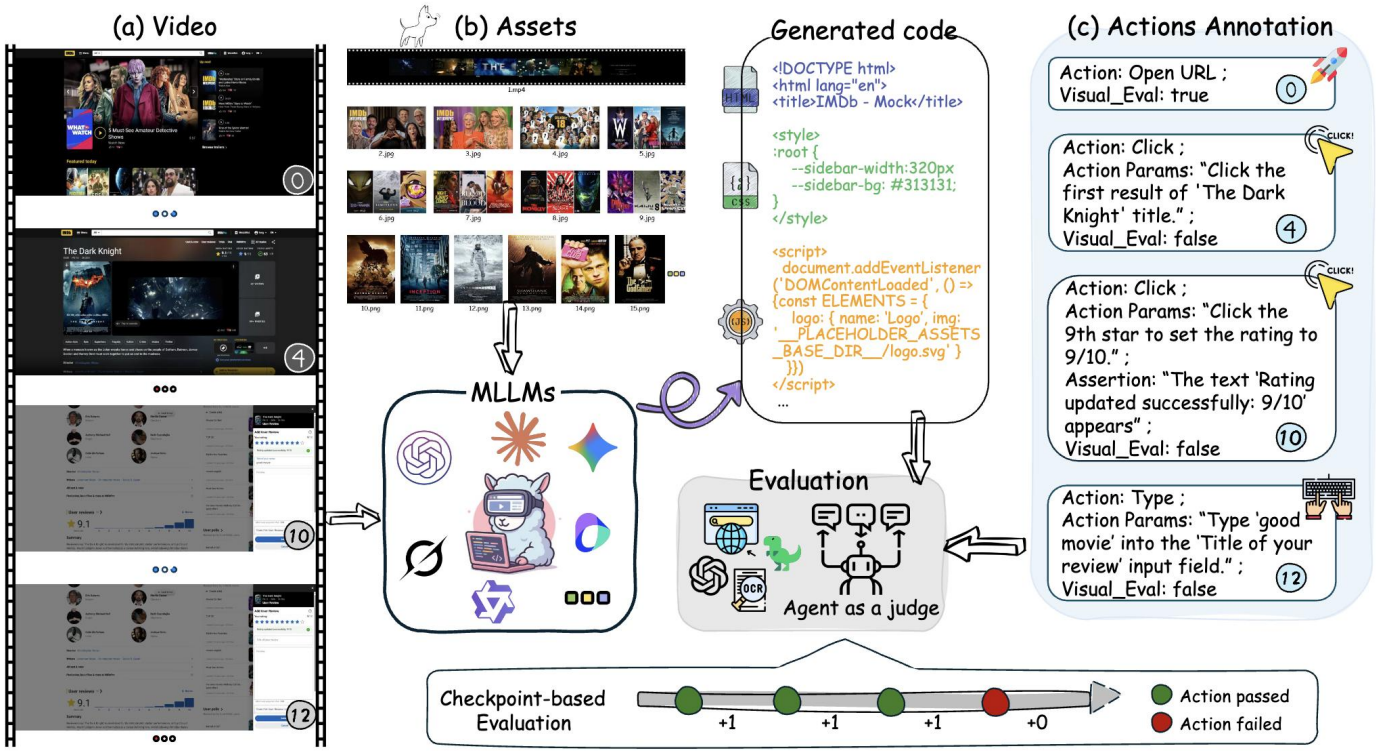
重新定义 "coding agent" 形态 —
之后 SWE-Agent / OpenHands 跟进开源



[Cognition AI. Introducing Devin, the first AI software engineer. 2024]

4.1 IWR-Bench — 视频到可交互网页

Can LVLMs Reconstruct Interactive Webpage from a User Interaction Video?



首个交互式网页重建 benchmark

» 113 任务 / 1001 actions

» 100 个真实网站

» Agent-as-a-Judge 评测

通过对28个LVLM系统测评，发现：

最佳模型总分36.35%，交互功能分数仅24.39%，视觉保真度分数为64.25%；

功能实现是最大瓶颈，所有模型VFS都显著高于IFS

通用多模态模型显著优于“视频专长”模型

[Chen, ..., Shen, Qiao, Shi. ICLR 2026]

4.2 Deep Research Agent

把数小时的深度研究压缩成一次 agent 调用

Pipeline

工作流

» Plan — 拆解研究问题
» Search — 多源检索
» Read — 长文阅读
» Synthesize — 综合归纳
» Cite & Write — 结构化写作

训练

新范式

» 长轨迹 SFT
» 偏好优化
» Search-RL
» Note-driven RL
» Process Reward Model

代表系统

工业版图

» OpenAI Deep Research
» Gemini Deep Research
» Perplexity DR · ChatGPT Agent
» Manus / Devin 变体
» 开源: WebDancer / Search-R1 / MiroFlow

瓶颈不在搜索引擎，而在怎么读 · 怎么记 · 怎么写

4.2 OpenAI Deep Research

OpenAI · 2025/02 · 首个商业化 Deep Research 产品

Pipeline

Plan → Search/Read → Verify → Synthesize → Report

关键能力

» 自主规划研究路径

» 自适应搜索深度 (5-30 分钟)

» 多源证据交叉验证

» 引用透明 / 段落级溯源

» 结构化报告输出

[OpenAI. Introducing Deep Research. 2025/02]

4.2 EviNote-RAG — 边调研变记笔记

OUR WORK

Enhancing RAG Models via Answer-Supportive Evidence Notes

RAG 的两个困境

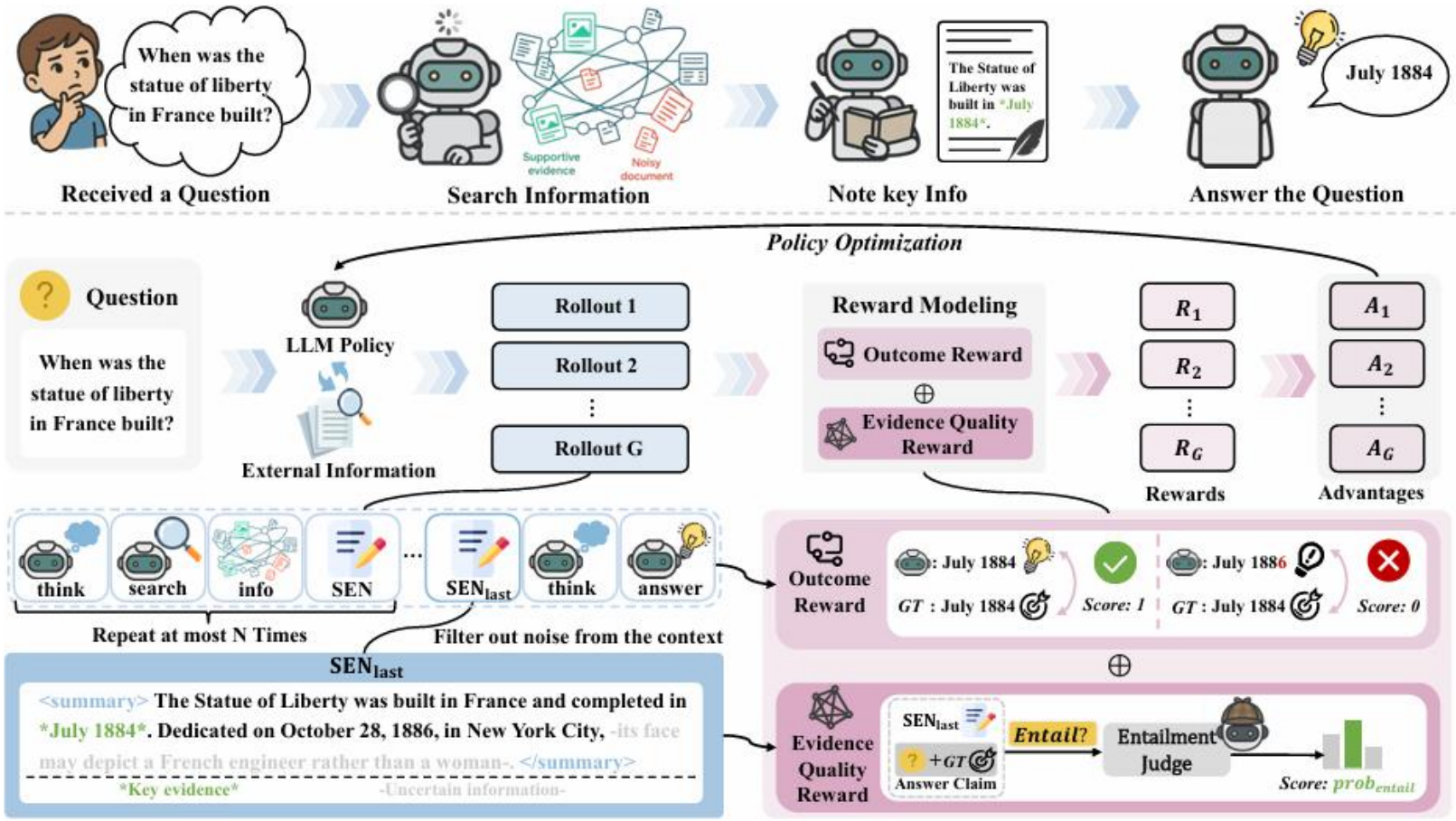
» 低信噪比 — 关键信息被淹没
» 多跳推理中错误累积

retrieve-note-answer 工作流

» SEN：提炼答案相关证据
» EQR：判断笔记是否支撑答案
» RL：优化做“笔记”策略

相对 Search-R1 的 F1 提升：

HotpotQA	+20%
Bamboogle	+40%
2Wiki	+91%



[Dai, Wang, ..., Shen, Lu. EMNL2026 under review]

4.3 GUI Agent

从 HTML DOM 到屏幕截图 · 从浏览器到操作系统，VLM 真正走出对话框

形态

做什么

» 浏览器自动化
» 桌面跨应用任务
» 移动端真机操控
» 跨设备协同

技术

怎么做

» Visual Grounding
» Long-horizon planning
» Test-time RL
» Memory + Tool integration

瓶颈

卡在哪

» 高密度 UI 上的 grounding 漂移
» 长程任务状态追踪与回退
» Real-world 分布漂移 + 反检测
» 数据 / 评测 / 反馈闭环未成型
» 推理成本 vs 实时响应权衡

闭环 — 状态更新触发新一轮截图

4.3 UI-TARS — 端到端 GUI Agent

UI-TARS: Pioneering Automated GUI Interaction with Native Agents

核心架构

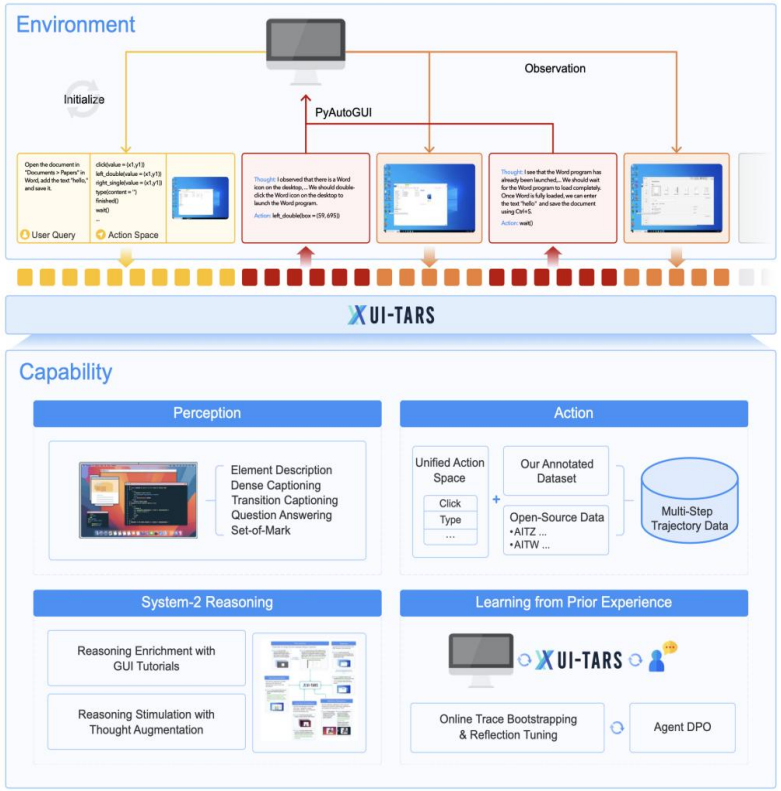
纯视觉端到端 — 截图 → 推理 → 动作
无需 HTML / a11y tree 辅助

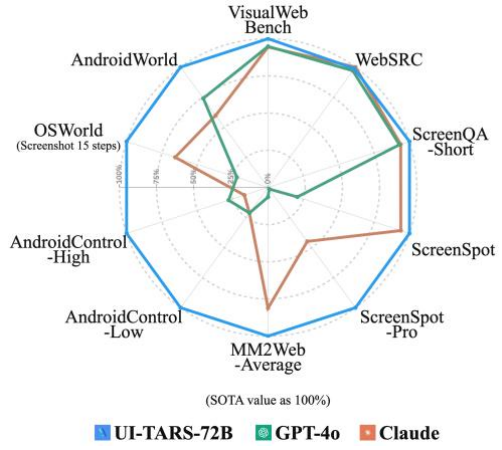
三大关键能力

» Visual Grounding — 看到 → 点对
» 长思考 + 短反应混合
» Reflection 自纠错
» System-2 慢思考 + System-1 快反射

结果

OSWorld： 在 50 步得分 24.6，15 步得分 22.7，超过了 Claude（ 22.0 和 14.9）。
AndroidWorld： 达到 46.6，大幅超越 GPT-4o 的 34.5。





Benchmark	Previous SOTA	Relative improvement of UI-TARS	
GUI-Odyssey	OS-Atlas-7B	+42.90%	+40.32%
OSWorld (Screenshot 15 steps)	Aguvis-72B w/ GPT-4o	+33.53%	+10.00%
ScreenSpot-Pro	UGround-V1-7B	+22.51%	+14.79%
MM2Web-Website	Aguvis-72B	+12.39%	+9.20%
AndroidControl-Low	OS-Atlas-7B	+7.16%	+6.57%
MM2Web-Task	Aguvis-72B	+7.19%	+4.84%
MM2Web-Domain	Aguvis-72B	+6.70%	+3.95%
ScreenSpot-v2	OS-Atlas-7B	+3.67%	+5.17%
ScreenQA-Short	Qwen2-VL-7B	+4.36%	+3.30%
VisualWebBench	GPT-4o	+5.48%	+1.53%
AndroidControl-High	OS-Atlas-7B	+4.92%	+1.83%

0.00% 50.00% 0.00% 50.00%

UI-TARS-72B UI-TARS-7B

[Qin et al. UI-TARS: Pioneering Automated GUI Interaction with Native Agents. 字节 2025]

4.3 GUI-G² — 高斯奖励建模

Gaussian Reward Modeling for GUI Grounding

动机 — 二元命中奖励太稀疏

» 传统 RL 奖励只有 hit / miss，反馈过于稀疏

» 人类点击天然围绕目标呈高斯分布

GUI Gaussian Grounding Reward

» Gaussian point reward

» Coverage reward

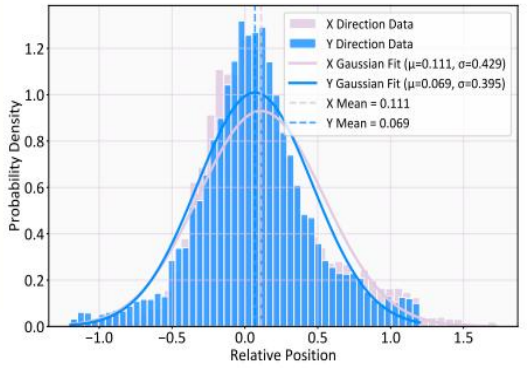
» Adaptive variance 处理多尺度元素

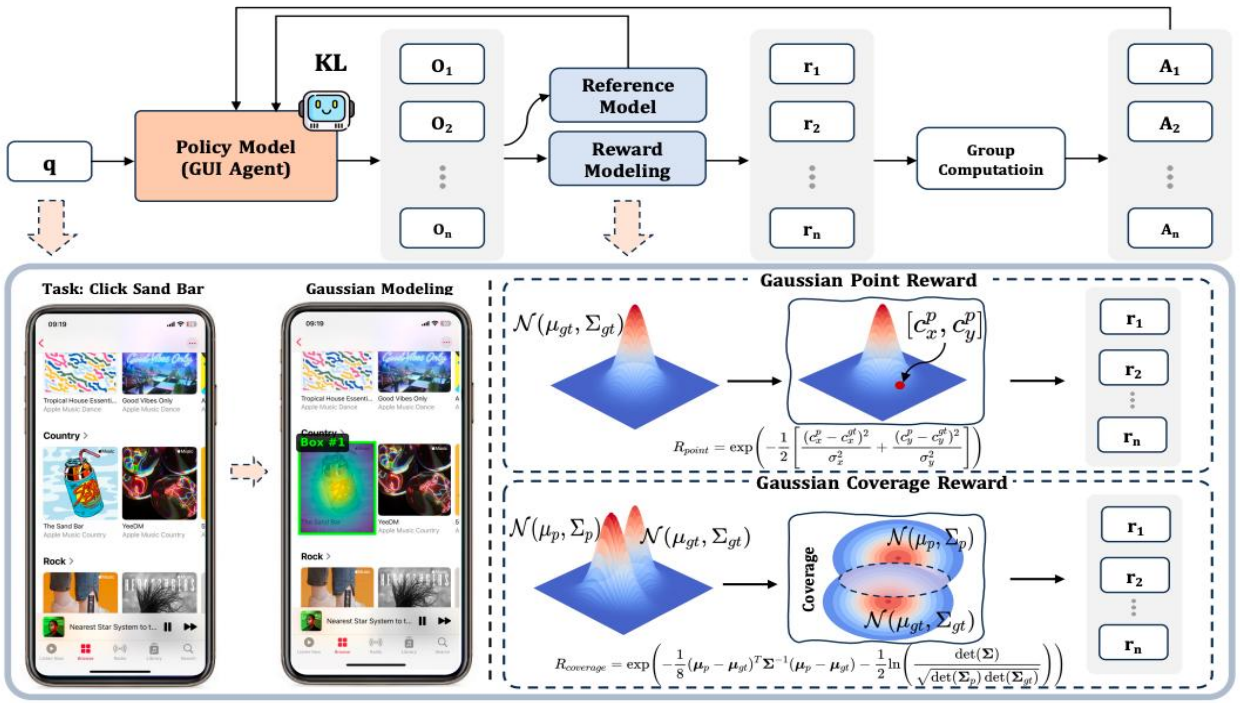
ScreenSpot-Pro: +9.4%

(vs UI-TARS-72B, Bytedance Seed)

ScreenSpot-V2: +3.0%

(vs UI-TARS-72B, Bytedance Seed)





[Tang, Gu, ..., Shen, Lu. AAAI 2026]

4.3 UI-S1 — Semi-online RL

Advancing GUI Automation via Semi-online Reinforcement Learning

Offline RL（离线强化学习）

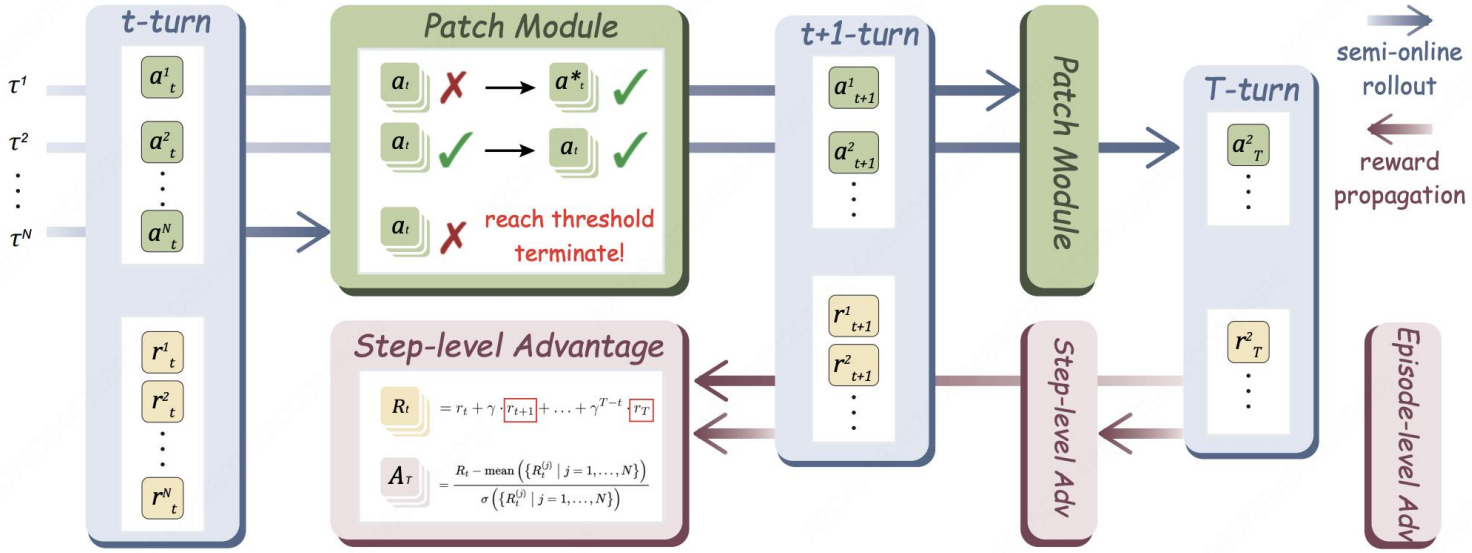
缺乏轨迹奖励，训练时交互历史离线（off-policy）构造，造成多轮推理时训推不一致

Online RL（在线强化学习）

部署成本高，同时缺乏细粒度的奖励

Semi-online RL（半在线强化学习）

1. 在 offline 轨迹上模拟 online RL，保留多轮推理过程
2. 提出**Patch Module** 修复Rollout偏差
3. 引入未来折扣奖励（考虑当前步骤对未来的影响）
4. 双层优势估计：加权步骤级别和轨迹级别优势



结果：在动态GUI benchmark上
AndroidWorld: +12.0%
AITW: +23.8%

[Lu, Ye, Tang, Shen et al. UI-S1: Advancing GUI Automation via Semi-online Reinforcement Learning. ACL 2026]

4.3 GUI-RC / GUI-RCPO — Test-time RL

OUR WORK

Test-time Reinforcement Learning for GUI Grounding via Region Consistency

问题背景

现有 GUI grounding 方法依赖大量人工标注，
难以泛化至多样化新场景

核心观察

模型对同一 GUI 元素进行多次采样时，
多次预测的空间一致性 = 隐式置信度信号

GUI-RC

» 空间投票提取模型共识区域

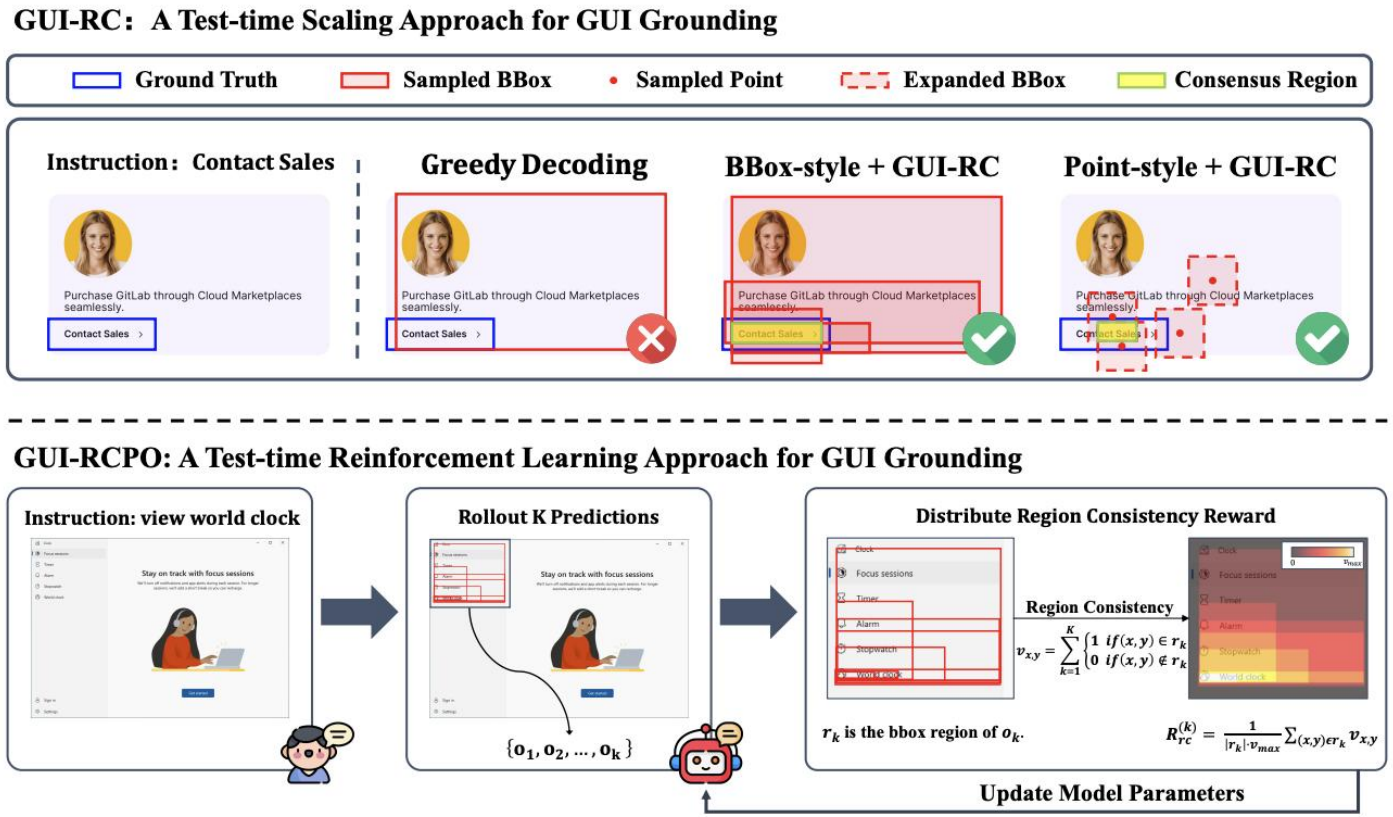
» training-free: +2-3%

GUI-RCPO

» 空间一致性转化为自监督奖励信号

» 1272 条无标注样本: +3-6%

» 低成本泛化至新场景

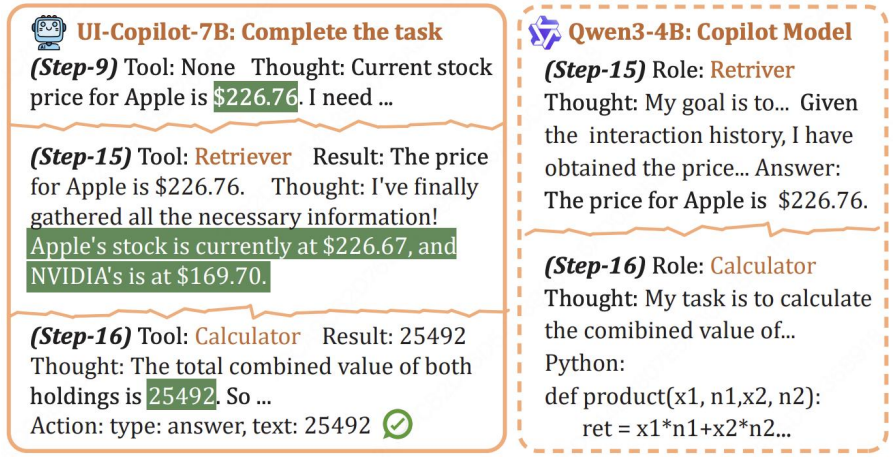


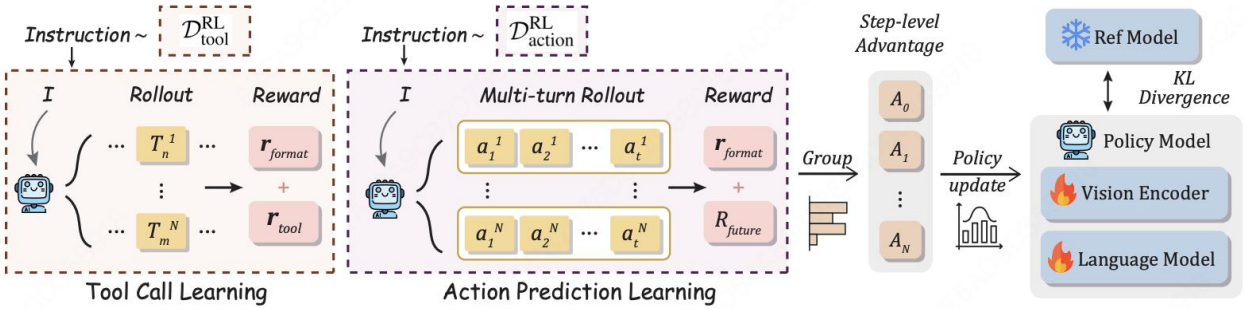
[Du, Yan, Tang, Lu, Zong, Lu, Jiang, Shen. AAAI 2026]

4.3 UI-Copilot — 长程 GUI 自动化

OUR WORK

Long-Horizon GUI Automation via Tool-Integrated Policy Optimization





长程 GUI 三大问题

» 记忆衰退：上下文过载使得注意力分散
» 进度混乱：agent遗忘当前进度，反复执行或顺序错乱
» 数学幻觉：模型计算和代码能力欠缺

我们的方案：

1. **记忆解耦**：上下文仅维护任务进度，将记忆类任务信息存放在本地，按需调取
2. **角色分工**：GUI Agent作为规划和执行者，同时搭配一个轻量级的助手模型进行数学计算，以及记忆调度等复杂任务
3. **TIPO**：通过强化学习同时训练GUI Agent调用助手模型和任务执行的能力

结果：

AndroidWorld +17.1%

长程记忆Benchmark（MemGUI-Bench）上SOTA

[Lu, Tang, ..., Shen. UI-Copilot: Advancing Long-Horizon GUI Automation via Tool-Integrated Policy Optimization. ACL 2026]

4.3 ClawGUI — GUI Agent 全栈框架

ZJU-REAL · 构建于开源 OpenClaw / nanobot 之上 · 跨设备真机部署

ClawGUI-Agent (OpenClawGUI)

» nanobot 驱动 GUI & CLI 操控

» Android / HarmonyOS / iOS 真机

» 12+ 聊天平台远程操控

ClawGUI-RL

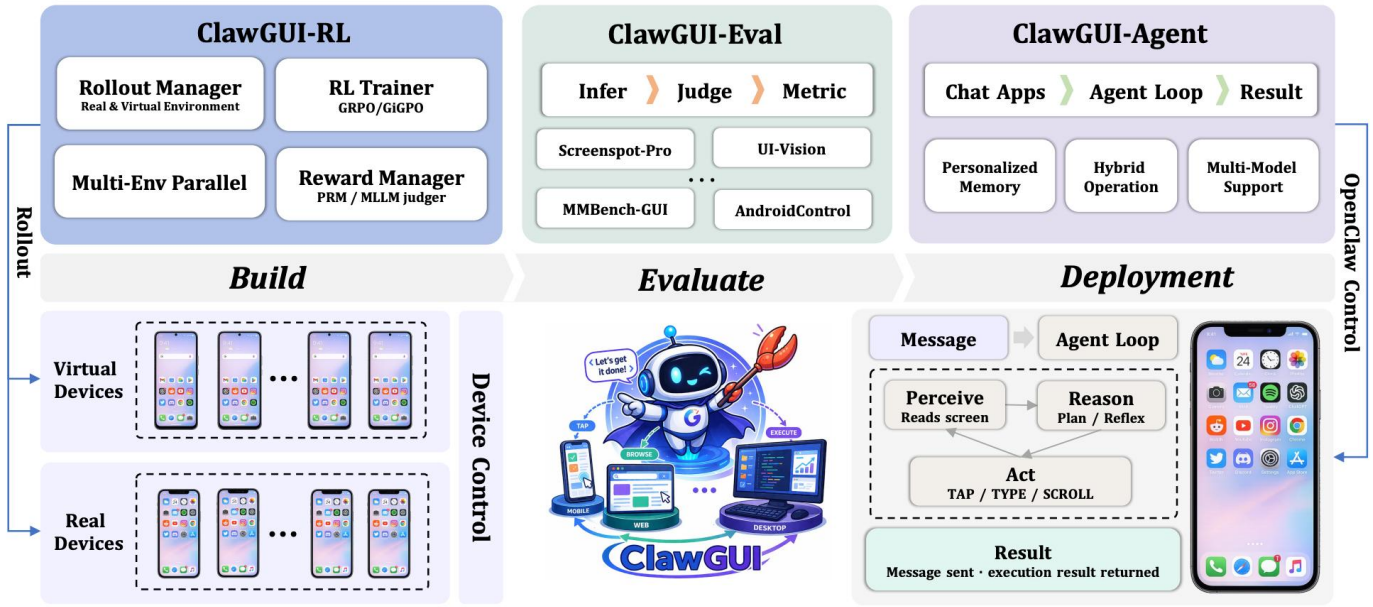
» GiGPO + PRM · 128 并发实例

» 兼容 GRPO/GSPO/DAPO/GiGPO

» 稳定训练 60 step +

ClawGUI-Eval

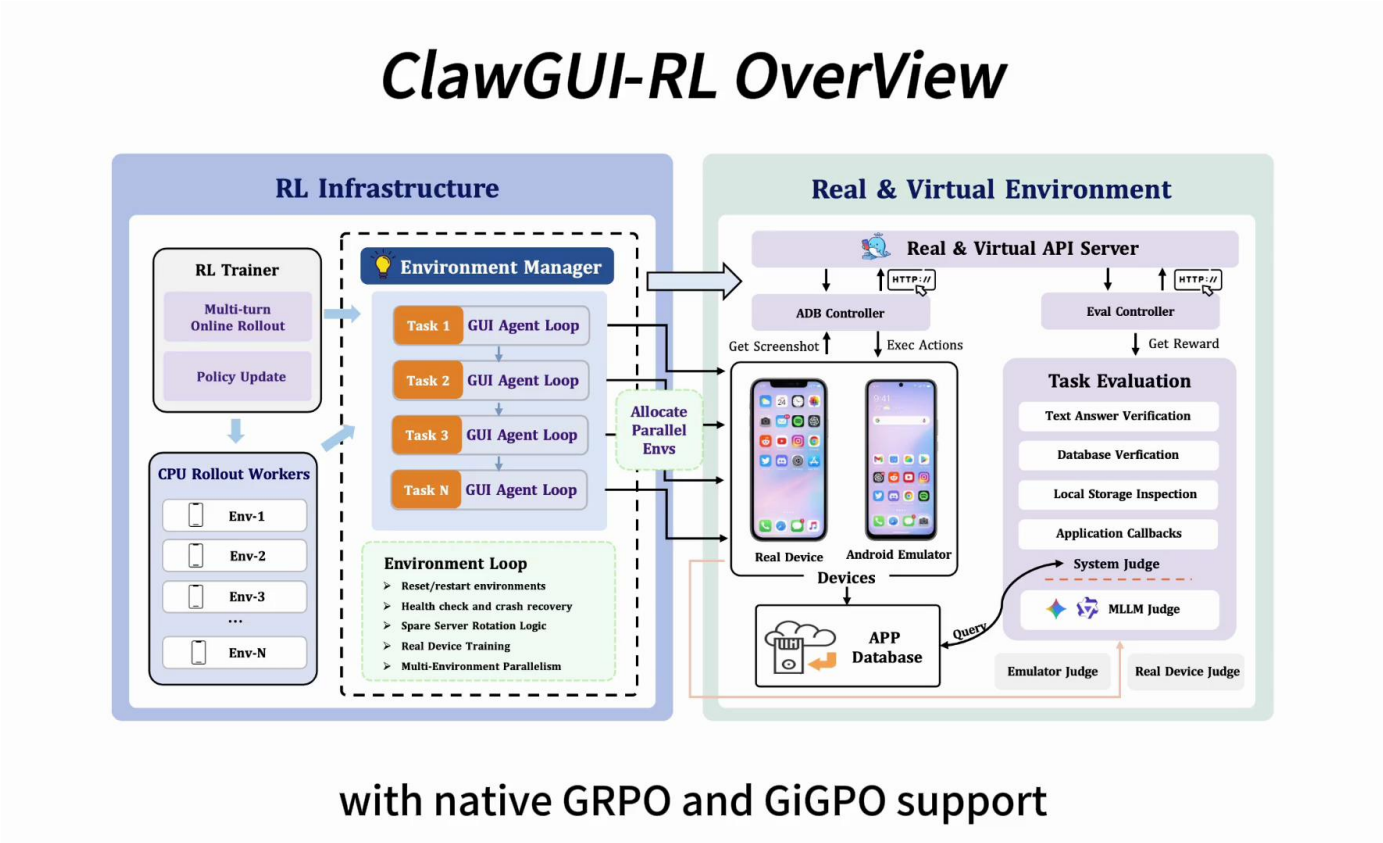
» 6 benchmark / 11+ 模型 · 95.8% 复现率



github.com/ZJU-REAL/ClawGUI

4.3 ClawGUI-RL — Online RL Demo

Live Demo — 真机环境中的 Online Agentic RL 训练全过程



1. Online Rollout — 128 并发 Android 模拟器 / 真机同步采样轨迹
2. GiGPO + PRM — 双层组优势 + 步级奖励，长程信用分配
3. 实时反馈 — reward 曲线 + agent 行为可视化 + 稳定训练 60 steps +

4.4 Embodied Agent — 从数字到物理世界

具身智能不是把 LLM 装进机器人 — 而是先教会它理解空间、视角与因果

定义

在物理或仿真环境中，具备感知 / 推理 / 行动闭环的智能体

从数字到物理的关键意义

» 数字 agent 走通后的下一站
» 让 AI 真正影响真实世界

核心挑战

» Sim-to-real 迁移
» 数据稀缺 / 安全约束
» 跨形态 / 跨任务泛化

四大关键能力

空间感知

Spatial Perception

多视角 · 第三人称 · 3D 几何

长程操作

Long-horizon Manipulation

数十步连续交互 · 错误恢复

灵巧控制

Dexterous Control

高频闭环 · 触觉反馈

因果理解

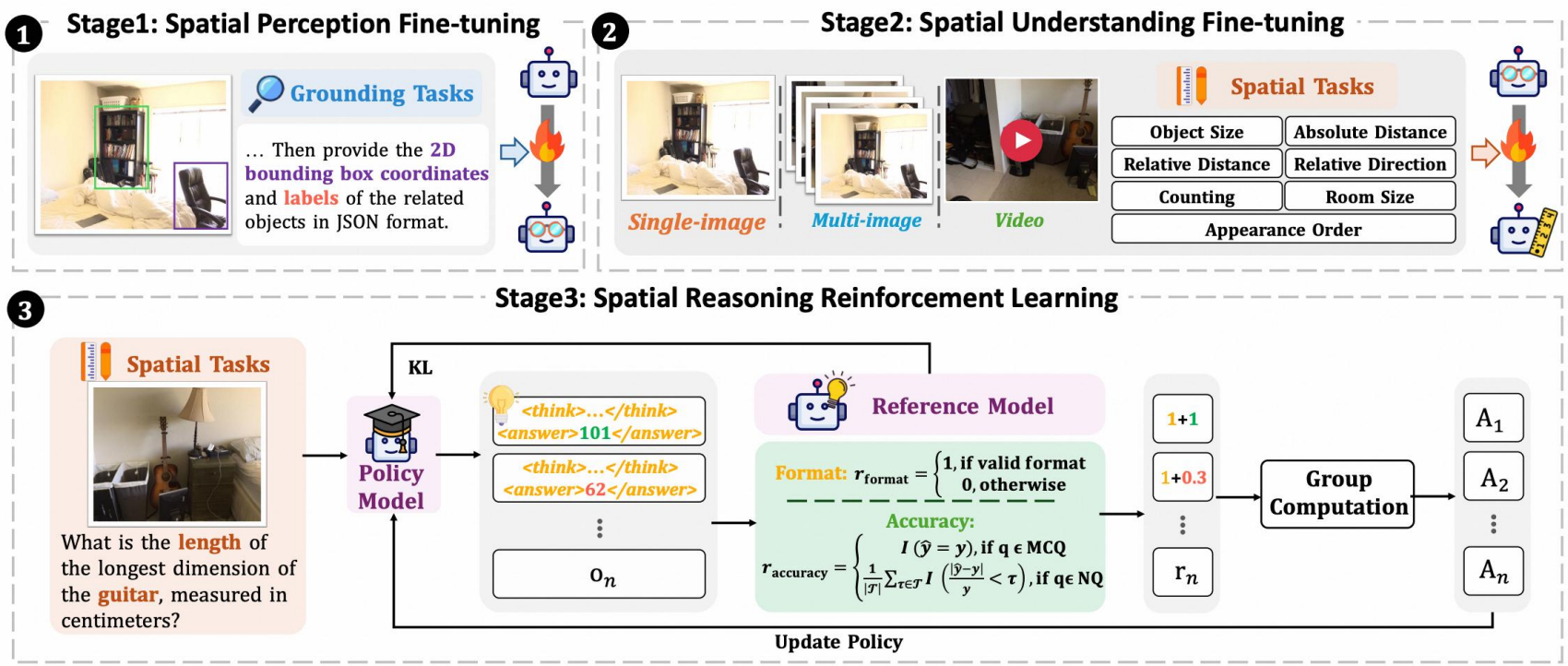
Causal Reasoning

物理因果 · 反事实推理

4.4 SpatialLadder — 渐进式空间智能训练

OUR WORK

SpatialLadder: Progressive Training for Spatial Reasoning in VLMs



关键洞察

VLM 缺少分层 "perception → understanding → reasoning" 训练

三阶段渐进训练

1. Object localization (感知)
2. Multi-dimensional 空间任务 (理解)
3. RL with verifiable rewards (推理)

实验结果

» 26.6k 多模态训练样本

» **3B 模型超 GPT-4o +20.8%**

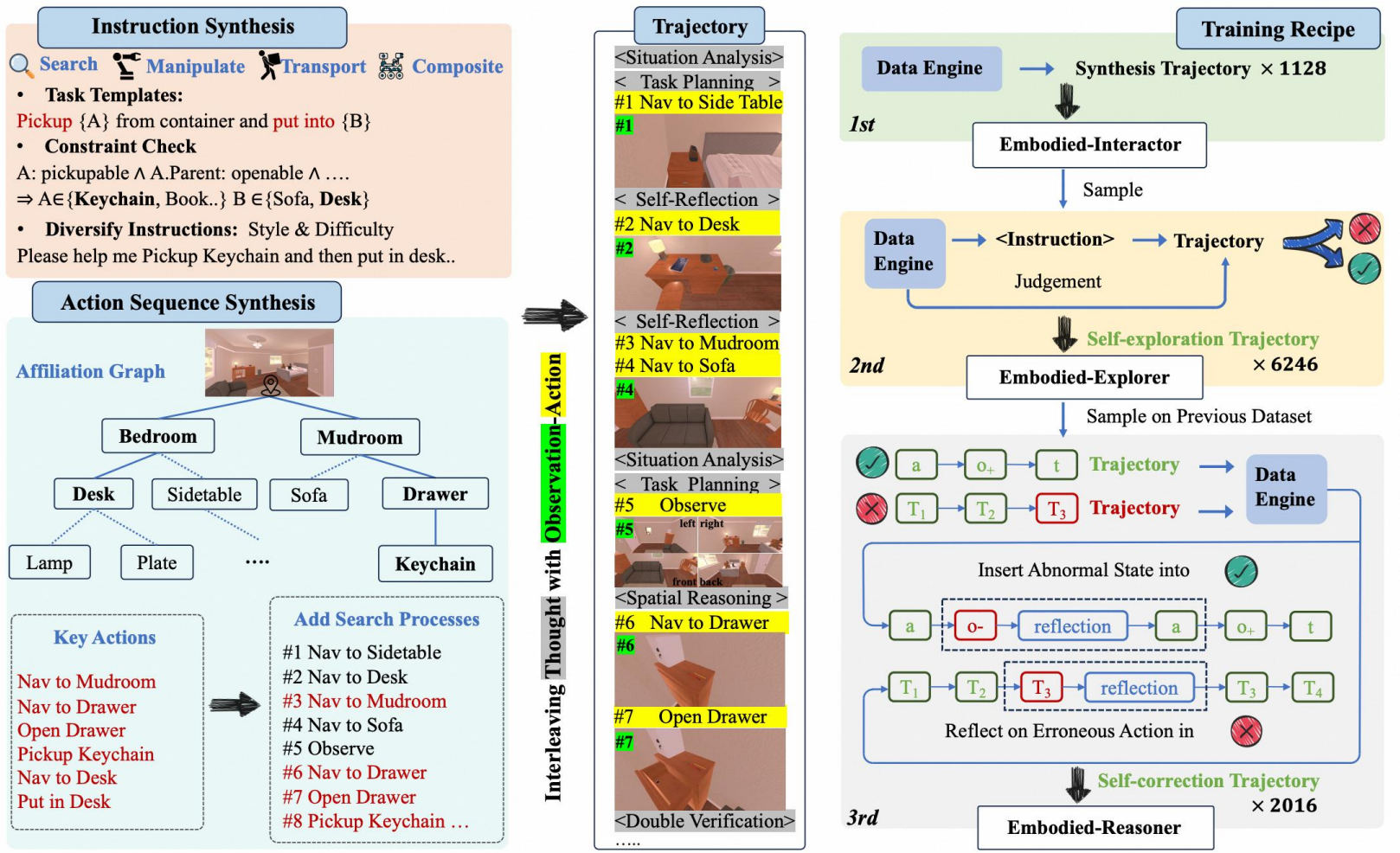
» **OOD 提升 +7.2%**

[Li, Li, Wang, Yan, ..., Shen et al. ICLR 2026 — arXiv:2510.08531]

4.4 Embodied-Reasoner — 长推理 → 具身

OUR WORK

Synergizing Visual Search, Reasoning, and Action for Embodied Interactive Tasks



问题动机

深度思维模型在图文交错的
具身交互轨迹中缺乏探索

数据合成

9.3k 轨迹 · 64k 图像 · 90k 思考

三阶段训练

» 模仿学习
» 拒绝采样自探索
» 反思自纠

vs OpenAI o1: +9%
vs o3-mini: +24%
vs Claude-3.7: +13%

[Zhang, Wang, ..., Shen, Hou, Zheng, ..., Zhuang. ACL 2026]

4.4 Embodied-Reasoner — 长推理 → 具身

Synergizing Visual Search, Reasoning, and Action for Embodied Interactive Tasks

真实物理世界实验

- 真实世界的交互任务
- 厨房、浴室、卧室
- 30 个任务

Model	Success Rate (%)
Qwen2.5-VL-72B-Instruct	43.3
OpenAI o1	50.0
OpenAI o3-mini	44.0
Embodied-Reasoner	56.7

[Zhang, Wang, ..., Shen, Hou, Zheng, ..., Zhuang. ACL 2026]

4.6 Part 04 — 前沿应用小结

01	**Code Agent**	最早工业化 — 从 Copilot 到 Cursor / Devin / Claude Code 三年走完产业周期
02	**Deep Research**	深度研究自动化 — Plan → Search → Read → Synthesize → Write 全 pipeline
03	**GUI Agent**	VLM 走出对话框 — Visual Grounding + 长程规划 + 真机部署
04	**Embodied Agent**	数字 → 物理 — ViewSpatial → SpatialLadder → Embodied-Reasoner 三步走

PART 05

展望：通往通用智能体

四大未来方向 · 总结

5.1 Conclusions from this tutorial

从 Agent 核心能力的技术发展

01 "推理" 被重新定义

不再是离线 CoT，而是 Reasoning ⊕ Acting ⊕ Reflection 的统一过程

02 "数据" 被重新定义

智能体Trajectory 取代交互对话，Verifiable Environment 取代 Human Preference

03 "训练" 被重新定义

从 offline SFT/RL → online Agentic RL · 算法 / 数据 / 基础设施 三者 co-design

04 "记忆 • 技能" 成为 first-class 能力

不再是 RAG 外挂 / prompt 加载 ，而是被训练进权重的可学习策略；Agent 越用越聪明从可能变现实

05 "进化" 从一次性训练变成持续过程

纯自博弈 / 对抗式 / 协同式 / 群体式 进化，Agent 闭环开始自启动

5.2 未来方向 1 · Agent 的环境

环境是 Agent 训练的天花板 · 从手工 gym 到自动合成，再到 World Model 生成式环境

agent ≈ policy × environment · 环境的丰富度与可验证性决定能力上限

Real Env	Hand-crafted	LLM-synth	Generative
真实世界	代码 gym / sim	自动合成环境	World Model
采集 / 部署成本高	*WebArena · TAU · SWE-Gym*	*AutoEnv · AWM · AgentGym-RL*	*Genie · Cosmos · V-JEPA 2*

评测 Benchmark

AgentBench · WebArena · TAU-Bench · AppWorld · SWE-bench Verified

RL Gym · 训练环境框架

AgentGym-RL · GEM (Gym for Agentic LLMs) · OpenEnv (Meta)

自动合成环境

Agent World Model · AutoEnv · Agent-World · SWE-Smith

World Model · 生成式环境

Genie 3 (DeepMind) · Cosmos (NVIDIA) · V-JEPA 2 (Meta) · DreamerV3

可验证 reward · 可重置 · 可大规模生成 · 与真实分布对齐

Trend · 环境正从 "评测平台" 演变为 Agent 训练数据生成器

5.2 未来方向 2 · 终身自我进化

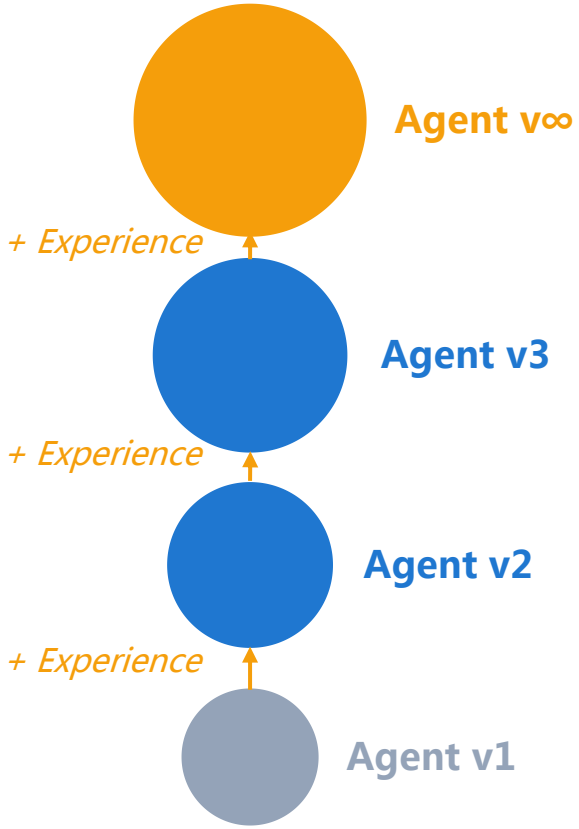
从一次性预训练到持续在线进化 — agent 越用越聪明

» 三件套
技能积累 + 自反思 + 自动课程

» 当前雏形
Self-Rewarding · AlphaEvolve · Code-A1 · CoVerRL · CheckMate · Skill0 · SpatialEvo

» 关键挑战
Reward hacking · 价值漂移 · 持续监督 — 静态安全训练已经追不上

» 风向标问题
参数自进化（让模型变聪明） vs **Harness 自进化（让 scaffold 更好用）**，
哪一侧才是 agent 真正的进化主轴？



5.2 未来方向 3 · 从单体到群体

Single Agent → Society of Agents — 群体智能是下一片新大陆

» 单 agent 的瓶颈

Context 上限 · 知识广度 · 串行延迟

» 群体 agent 的红利

并行探索 · 任务分工 · 同伴验证

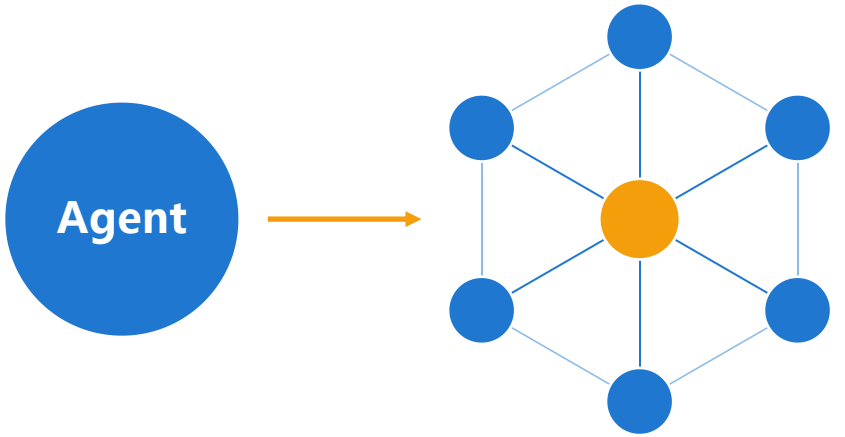
» 实证信号

Anthropic Multi-Agent: +90% over single Opus

Kimi K2.6 Agent Swarm: 300 sub-agents · 4000 步协同

» 开放问题

通信协议 · 跨 agent 信用分配 · 涌现规范



Society of Agents

5.2 未来方向 4 • CLI + GUI → Embodied

一个争议、一个共识、一个长期愿景

争议 **CLI or GUI ?**

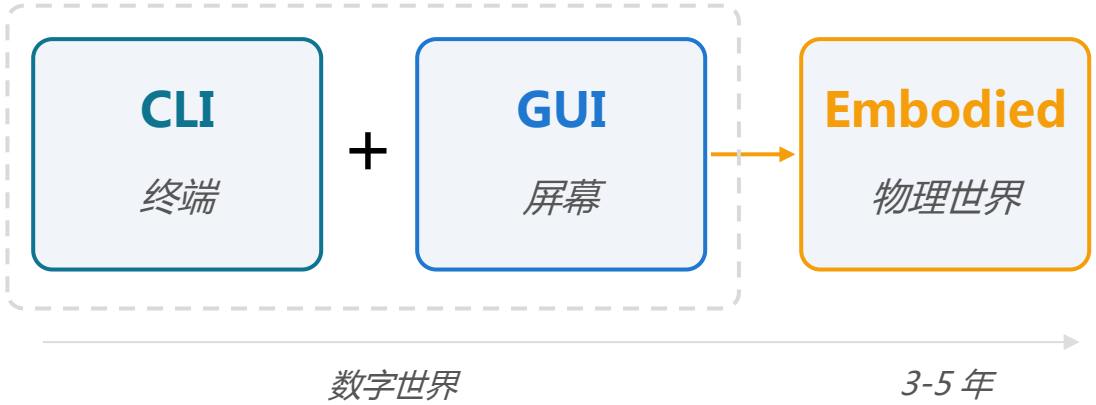
Claude Code 走纯 CLI · UI-TARS / 豆包手机 走纯 GUI

共识 **CLI + GUI**

终端的精确性 + GUI 的通用性 — ClawGUI / Codex 已验证

未来 **Embodied Agent**

数字世界跑通后 → 物理世界 — VLA + 世界模型



通往 AGI 的现实路径

5.3 总结

能思考 → 会行动 → 持续学习 → 共同协作 → 改造世界

我们正站在 AI 从"懂语言"到"懂世界"再到"改世界"的第二次跨越的起点

致谢 · Acknowledgments

浙江大学 REAL Lab · Reasoning Embodied Agentic and Learnable AI

实验室主页：https://zju-real.github.io

Reasoning

认知基础


颜聿辰


徐皓雷


张航

Agentic RL

训练范式


王子轩


潘腾


张锴文

Agent Core

核心能力


卢政希


王子轩


邱怡文


杜庸

Applications

应用前沿


李弘幸


李鼎铭


唐飞


董欣谕

本 Tutorial 凝聚了实验室同学们的研究工作与一线思考。特此致谢！

参考文献 · References (1/2)

按章节顺序 · 我们组工作以斜体标注

§1 推理 · Reasoning

[1] Wei et al. Chain-of-Thought Prompting. NeurIPS 2022.
[2] Wang et al. Self-Consistency. ICLR 2023.
[3] Yao et al. Tree of Thoughts. NeurIPS 2023.
[4] Besta et al. Graph of Thoughts. AAAI 2024.
[5] Shao et al. DeepSeekMath. 2024.
[6] Yu et al. DAPO. 2025.
[7] DeepSeek-AI. DeepSeek-R1. 2025.
[8] Aggarwal & Welleck. L1. 2025.
[9] Thinking Machines. On-Policy Distillation. 2026.
[10] Xu et al. NeurIPS 2025. arXiv:2505.14684.
[11] Yan et al. ICLR 2026. arXiv:2502.11684.
[12] Wu et al. LAPO. ACL 2026. arXiv:2507.15758.
[13] Zhao et al. SBT. NeurIPS 2025. arXiv:2505.14604.
[14] Xu et al. OPD. ACL 2026. arXiv:2509.25175.
[15] Li et al. ViewSpatial-Bench. ECCV 2026.
[16] Yuan et al. GSM8K-V. ICML 2026. arXiv:2509.25160.
[17] Routing Distraction. ACL 2026 Main. arXiv:2604.08541.

§2 Agentic RL

[18] Wu et al. ARPO. 2025.
[19] Zhang et al. GiGPO. 2025.
[20] Jin et al. Search-R1. UIUC 2025.
[21] Yan et al. VerifyBench. ICLR 2026. arXiv:2505.15801.
[22] Qiu et al. GRIL. ACL 2026 Findings.

§3 Agent 核心技术 · Core Tech

[23] Shen et al. HuggingGPT. NeurIPS 2023.
[24] Anthropic. Claude Releases 2024-2026.
[25] Anthropic. Claude Code. 2025/02 RP · 2025/05 GA.
[26] Moonshot AI. Kimi K2.6 Tech Blog. 2026/04.
[27] MiniMax. M2.7: Early Echoes of Self-Evolution. 2026/04.
[28] Schick et al. Toolformer. NeurIPS 2023.
[29] Zhang, Shen et al. EMNLP 2025.
[30] Zhang et al. CheckMate. ICML 2026.
[31] Zhou et al. LATS. ICML 2024.
[32] Packer et al. MemGPT. UC Berkeley 2023.
[33] Yan et al. Memory. ICML 2026.
[34] Chen et al. ToMBench. ACL 2024.
[35] Hou et al. TimeToM. ACL Findings 2024.
[36] Hou et al. Sprinkle Cognitive Aha Moment. ACL Findings 2025.
[37] Anthropic. Skills Engineering Blog. 2025.
[38] Yao et al. SkillRL. 2026. arXiv:2602.08234.
[39] Lu et al. *SkillZero. 2026. arXiv:2604.02268.*
[40] Zhao et al. Absolute Zero. NeurIPS 2025. arXiv:2505.03335.
[41] Wang et al. *Code-A1. 2026. arXiv:2603.15611.*
[42] Pan et al. CoVerRL. ACL 2026 Main.
[43] Nous Research. Hermes Self-Evolution. ICLR 2026 Oral.
[44] DeepMind. AlphaEvolve. 2025. arXiv:2506.13131.

参考文献 · References (2/2)

§4 Agent 应用 · Applications & §5 讨论

§4.1 Code Agent

[45] Anysphere. Cursor. cursor.com · 2024-2026.
[46] Cognition AI. Devin: First AI Software Engineer. 2024.
[47] Ma, Shen et al. EMNLP 2024 Findings.
[48] Chen, Shen, Qiao, Shi et al. ICLR 2026.

§4.2 Deep Research

[49] OpenAI. Introducing Deep Research. 2025/02.
[50] Dai, Wang, ..., Shen, Lu. ACL 2026.
[51] Liu, Ma, ..., Shen, Zhang, Lu. ACL 2026.

§4.3 GUI Agent

[52] Qin et al. UI-TARS. ByteDance 2025.
[53] Tang, Gu, ..., Shen, Lu. AAAI 2026.
[54] Lu, Ye, Tang, Shen et al. arXiv:2509.11543.
[55] Du, Yan, Tang, ..., Shen. AAAI 2026.
[56] Lu, Tang, ..., Shen. ACL 2026.

§4.4 Embodied Agent

[57] Li et al. SpatialLadder. ICLR 2026. arXiv:2510.08531.
[58] ZJU-REAL. SpatialEvo. in submission.
[59] Zhang, Wang, ..., Shen et al. Embodied-Reasoner. ACL 2026.

§3.x Agent 框架 • 生态

[60] LangGraph · AutoGen · CrewAI · MetaGPT.
[61] OpenClaw. github.com/openclaw/openclaw.
[62] Nous Research. Hermes 4 Technical Report. 2025.
[63] Tang et al. ClawGUI. github.com/ZJU-REAL/ClawGUI.

§5 讨论 • 其他参考

[64] Anthropic. Claude Code Source Leak Analysis. 2026/03.
[65] Anthropic. Engineering Blog · Claude.ai/blog.
[66] Latent Space. AI News — The Claude Code Source Leak. 2026.
[67] Anthropic. Building Effective Agents. 2024.

Thank You

Q & A

沈永亮 / Yongliang Shen

浙江大学百人计划研究员

zju-real.github.io · github.com/ZJU-REAL