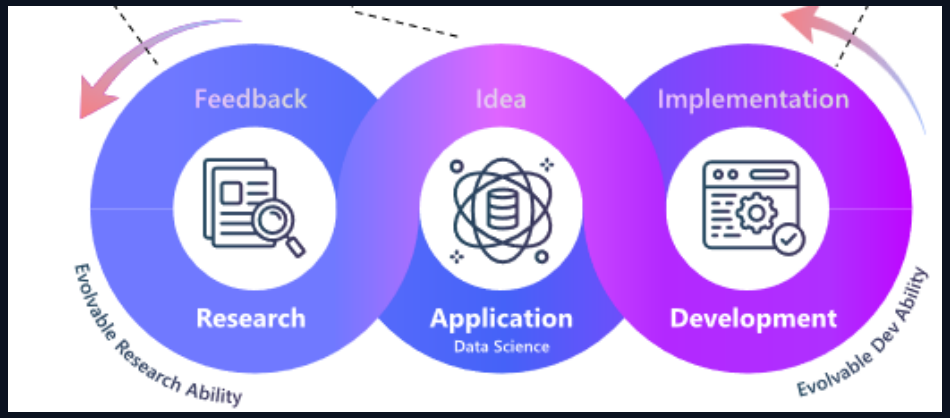
面向科学和工程场景的自进化Agent



浙江大学
方棡楷

NICE 交流 · 2026.10.24

个人介绍

浙江大学 ZHEJIANG UNIVERSITY



方棣楷 博士

百人计划研究员 | 博士生导师

单位 信息与电子工程学院

邮箱 fsk@zju.edu.cn

研究方向
- 机器学习
- 生成式物理仿真和信号处理
- LLM多智能体系统
- 大规模高维时空序列

中国科学技术大学 University of Science and Technology of China 1958

TEMPLE UNIVERSITY Of The Commonwealth System Of Higher Education

THE UNIVERSITY OF UTAH

Morgan Stanley

達摩院 ALIBABA DAMO ACADEMY

Microsoft Research 微软亚洲研究院

RD-Agent

Fork 1.8k Starred 13.8k

MarS

Fork 187 Starred 1.8k

Agent：智能的最小工作单元

模型、工具与指令共同定义一个可执行系统。

What Is Agent?

Agents are systems that **independently** accomplish tasks on your behalf.



Python

```python
weather_agent = Agent(
    name="Weather agent",
    instructions="You are a helpful agent who can talk to users about the weather.",
    tools=[get_weather],
)
```

概念扩张与系统边界

同一类系统常被称为 workflow、orchestrator、optimizer 或 autonomous agent。

What Is Agent in Real World..

- “Fancy words are invented everyday… and combined randomly”

Perception

Structured feedback (logs, metrics, curves)

Reasoning

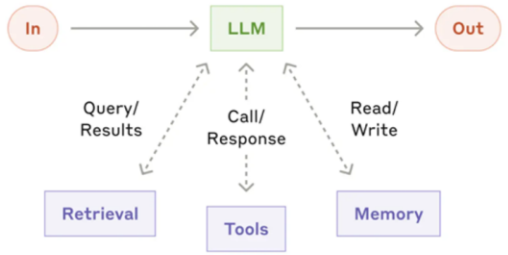
LLM-driven hypothesis formation

Action

Code execution, API calls, simulator runs

Memory

Accumulates outcomes · makes it evolving



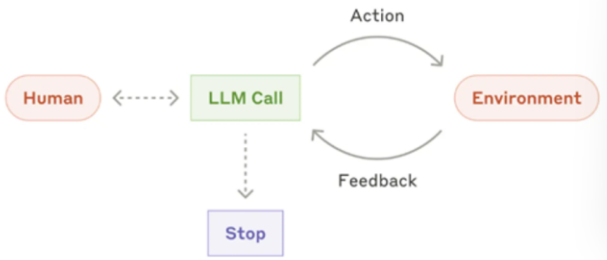
“Augmented LLM”



“evaluator-optimizer workflow”



“Orchestrator-workers”



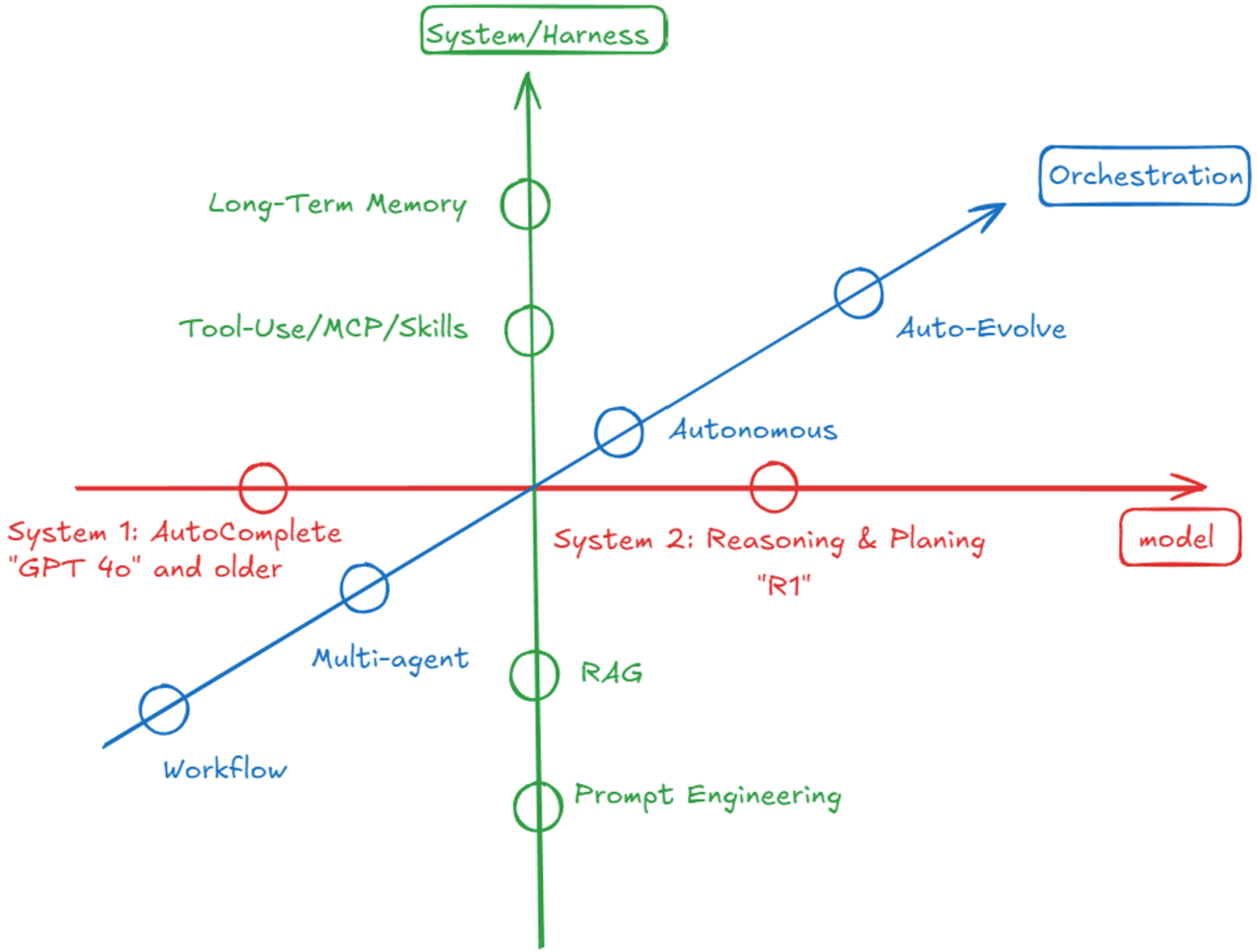
“Fully Autonomous”

Figs: “Building effective agents”, blog of anthropic

Agent 能力的组织维度

模型能力、工具使用、记忆、编排与系统边界共同决定系统有效性。

- “Trying to sort out such fancy words..”



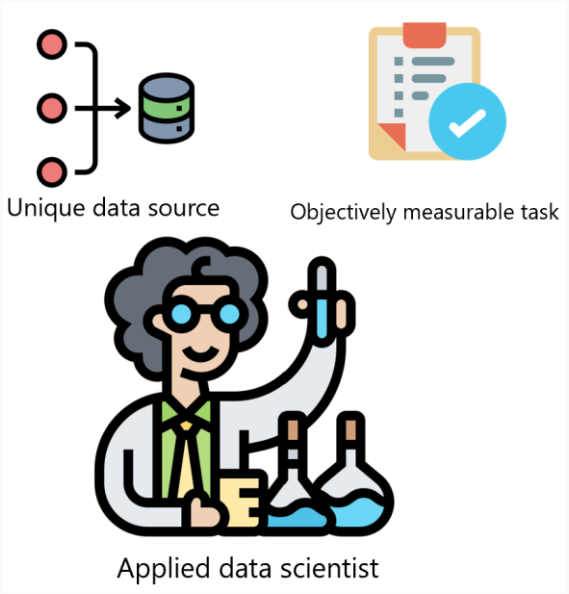
Fancy Enough!...?

for tasks like
AI4Coding/GUI/Web:

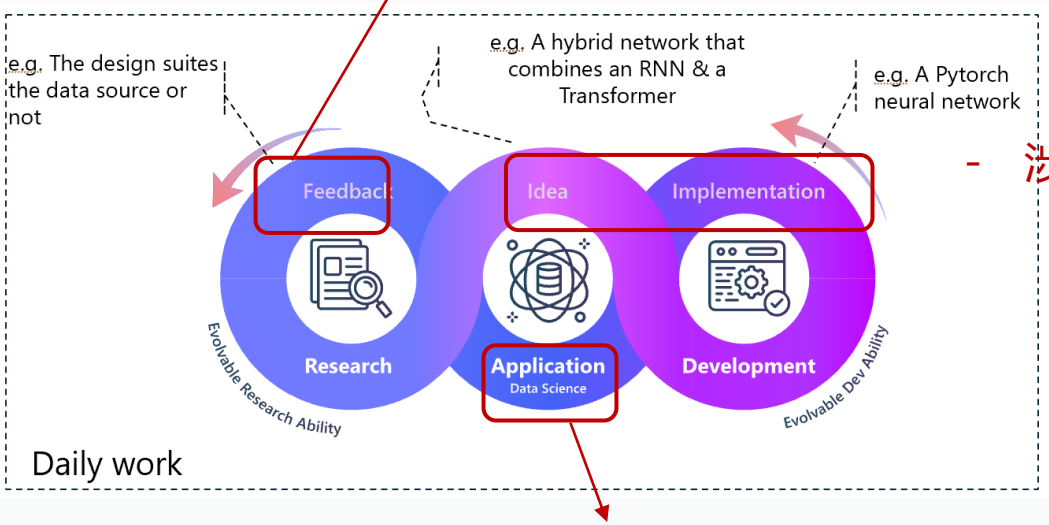
- Clear Feedback (Yes/No)
- Short-Term Evaluation

高价值工业场景任务的边界条件

面向可度量、数据驱动、长链路的工业与科研应用。



- 异质、异步、多模态的反馈 （不只是 Yes/No）



- 涉及的环节多：
  - Bad Idea?
  - Wrong feature?
  - 实现有问题?

长周期，高成本，黑箱

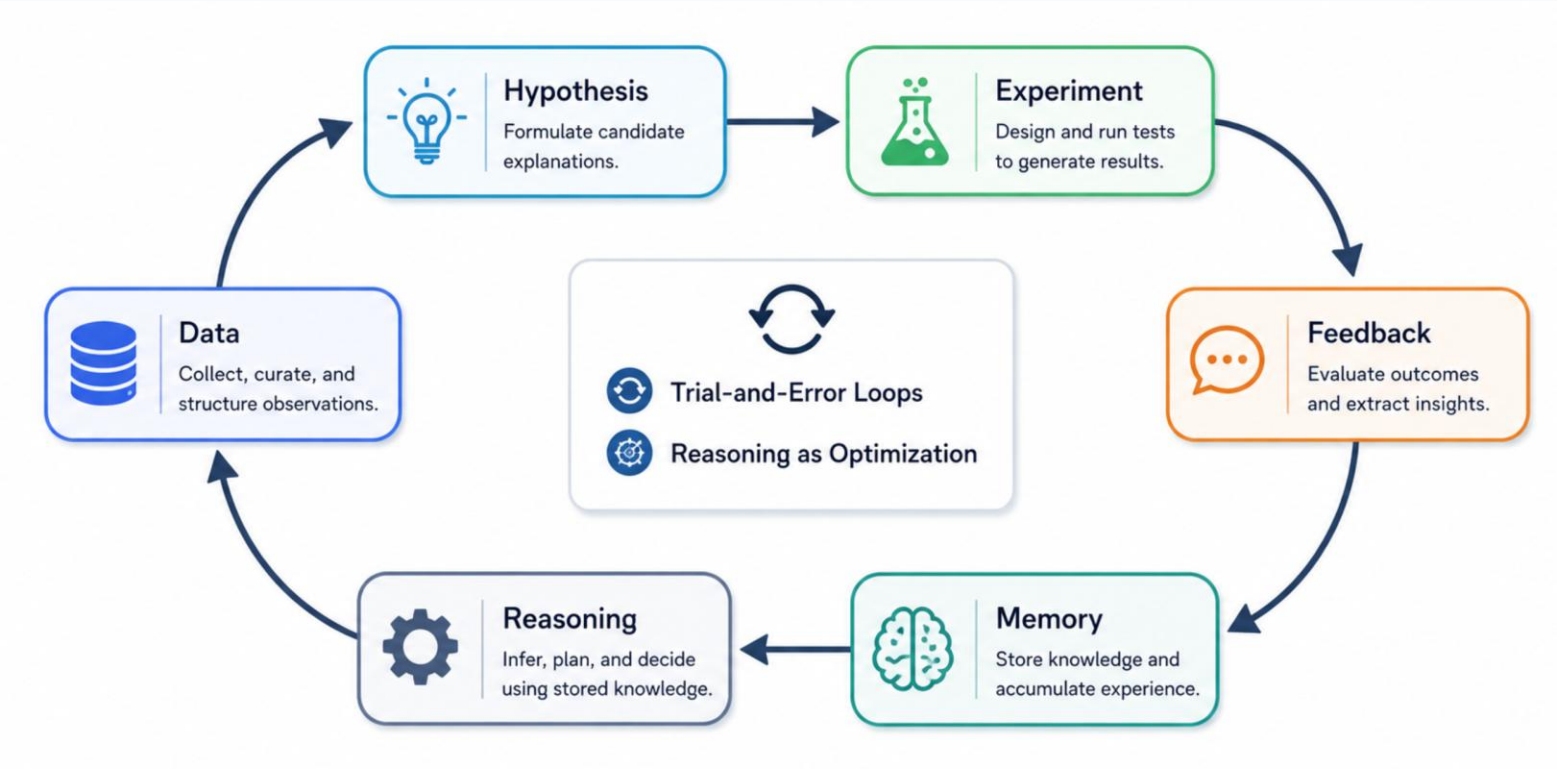
工业R&D与通用交互任务

二者共享 Agent 形式，但优化信号与工程约束不同。

维度	通用交互任务	工业R&D任务
反馈	明确、短期、用户可直接判断	多目标、长链路、包含日志/曲线/约束
评估	秒级到分钟级	分钟到小时，真实实验甚至更长
动作	网页、GUI、API等离散动作	代码、模型、因子、物理参数的连续空间
风险	重试成本低	错误归因困难，可能产生虚假进步
目标	完成任务	提升可复用的研究能力与研发吞吐

自进化R&D闭环-核心组件和设计原则

反馈被结构化，经验被保留，下一轮行动由诊断推理驱动。



形式化视角

- 感知：结构化反馈
- 推理：诊断与假设
- 行动：实验或实现
- 记忆：跨轮次经验
- 验证：约束与可信评估

第一部分

工业场景数据驱动R&D范式

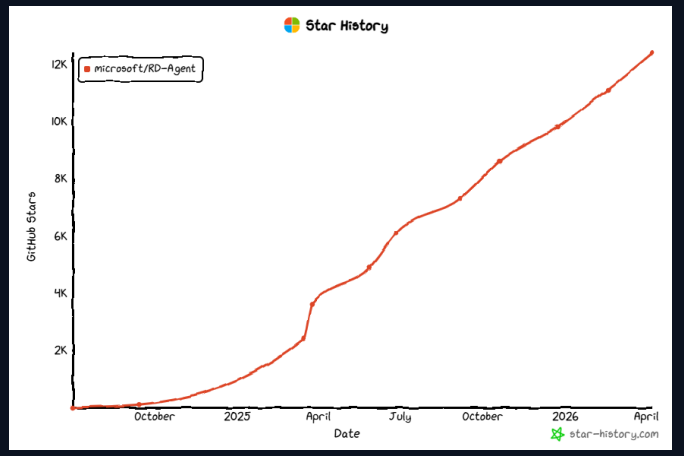
R&D-Agent：研究与开发解耦的自动化框架 (tech report arxiv. 2505.14738v2, 2603.01692v2)

Reasoning as Gradient：推理作为优化信号 (ACL 2026， arXiv:2603.01692)

核心问题：如何让反馈驱动可验证的研究迭代

RD-Agent

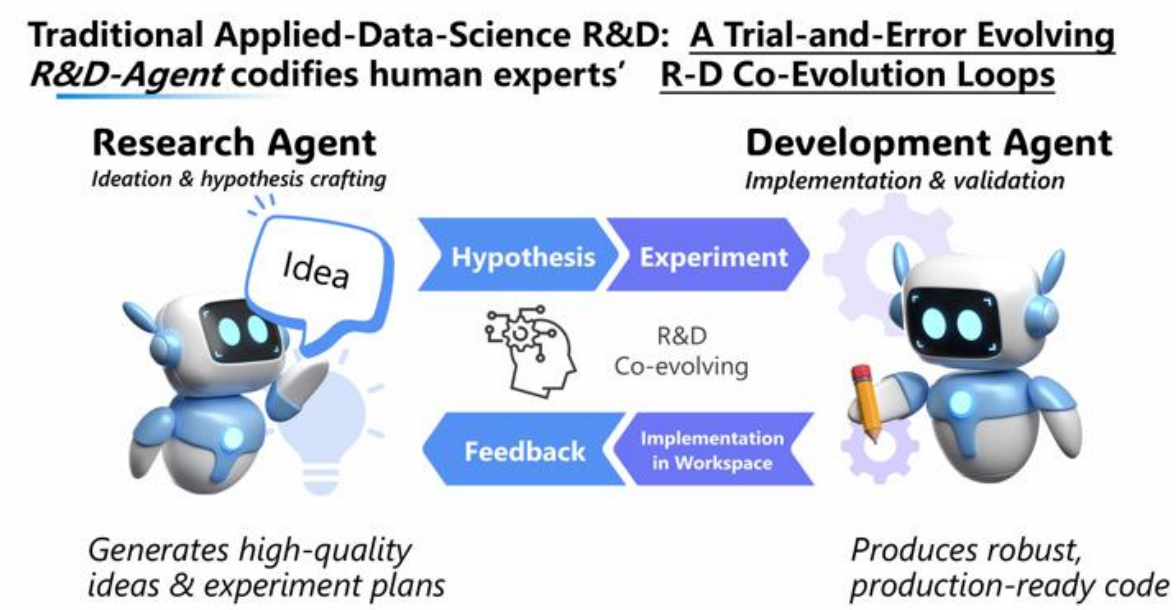
Fork 1.8k Starred 13.8k



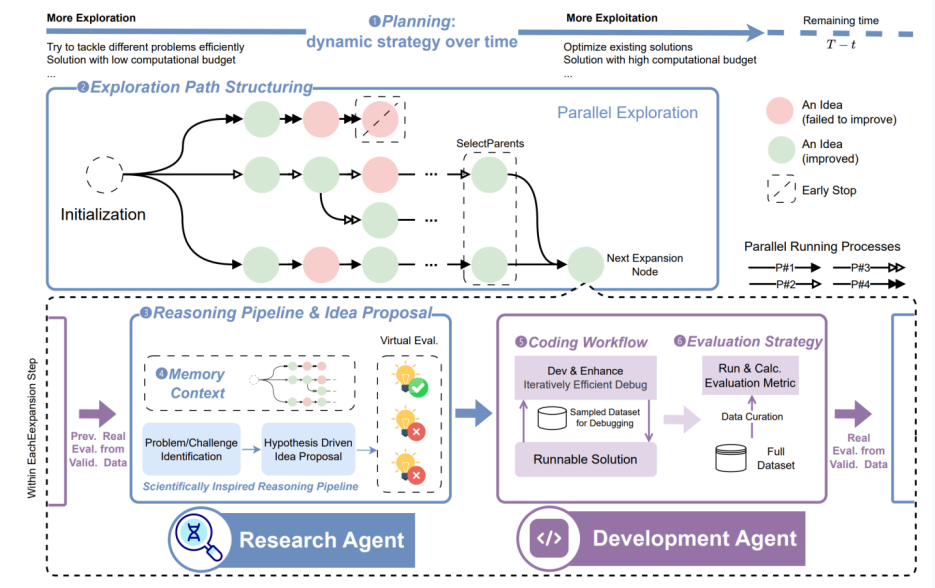
- https://github.com/microsoft/RD-Agent

R&D-Agent框架：化繁为简，双Agent框架

将机器学习工程拆解为研究-落地的两阶段流程。

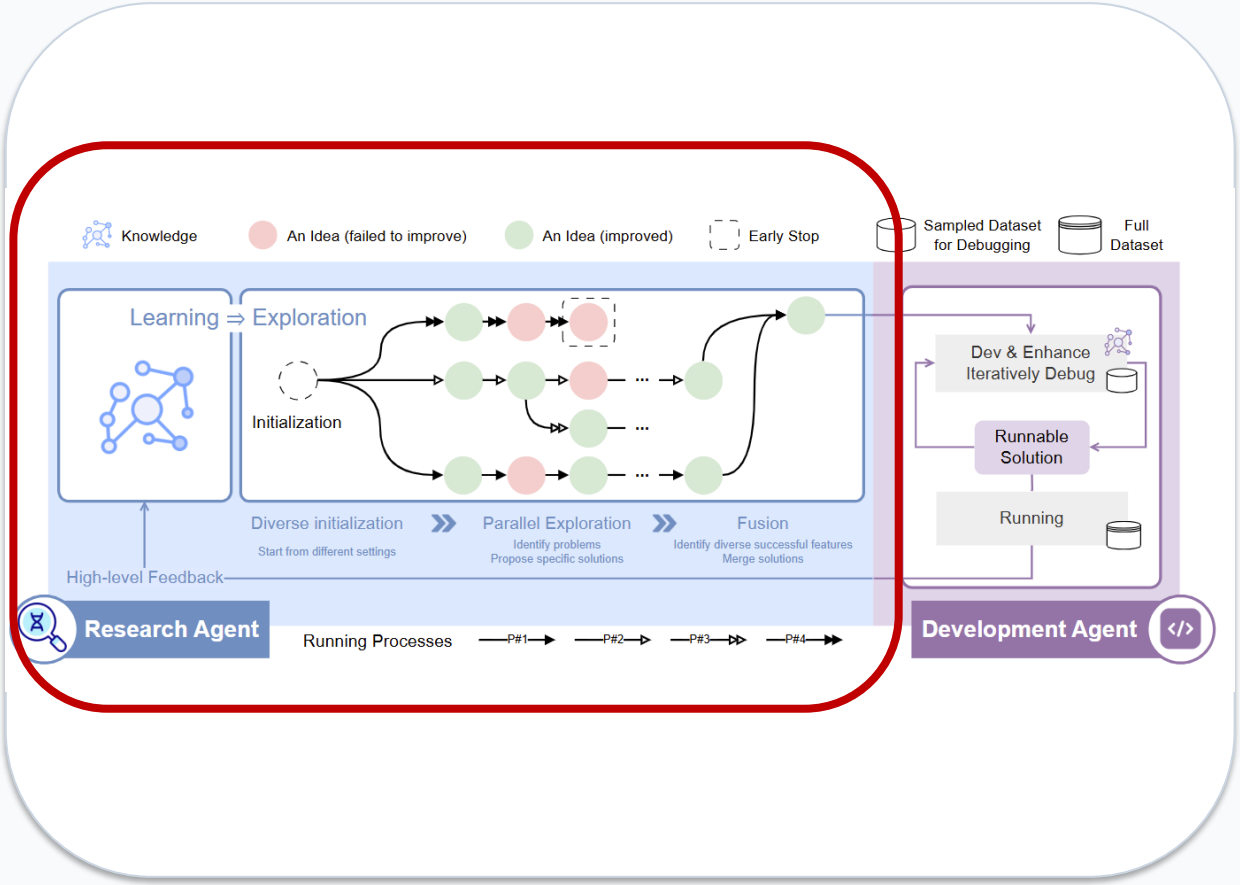


- Research Agent 负责规划、探索、记忆与假设形成。
- Development Agent 负责实现、调试与稳定评估。
- 共享记忆使多分支探索具有协同效应 + 协同进化。



研究阶段/Research agent： 更高效的探索

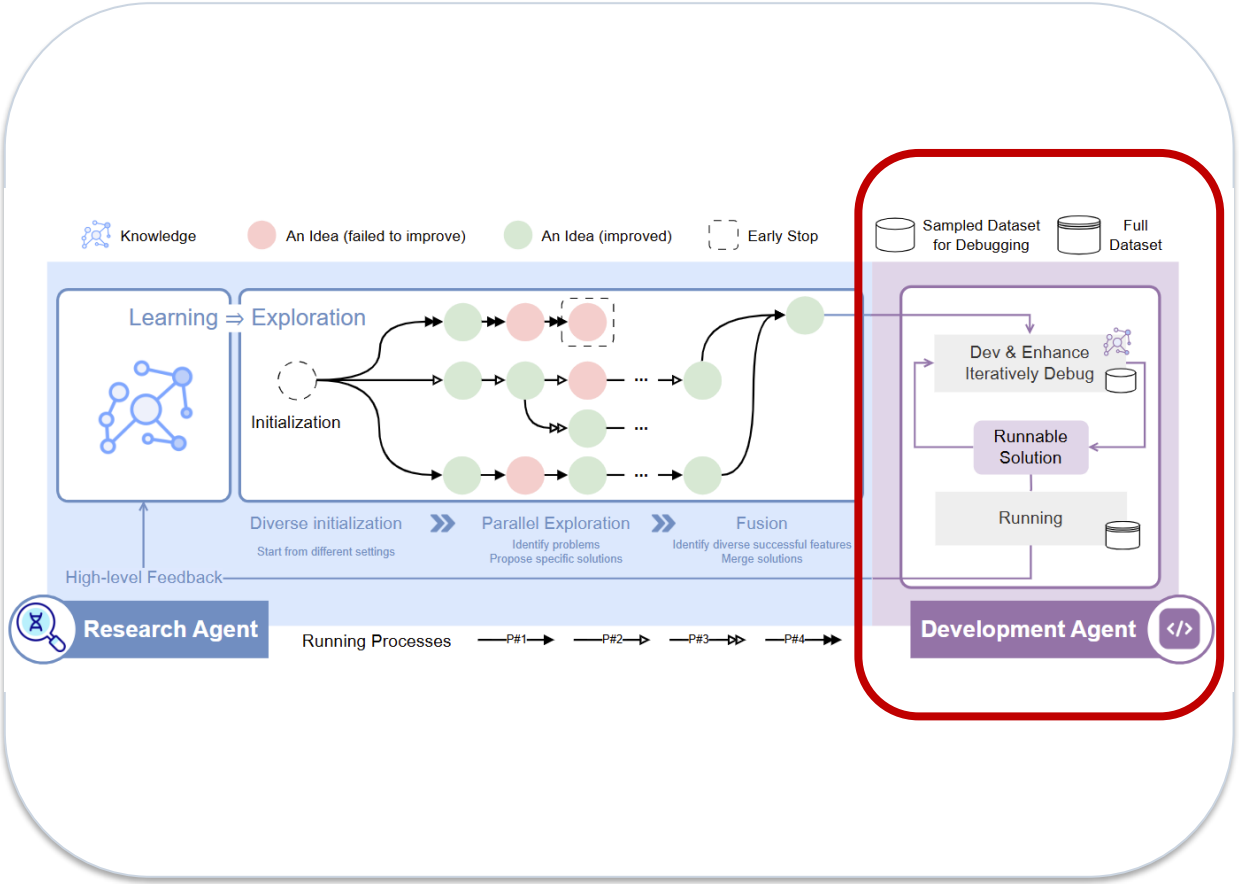
多路探索与结构化推理共同决定 idea 的质量。

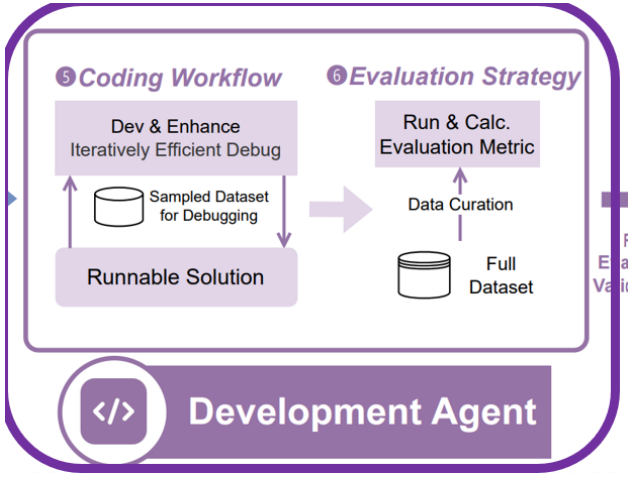


- 强制多样性避免所有分支过早收敛。
- 三步推理：问题识别、机制假设、具体改动。
- 高分经验通过记忆跨分支传播。

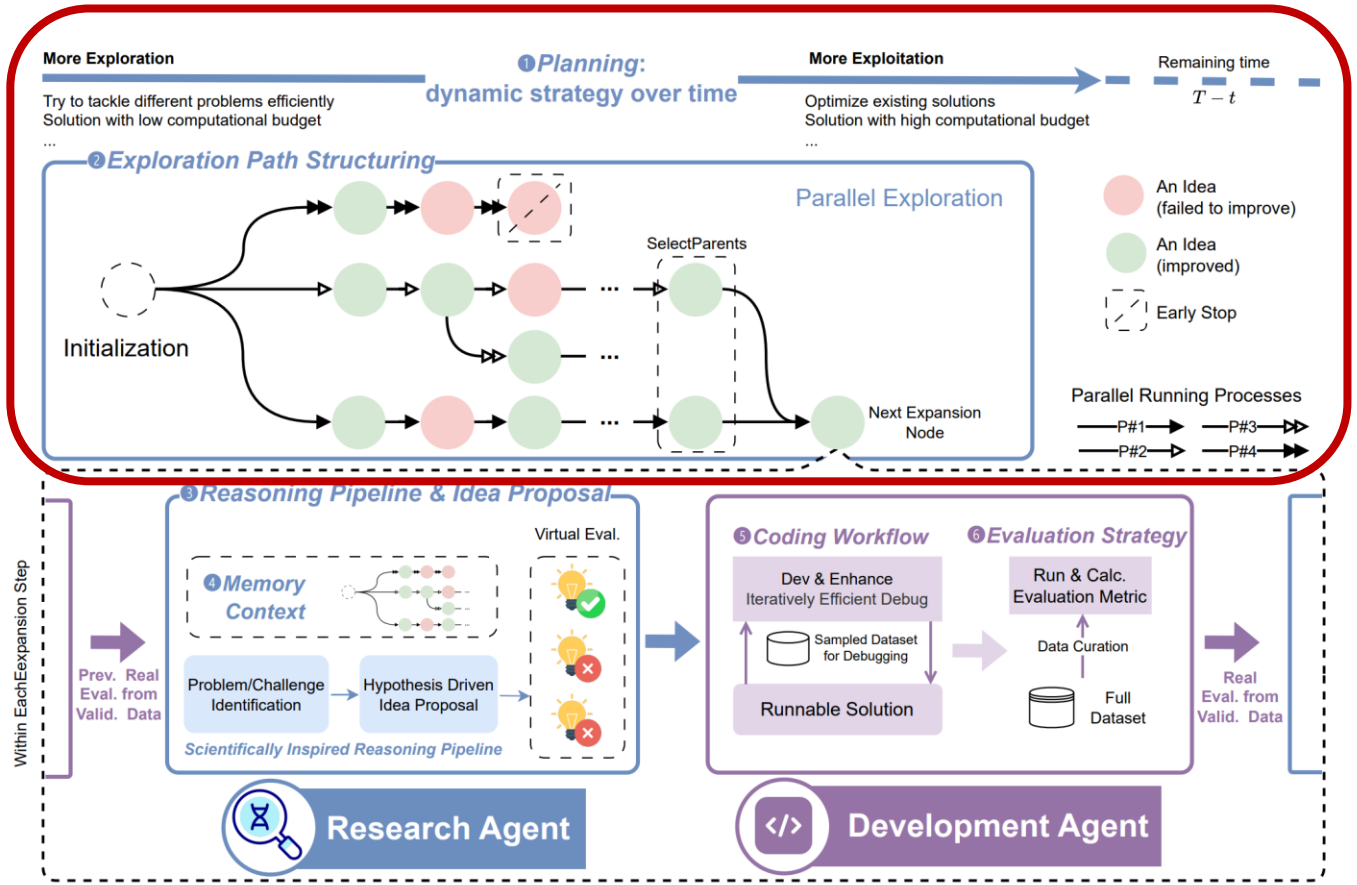
开发阶段/ Development agent： 沙盒与自检

在codin正式进入完整训练之前， 进行充分的可行性与压力测试。





成本感知的探索路径规划



- 在整个探索流程的不同阶段，尝试不同风格的探索策略
- 及时早停
- 可作为更加复杂探索策略的接口

Prompt示例： 鼓励结构化， 但不把结构化写死； 鼓励使用工具 inner-loop 来debug

Scientific Hypothesis Generation —— Reasoning Pipeline Component

You are a research scientist formulating testable hypotheses. For each hypothesis, perform two main tasks: hypothesis proposal and rigorous five-dimensional evaluation.

Hypothesis Development Guidelines:

1. **Specificity & Decisiveness:** State exact, unambiguous changes. Avoid vague goals or alternatives.
2. **Testability & Actionability:** Describe implementable and measurable changes. Focus on single, unified conceptual improvements.
3. **Evidence-Based Reasoning:** Ground hypotheses in experimental history or domain knowledge.
4. **Implementation Feasibility:** Consider resource constraints and technical complexity.

Five-Dimensional Evaluation Protocol: Score each hypothesis (1-10) across:

- **Problem-Hypothesis Alignment:** How well the hypothesis addresses the identified problem
- **Expected Impact:** The estimated improvement after applying the hypothesis
- **Novelty:** Degree of innovation compared to previous attempts
- **Feasibility:** The ease of implementing the proposed hypothesis
- **Risk-Reward Balance:** The exploration-exploitation balance of the proposed hypothesis

Component Classification: Assign each hypothesis to: `DataLoadSpec`, `FeatureEng`, `Model`, `Ensemble`, or `Workflow`.

Memory-Enhanced Code Generation —— Memory Context Component

You are a grandmaster-level data scientist generating robust, debuggable code following systematic development process.

Important Context: You are working on sample datasets and your code will go through automated iterations. Design your code to be iteration-friendly with comprehensive print statements and clear debugging information to facilitate the automatic improvement process.

Memory-Enhanced Guidelines:

1. **Historical Learning:** Reference previous failed attempts and their feedback
2. **Pattern Reuse:** Apply successful patterns from similar tasks
3. **Error Prevention:** Avoid mistakes that occurred in previous experiments
4. **Performance Optimization:** Implement improvements suggested in historical feedback

Quality Assurance Requirements:

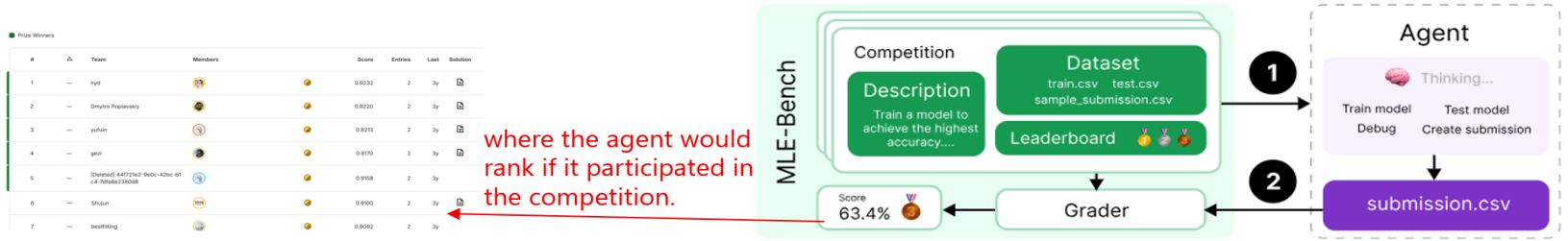
- **Debug Mode Integration:** Support `-debug` flag with data sampling and timing estimation
- **Structured Output:** Use print statements for progress tracking, avoid external logging dependencies
- **Reproducibility:** Implement proper random seed management and deterministic behavior
- **Resource Management:** Dynamic resource allocation and proportional data splitting

Open-AI MLE-Bench:

75个真实Kaggle竞赛上的端到端评估。

➥ MLE-Bench: the largest public ML-engineering benchmark

- 75 kaggle competitions crossing several domains published and open-sourced by OpenAI



- Industry span



- 覆盖CV、NLP、表格、时序与多模态任务。
- 任务要求特征工程、模型选择、调参、集成与调试。

- Task span
  - classification, regression, object detection, semantic segmentation, sequence prediction, QA...

MLE-Bench 评估

75个真实Kaggle竞赛上的端到端评估。

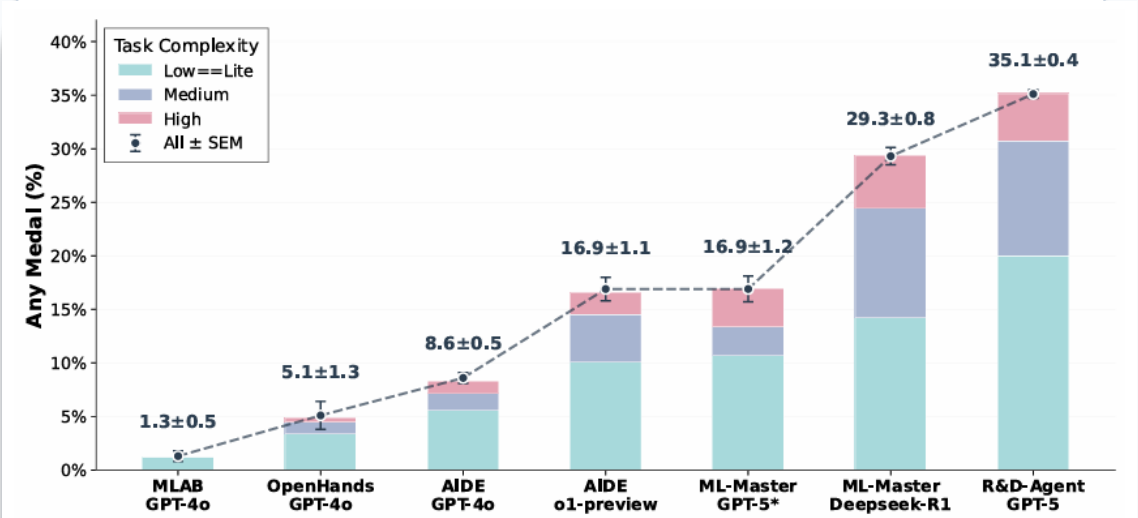
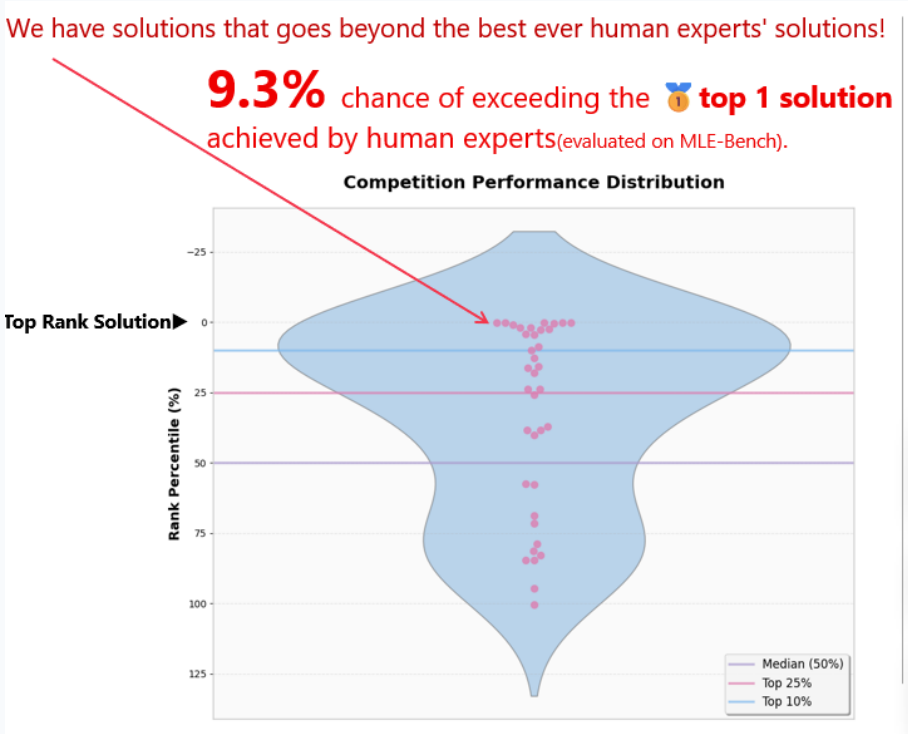


Figure 1: Agent performance on MLE-Bench. Stacked bars show any medal rates for Low==Lite (22 tasks), Medium (38 tasks), and High (15 tasks) complexity levels. The dashed line indicates overall performance (mean ± SEM). R&D-Agent achieves SOTA performance at 35.1 ± 0.4%. * indicates our re-evaluation of ML-Master within our environment.

35.1%
奖牌率

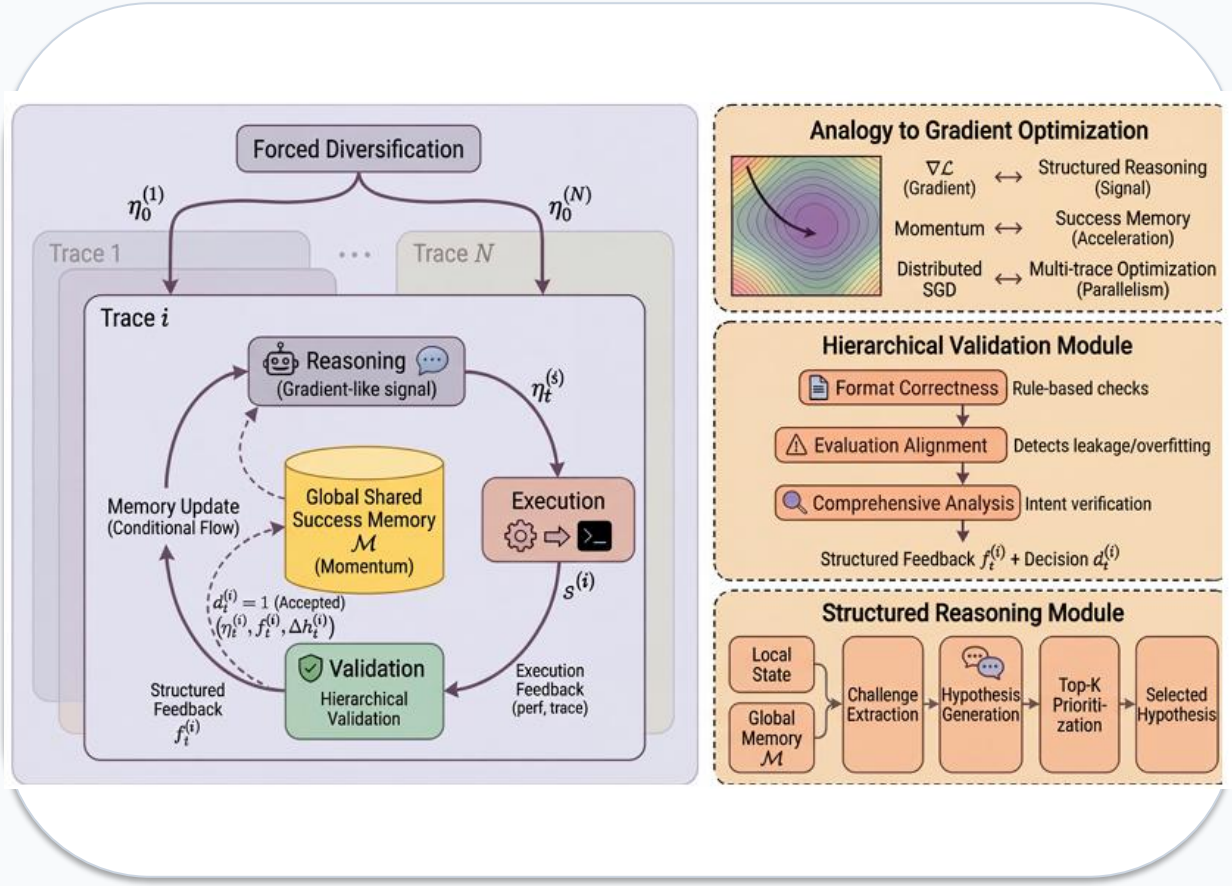
96%
有效提交率

13.8k
GitHub星标



更进一步: 如何用好大模型的推理能力——推理作为优化信号

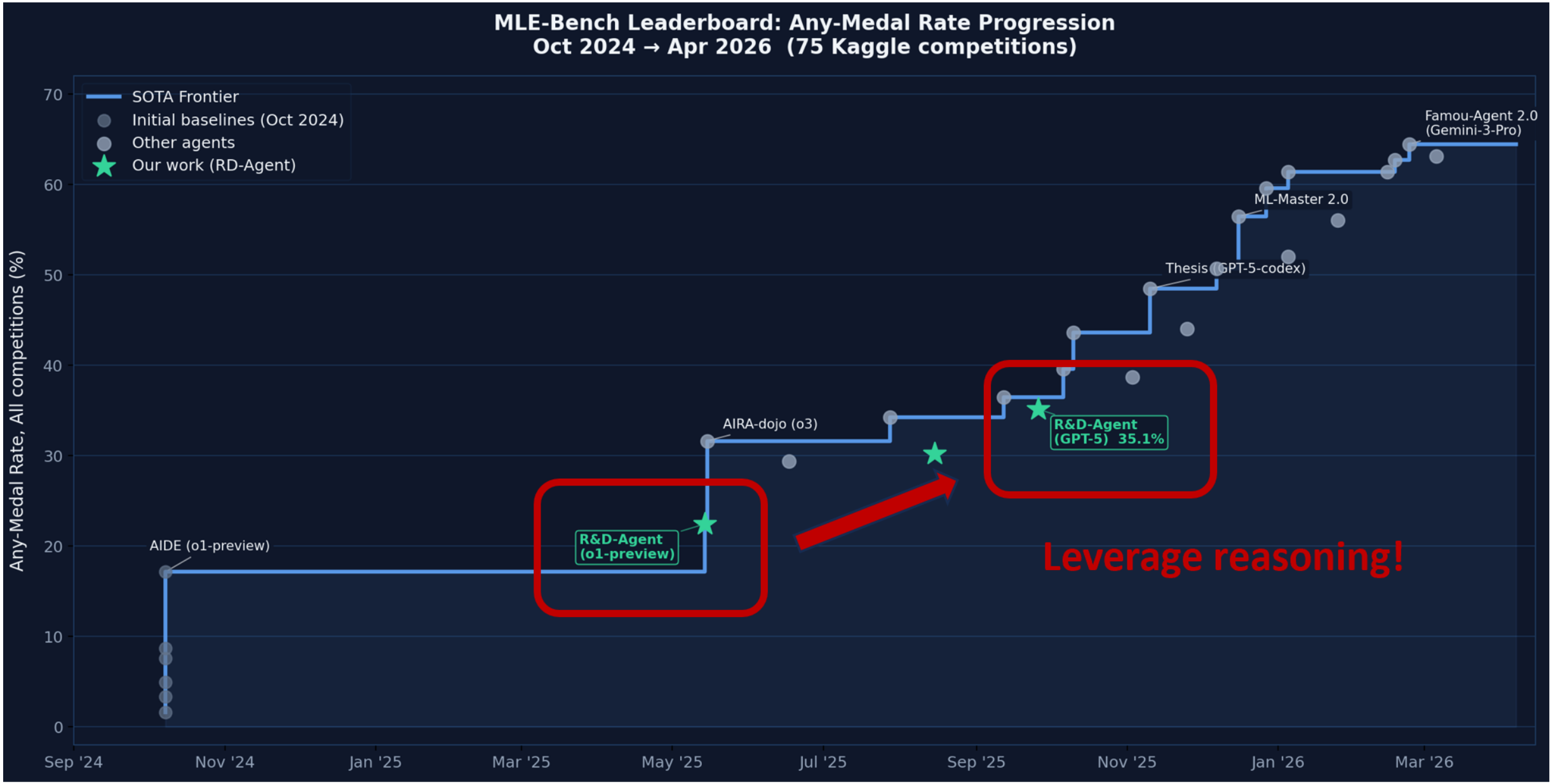
R&D Agent -> GOME 将诊断推理、成功记忆与多轨迹执行映射到优化器组件。



- 结构化诊断推理对应更新方向。
- 成功记忆对应动量。
- 多轨迹执行对应分布式优化。

Concept	GOME Component	Functional Role
Gradient ∇L	Structured Reasoning	Signal: decides how to update
Momentum	Success Memory	Acceleration via proven patterns
Distributed SGD	Multi-trace Optimization	Parallelism with knowledge sharing

Table 2: Analogy between GOME and gradient-based optimization.


MLE-Bench Leaderboard: Any-Medal Rate Progression
Oct 2024 → Apr 2026 (75 Kaggle competitions)
SOTA Frontier
Initial baselines (Oct 2024)
Other agents
Our work (RD-Agent)
Any-Medal Rate, All competitions (%)
0
10
20
30
40
50
60
70
Sep '24
Nov '24
Jan '25
Mar '25
May '25
Jul '25
Sep '25
Nov '25
Jan '26
Mar '26
AIDE (o1-preview)
R&D-Agent (o1-preview)
AIRA-dojo (o3)
Thesis (GPT-5-codex)
ML-Master 2.0
R&D-Agent (GPT-5) 35.1%
Famou-Agent 2.0 (Gemini-3-Pro)
Leverage reasoning!

第二部分

垂直工程和科学场景中的智能体设计

R&D-Agent-Quant：因子与模型的联合优化 (NeurIPS 2025，arxiv 2505.15155)

Battery-Sim-Agent：电池科学数字孪生中的参数反演 ((KDD 2026, arxiv 2605.29560)

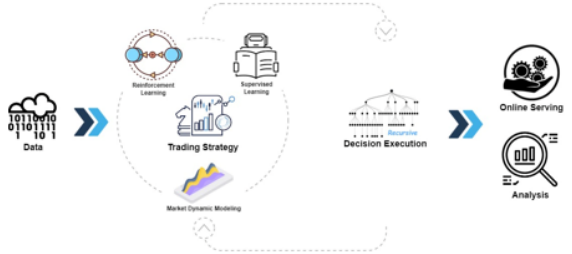
重点：将通用闭环接入领域反馈、约束与机理语言

量化研发场景

金融市场提供可回测反馈，但噪声、漂移与约束显著。

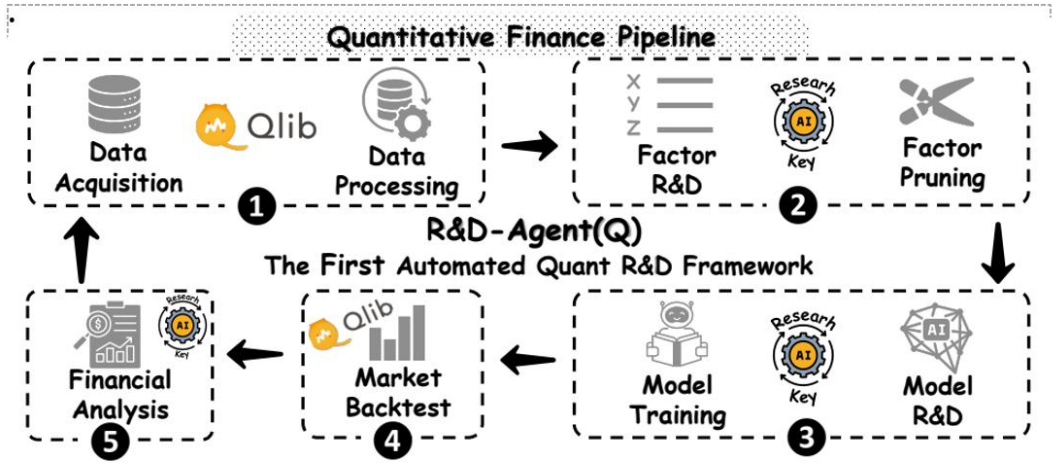
Qlib https://github.com/microsoft/qlib

Fork 5.3k Starred 34.2k



Qlib is an AI-oriented quantitative investment platform that aims to realize the potential, empower research, and create value using AI technologies in quantitative investment, from exploring ideas to implementing productions. Qlib supports diverse machine learning modeling paradigms. including supervised learning, market dynamics modeling, and RL.

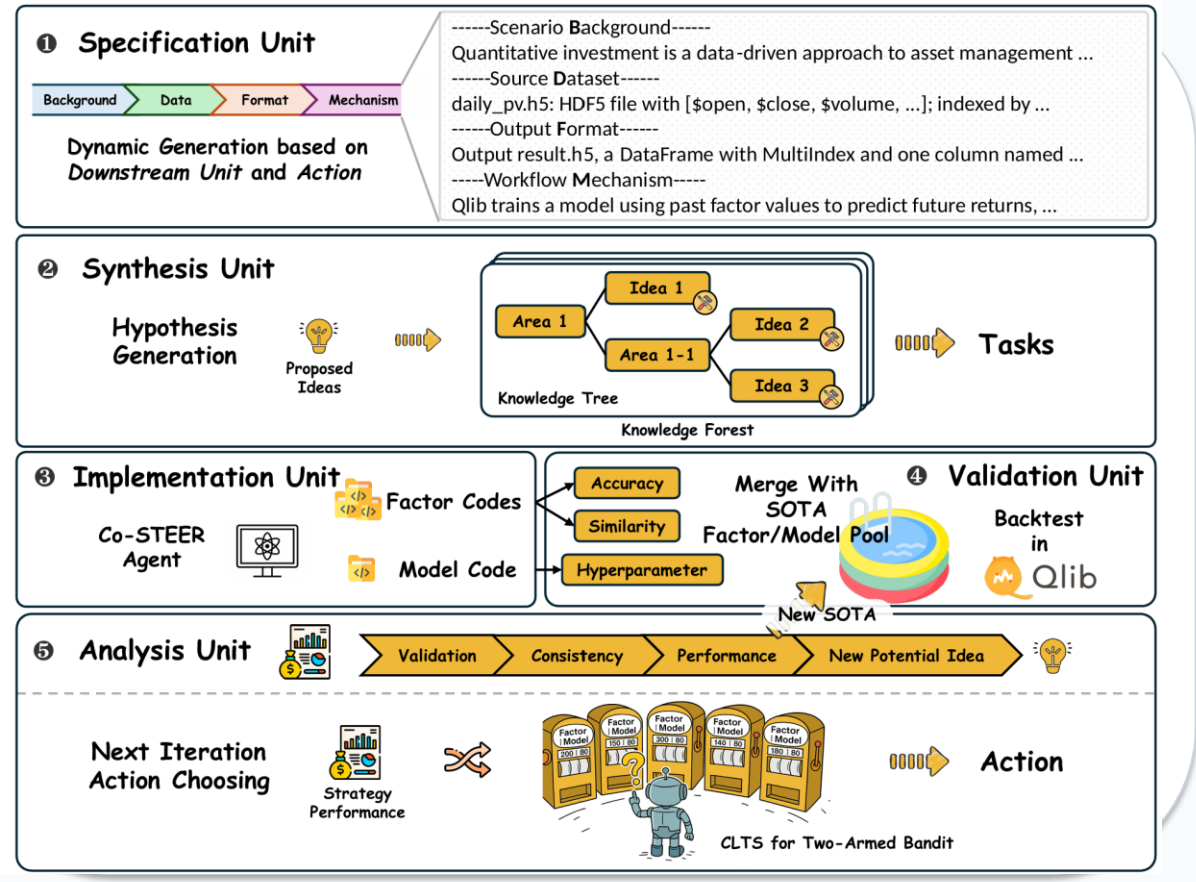
Industry Engagement: R&D-Agent-Quant
➥Automate Quant Research & Development



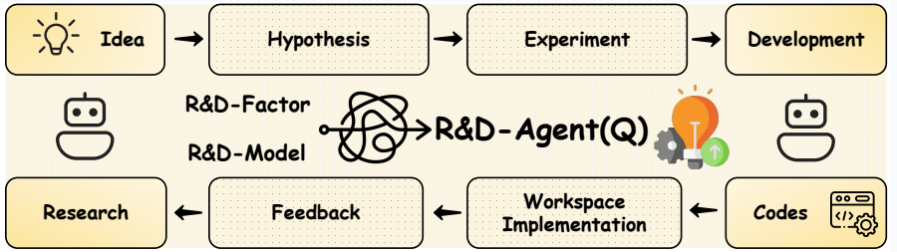
- 核心对象不只是模型，还包括数据、因子、交易约束与风险指标。
- 单独优化因子或模型容易造成局部最优。
- 研究方向需要根据回测反馈自适应调度。

R&D-Agent(Q)

面向因子挖掘与模型创新的多智能体协同框架。



- Specification Unit 将研究意图转成任务。
- Research Unit 形成因子与模型假设。
- Implementation Unit 生成代码并接入Qlib回测。
- Analysis Unit 将收益、稳定性与风险反馈给下一轮。



量化实验结果

联合优化在收益、因子数量与稳定性之间取得更好平衡。

Table 1: Experimental results of all models on the CSI 300 constituent stock dataset, including factor predictive metrics and strategy performance metrics. Visual cues indicate ranking groups: **Best**, Second Best, Good (3–8), Average (9–14), Poor (15–20), and Worse (21–26).

Models		CSI300							
		Factor Predictive Power Metrics				Performance Metrics			
		IC	ICIR	Rank IC	Rank ICIR	ARR	IR	MDD	CR
Machine-Learning Models	Linear	0.0134	0.0992	0.0273	0.1962	-0.0302	-0.3710	-0.1987	-0.1520
	MLP	0.0291	0.2096	0.0412	0.3238	0.0003	0.0037	-0.1390	0.0022
	LightGBM	0.0277	0.2211	0.0386	0.3120	0.0397	0.5664	-0.0855	0.4643
	XGBoost	0.0291	0.2410	0.0384	0.3257	0.0316	0.4620	-0.1139	0.2774
	CatBoost	0.0279	0.2181	0.0393	0.3110	0.0513	0.7008	-0.0924	0.5552
	DoubleEnsemble	0.0294	0.2246	0.0417	0.3211	0.0551	0.7968	-0.0971	0.5675
Deep-Learning Models	Transformer	0.0317	0.2538	0.0434	0.3624	0.0293	0.4267	-0.0987	0.2969
	GRU	0.0315	0.2450	0.0428	0.3440	0.0344	0.5160	-0.1017	0.3382
	LSTM	0.0318	0.2367	0.0435	0.3389	0.0381	0.5561	-0.1207	0.3157
	ALSTM	0.0362	0.2789	0.0463	0.3661	0.0470	0.6992	-0.1072	0.4384
	GATs	0.0349	0.2511	0.0462	0.3564	0.0497	0.7338	-0.0777	0.6396
	PatchTST	0.0247	0.1945	0.0315	0.2463	0.0571	0.7191	-0.1327	0.4303
	iTransformer	0.0270	0.1946	0.0340	0.2365	0.0979	1.2337	-0.1151	0.8506
	Mamba	0.0281	0.2244	0.0374	0.2952	0.0229	0.3163	-0.1154	0.1984
	TRA	0.0404	0.3197	0.0490	0.4047	0.0649	1.0091	-0.0860	0.7547
	MASTER	0.0215	0.1925	0.0296	0.2486	0.0896	1.3406	-0.0851	1.0528
Factor Libraries	Alpha 101	0.0308	0.2588	0.0331	0.2749	0.0512	0.5783	-0.1253	0.4085
	Alpha 158	0.0341	0.2952	0.0450	0.3987	0.0570	0.8459	-0.0771	0.7393
	Alpha 360	0.0420	0.3290	0.0514	0.4225	0.0438	0.6731	-0.0721	0.6074
	AutoAlpha	0.0334	0.2656	0.0361	0.2967	0.0400	0.4288	-0.1225	0.3266
RD-Agent Series Framework*	RD-Factor$_{GPT\text{-}4o}$	0.0489	0.4050	0.0521	**0.4425**	**0.1461**	1.6835	-0.0750	**1.9468**
	RD-Factor$_{o3\text{-}mini}$	0.0497	0.3931	0.0500	0.4246	0.1184	1.3566	-0.0910	1.3016
	RD-Model$_{GPT\text{-}4o}$	0.0326	0.2305	0.0401	0.2767	0.1229	1.6676	-0.0876	1.4029
	RD-Model$_{o3\text{-}mini}$	0.0469	0.3688	**0.0546**	0.4385	0.1009	1.7009	**-0.0694**	1.4538
	RD-Agent(Q)$_{GPT\text{-}4o}$	0.0497	0.4069	0.0499	0.4122	0.1144	1.3167	-0.0811	1.4108
	RD-Agent(Q)$_{o3\text{-}mini}$	**0.0532**	**0.4278**	0.0495	0.4091	0.1421	**1.7382**	-0.0742	1.9150

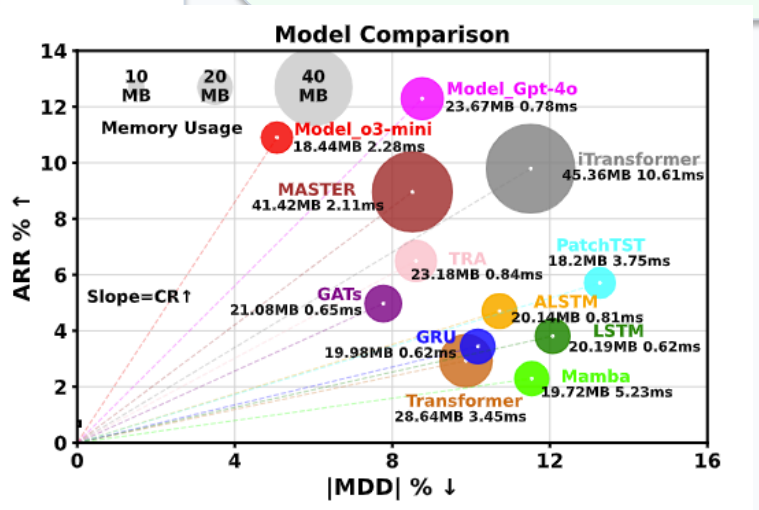
2x

年化收益

70%

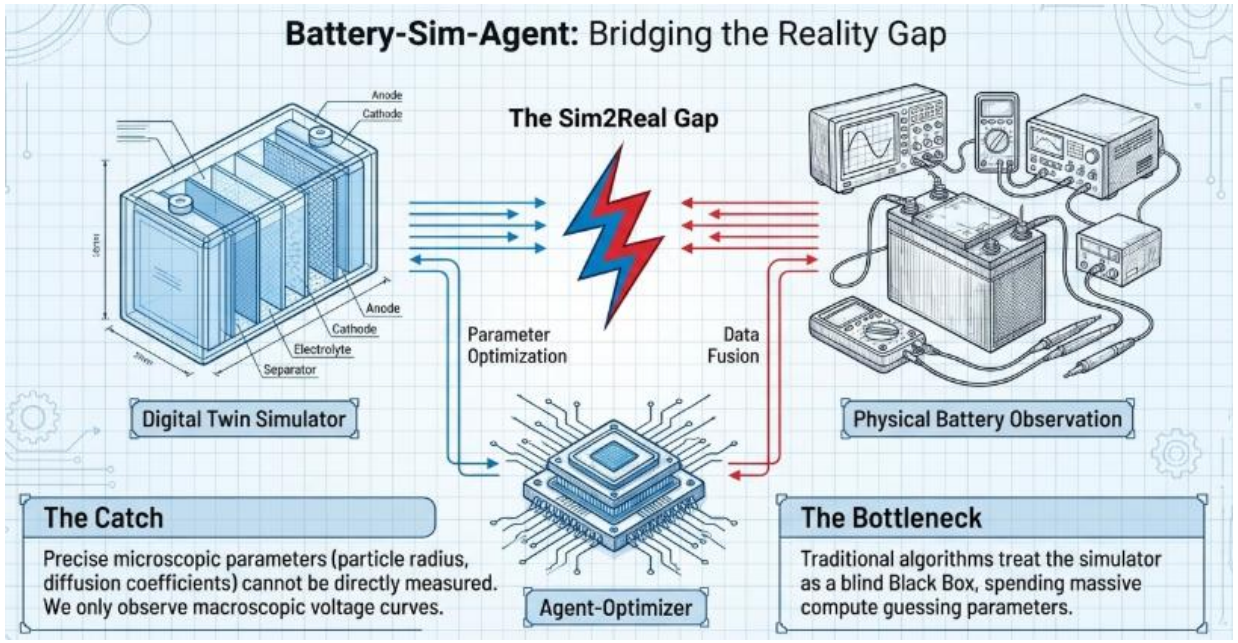
更少因子

- 最高达到经典因子库约2倍年化收益。
- 以更少因子维持策略表现，降低冗余。
- 多臂老虎机调度用于研究方向选择。



Battery-Sim-Agent: 电池数字孪生搭建 & 参数反演

高保真模拟器存在，但微观参数无法直接测量。



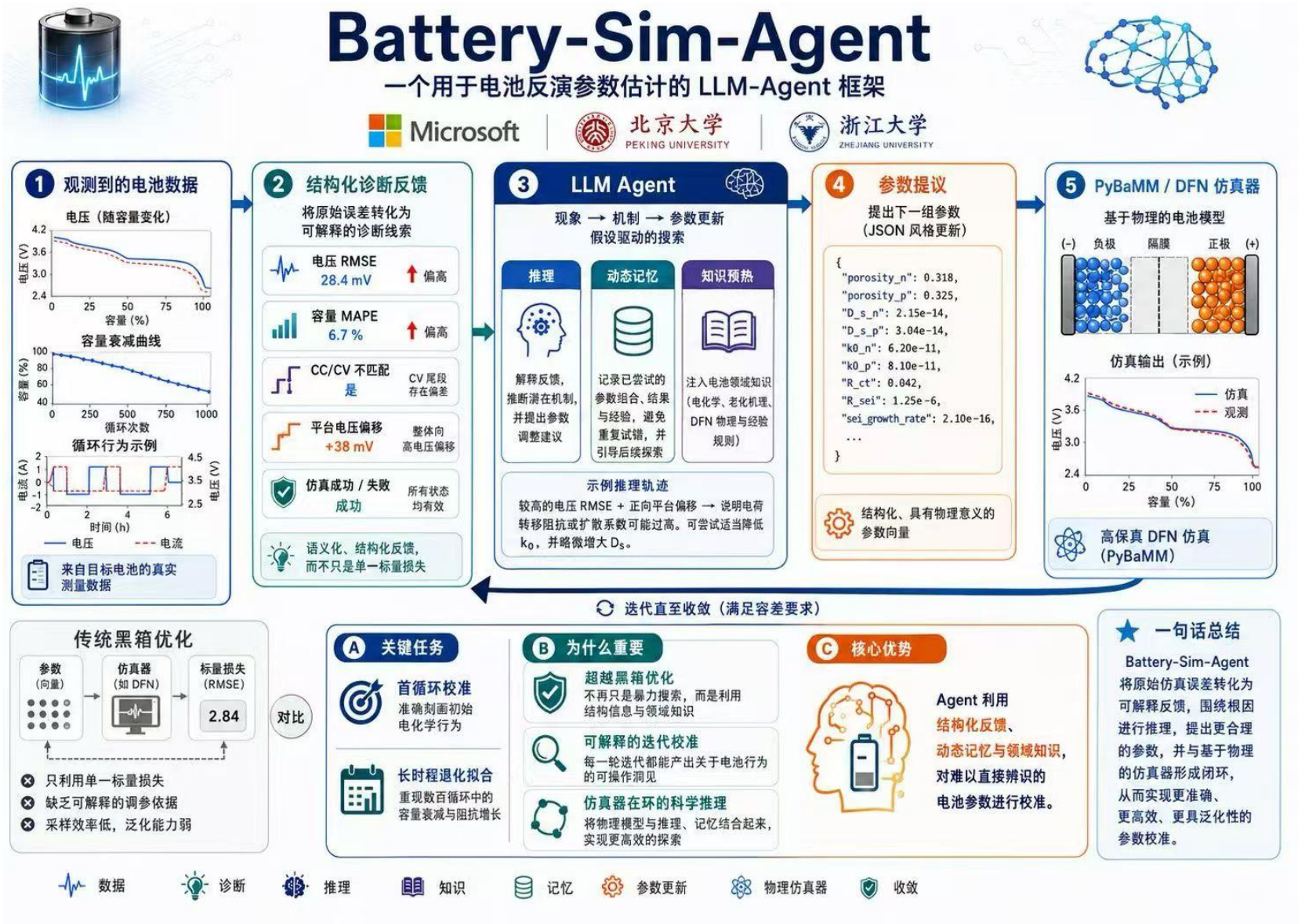
数字孪生搭建问题 <->高维黑盒优化问题

把Agent当一个有领域知识的万能优化器！

- 观测量是宏观电压、容量、温度等曲线。
- 参数空间高维、耦合、不可微且病态。
- 真实老化实验周期长，样本效率至关重要。

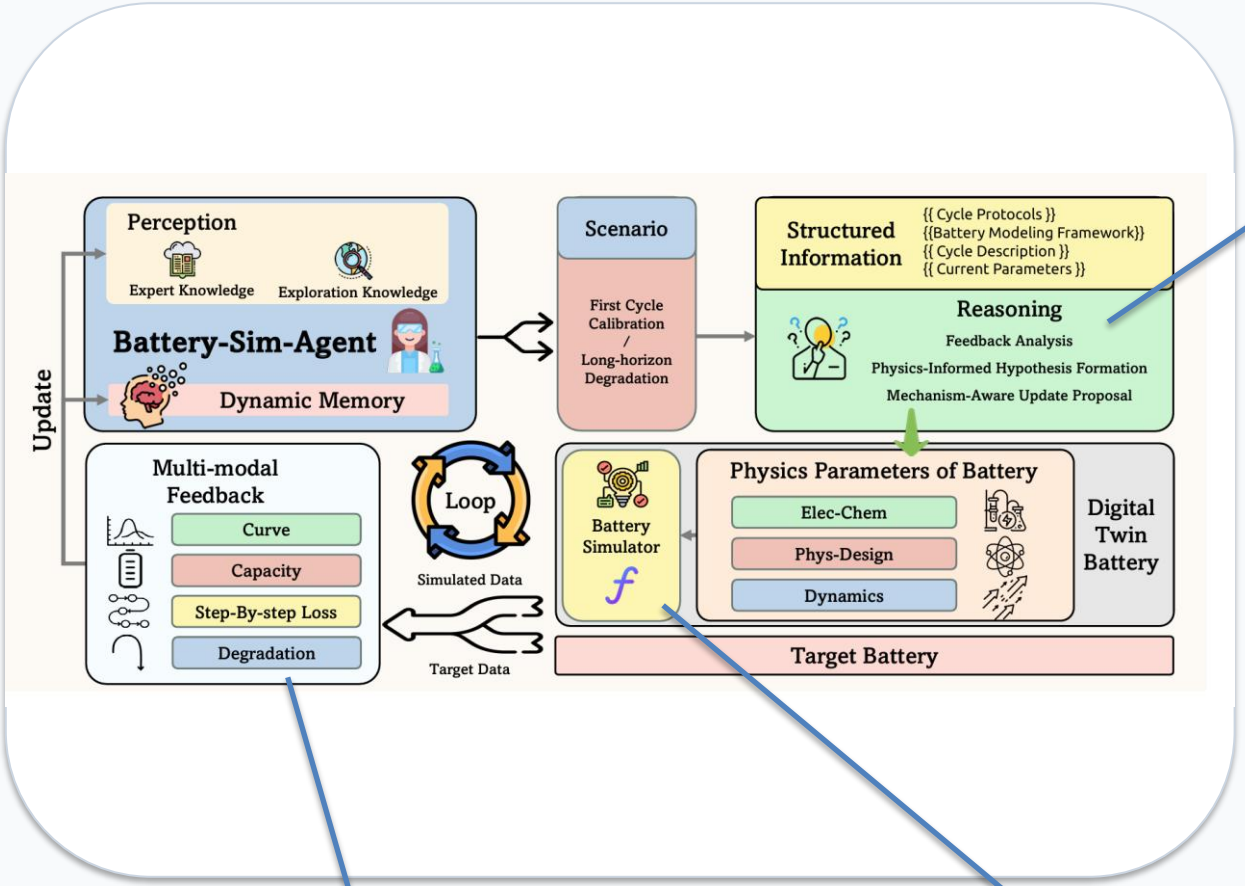
Battery-Sim-Agent: 电池数字孪生搭建 & 参数反演

高保真模拟器存在，但微观参数无法直接测量。



Battery-Sim-Agent: 电池数字孪生搭建 & 参数反演

将曲线残差转化为物理假设，再提出受约束参数更新。



Agent输出科学假设机制

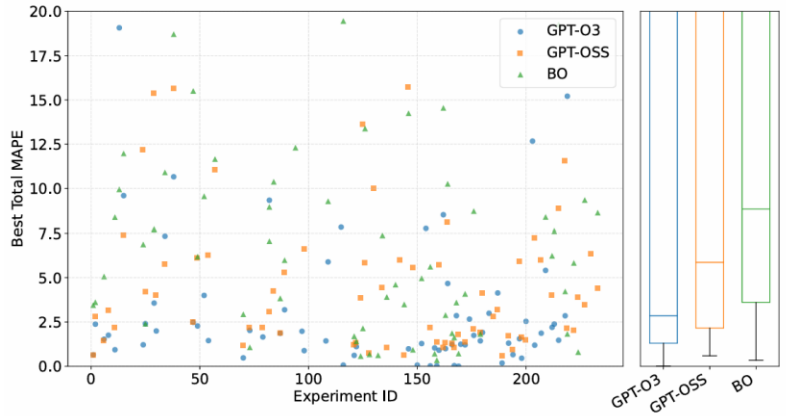
- 反馈翻译器把原始曲线转成命名诊断字段。
- Agent 必须先给出物理机制，再给出参数更新。
- 记忆记录每次假设的证实程度与副作用。

多模态异构异步反馈信号

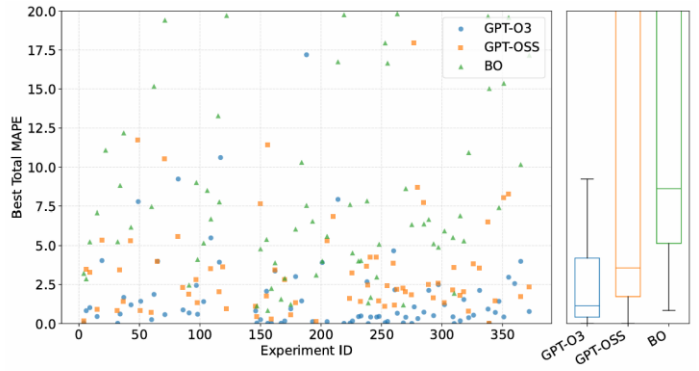
从”**仿真器as验证器**” 到 **“仿真器 + Agent 校准”**

电池参数估计结果

推理型优化器在样本效率与可解释性上优于黑箱搜索。



(a) Regular Mode



(b) Extreme Mode

67-95%

误差降低

5-10x

样本效率

- 相对贝叶斯优化显著降低曲线匹配误差。
- 参数更新具有机理解释，便于专家审查。
- **框架可扩展到其他高保真仿真逆问题。**

一切涉及到工艺优化、参数校准的黑盒优化问题，
理论上都可以用Agent重做一遍！

第三部分

教训与暴论：不要迷信“银弹”

教训1：迷信智能体数量的“银弹”

在大多数科学与工程场景下，并不需要多智能体

Google Research Google DeepMind

Towards a Science of Scaling Agent Systems

Yubin Kim[1,3,†], Ken Gu[1], Chanwoo Park[3], Chunjong Park[2], Samuel Schmidgall[2], A. Ali Heydari[1], Yao Yan[1], Zhihan Zhang[1], Yuchen Zhuang[2], Yun Liu[1], Mark Malhotra[1], Paul Pu Liang[3], Hae Won Park[3], Yuzhe Yang[1], Xuhai Xu[1], Yilun Du[1], Shwetak Patel[1], Tim Althoff[1], Daniel McDuff[1,†] and Xin Liu[1,†]
[1]Google Research, [2]Google DeepMind, [3]Massachusetts Institute of Technology, [†]Corresponding Author

Agents, language model-based systems capable of reasoning, planning, and acting are widely adopted in real-world tasks, yet how their performance changes as these systems scale across key dimensions remains underexplored. We introduce quantitative *scaling principles* for agent systems as a predictive model, capturing how performance varies with coordination, model capability, and measurable system and task factors. Across 260 configurations spanning six agentic benchmarks, five canonical architectures (Single-Agent and four Multi-Agent: Independent, Centralized, Decentralized, Hybrid), and three LLM families, we perform controlled evaluations standardizing tools, prompts, and compute to isolate architectural effects. The resulting model achieves a cross-validated R^2=0.373 across all six benchmarks (R^2=0.413 with a task-grounded capability metric). We identify a robust capability-saturation effect and

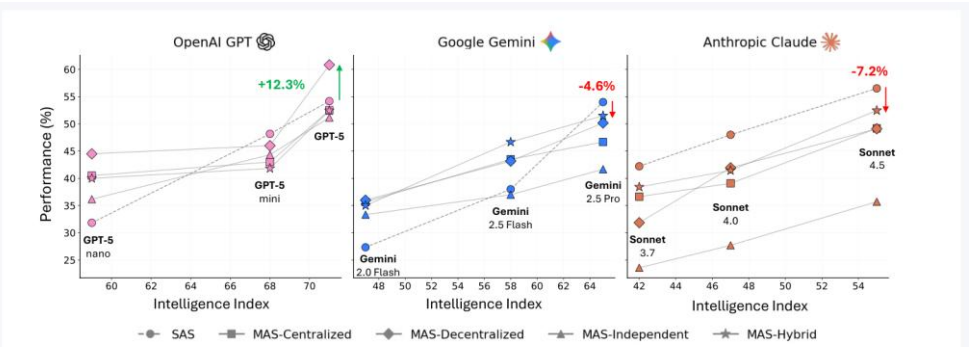
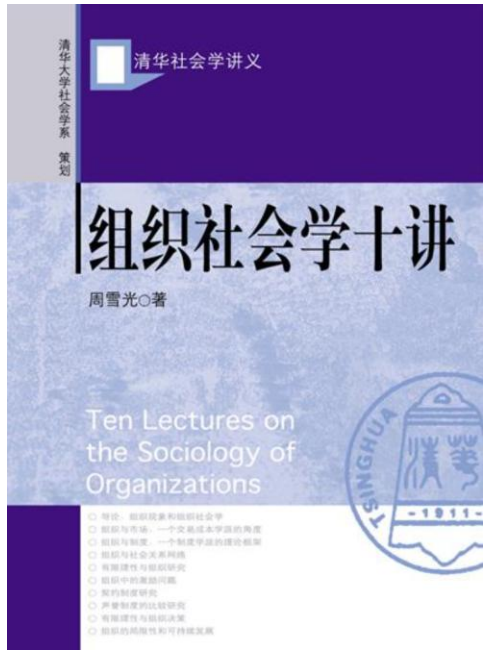


Figure 1 | **Agent Scaling across model intelligence and system topologies.** Average performance

Google Research / Agent Systems



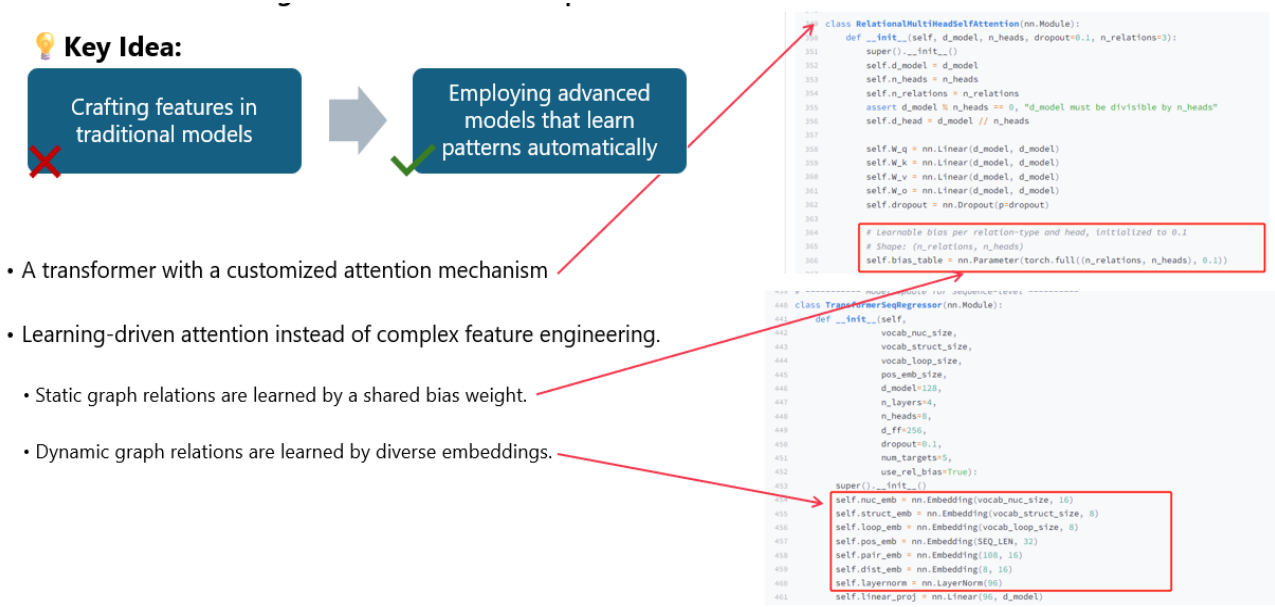
参考：Google Research, Towards a Science of Scaling Agent Systems；frontier engineering practice.

教训2：迷信基模能力与前沿文献的“银弹”

Too much digital prior, Too less real-world belief，Too expensive storytelling-tax

- 正向经验：真实日志和失败样例，比静态RAG和抽象行业知识更可靠。
- 负向经验：前沿论文常有复现偏差与叙事成本，不能替代场景内试错。





参考：Google Research, Towards a Science of Scaling Agent Systems；frontier engineering practice.

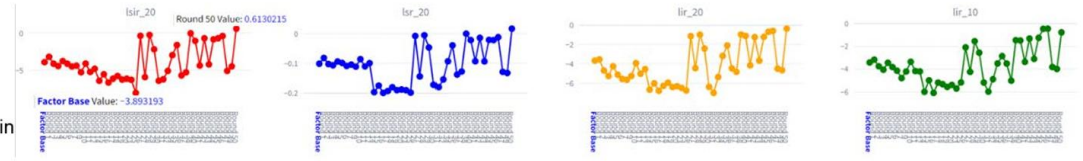
教训3：迷信自进化步数的“银弹”

Agent for factor mining- 长程自进化的低劣循环：

在上千步自循环迭代后，找到的交易因子高度趋同，没有新的信息增量

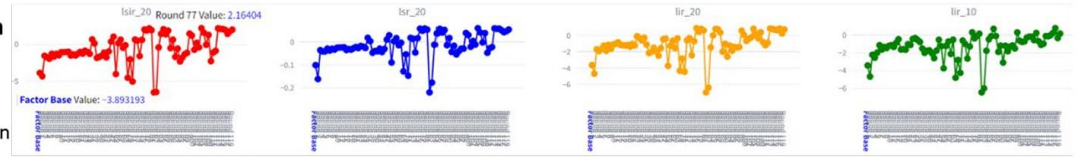
qwen 3.6-max

lsir_20_max = 0.61
rounds = 50
time per round = 12.6min
大模型瓶颈明显



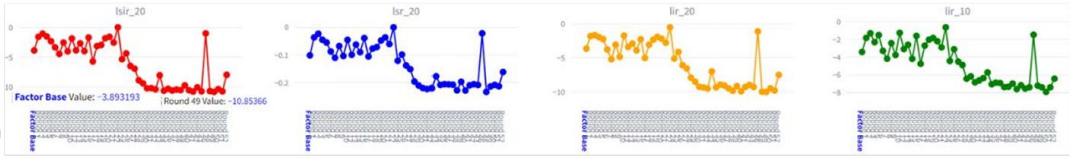
deepseek v4-flash

lsir_20_max = 2.16
rounds = 118
time per round = 5.4min
持续优化，效果最好



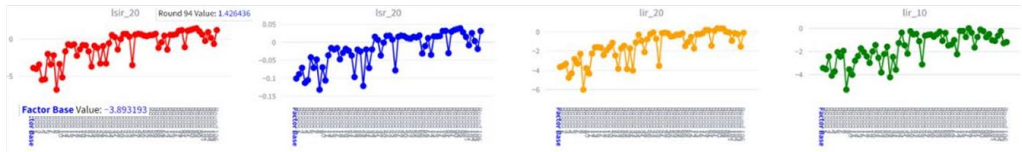
deepseek v4-pro

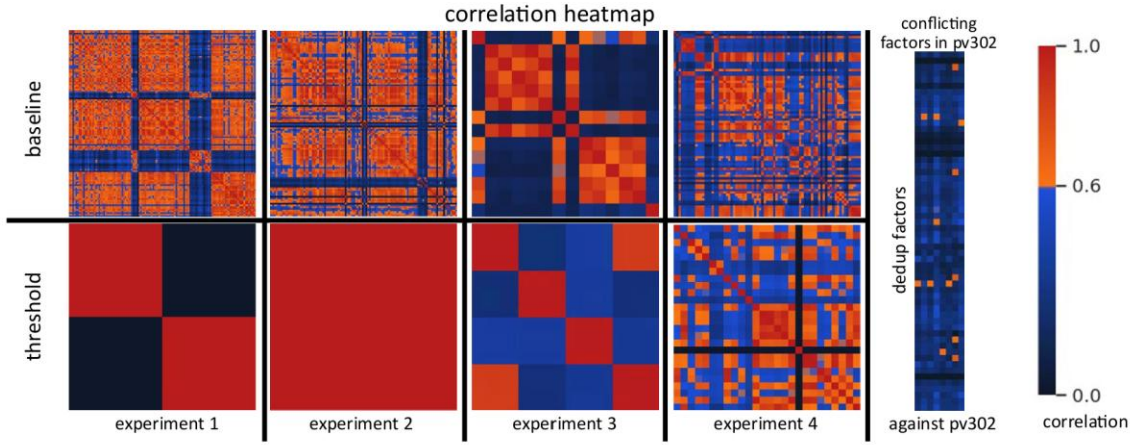
lsir_20_min = -10.85
rounds = 52
time per round = 11min
LLM误解，持续“优化”

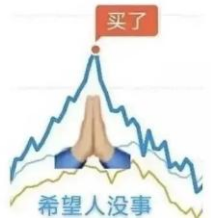


gpt5.2

lsir_20_max = 1.42
rounds = 106
time per round = 6.0min
持续优化，效果较好







第一性原理：尊重并充分利用领域里已有的工具、仿真器和小模型

在工业原生场景中，最重要的不是重写专业系统，而是组织已有小模型、工具与模拟器。



- MLE-Bench 中的核心动作，是训练、选择和集成合适的小模型。
- 金融与电池场景的核心对象，是场景特异的回测器或物理模拟器。
- 重复造轮子的代价很高：仿真器、CAD/CAE工具、因子库不应由通用模型临时重写。
- 大模型的角色更接近系统承载层：调度、记忆、诊断、验证与接口转换。

参考：R&D-Agent / R&D-Agent-Quant / Battery-Sim-Agent；Anthropic, Building Effective Agents.

Notes:

- Agent应用领域需要的是更多的新场景和新问题，而不是重复发明/雕花新名词的trivial solution
- Agent Harness绝大多数是工程问题，想不被基模吃掉，需结合场景提炼更多真科学问题
- 真实世界工程&科学场景的异质性，会定义下一代Agent乃至基模的边界
- 不要迷信任何 “sexy-word-driven”的银弹：Skill/自进化/RSI/…

Thanks!

邮箱：**fsk@zju.edu.cn**
知乎/小红书：方轩固
微信：**xuangu1996 (请备注信息)**