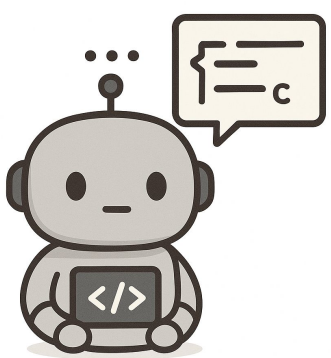
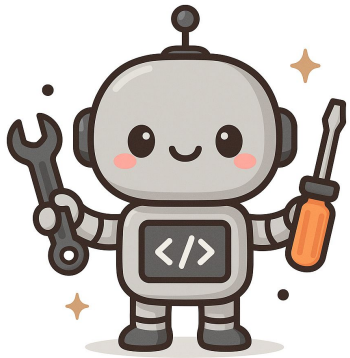
BOAD: Discovering Hierarchical Software Engineering Agents via Bandit Optimization

Iris Xu · Guangtao Zeng · Zexue He · Charles Jin · Aldo Pareja · Dan Gutfreund · Chuang Gan · Zhang-Wei Hong

From LLMs to Autonomous SWE Agents

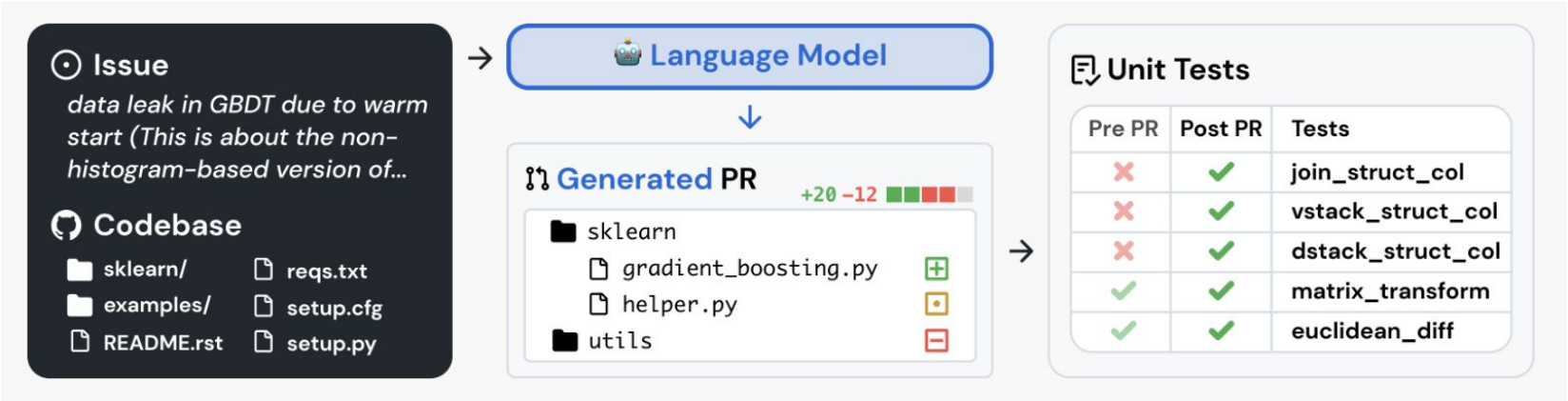


Code Completion



Autonomous Code Agents

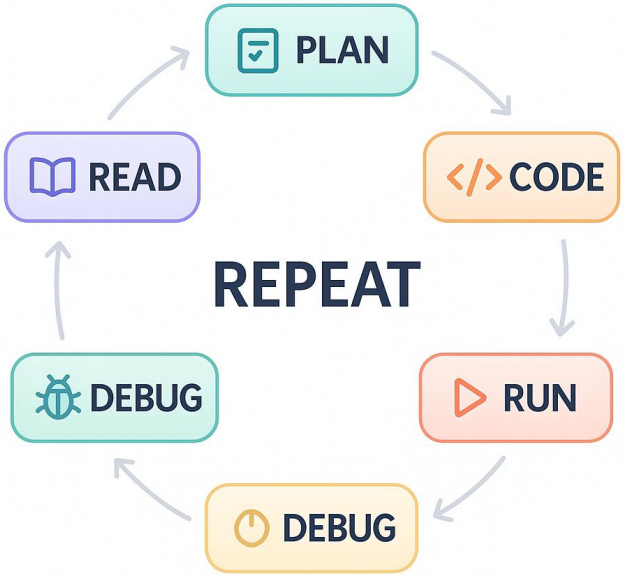
SWE-Bench



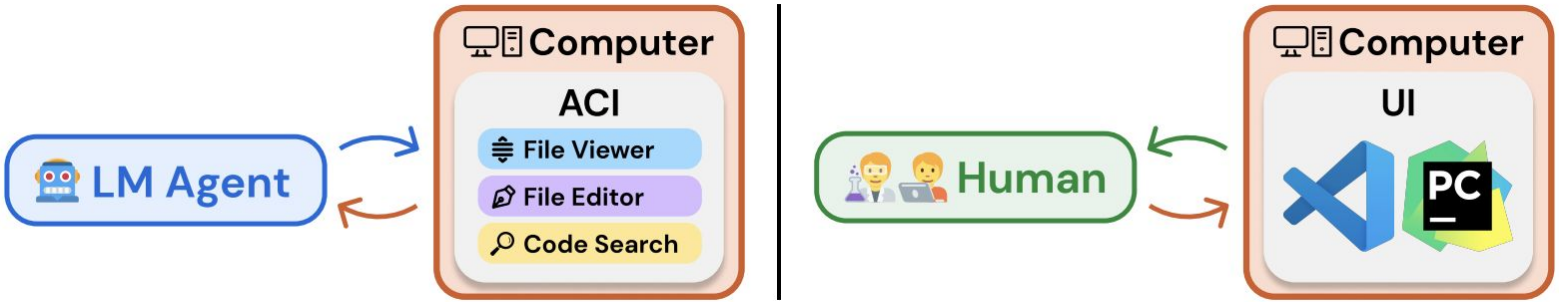
SWE-bench: Can Language Models Resolve Real-World GitHub Issues? (Yang et al., 2023)

SWE-bench is hard for LLMs!

- Lots of context
- Long-horizon



SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering



SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering (Yang et al., 2024)

System Prompt

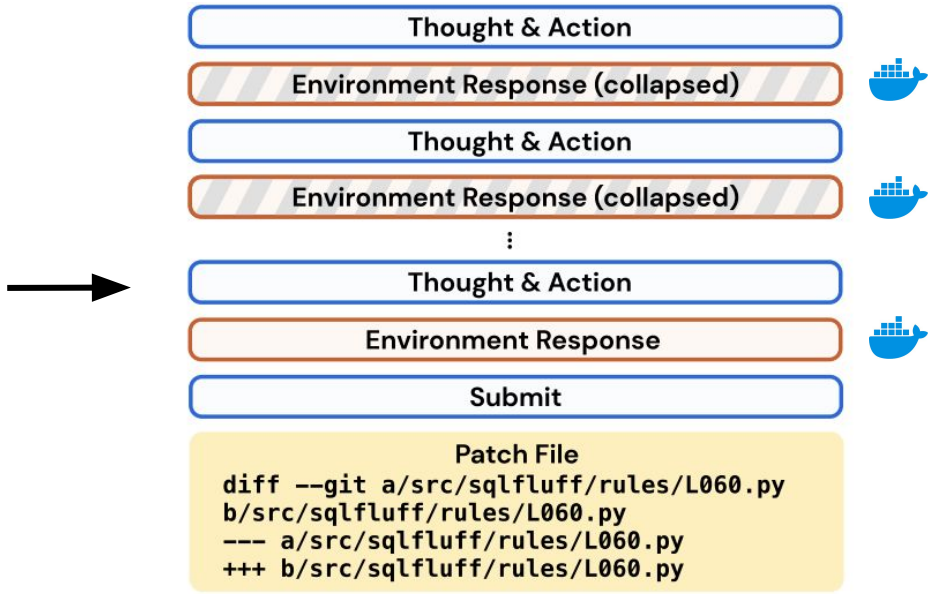
- Describe environment and commands
- Specify response format

Demonstration

Full trajectory of a successful example

Issue statement

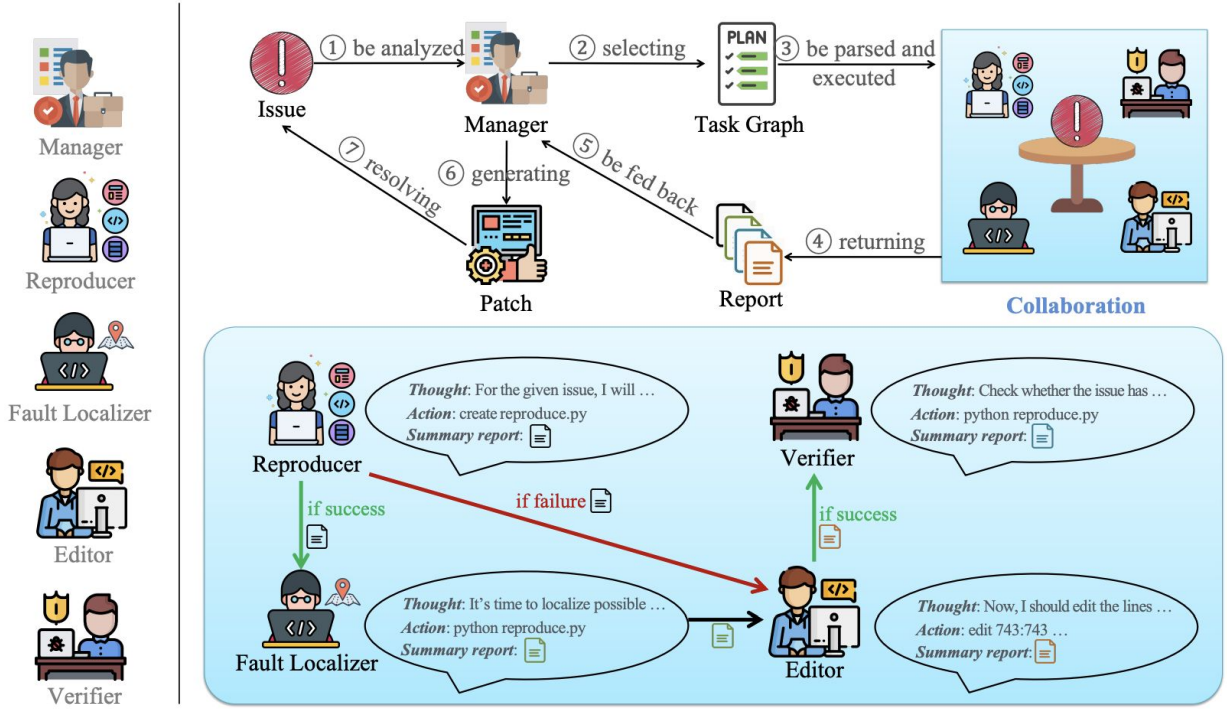
- Give reported issue description
- Instructions to resolve issue
- High-level strategy tips



Long context is difficult for single-agent!

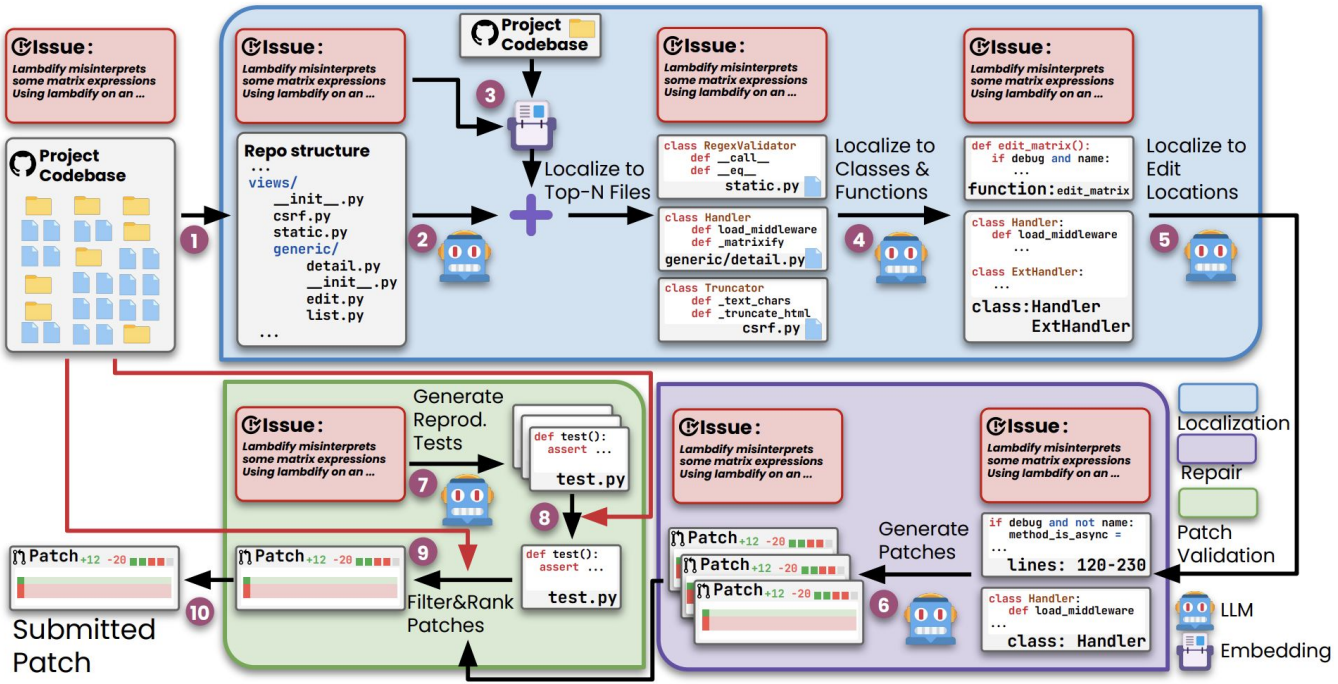
SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering (Yang et al., 2024)

Multi-agent systems



CodeR: Issue Resolving with Multi-Agent and Task Graphs (Chen et al., 2024)

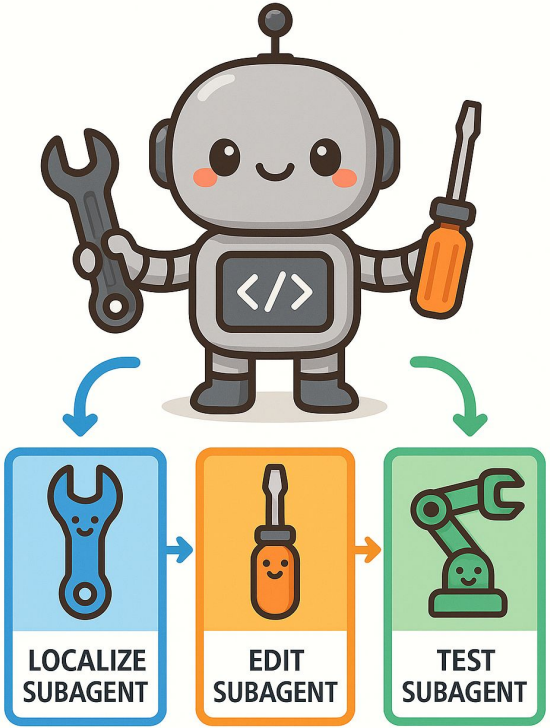
Multi-agent systems



Agentless: Demystifying LLM-based Software Engineering Agents (Xia et al., 2024)

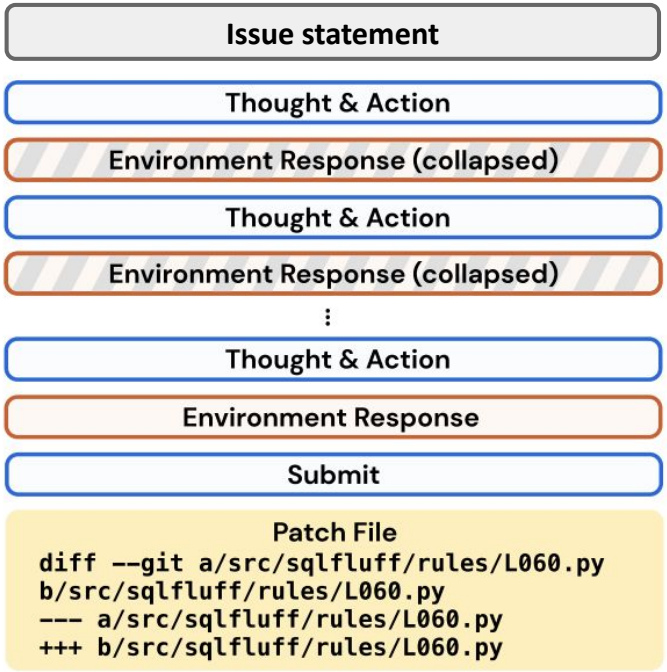
<u>Idea:</u> sub-agents

- Task decomposition
- Main agent interfaces with sub-agents as tools

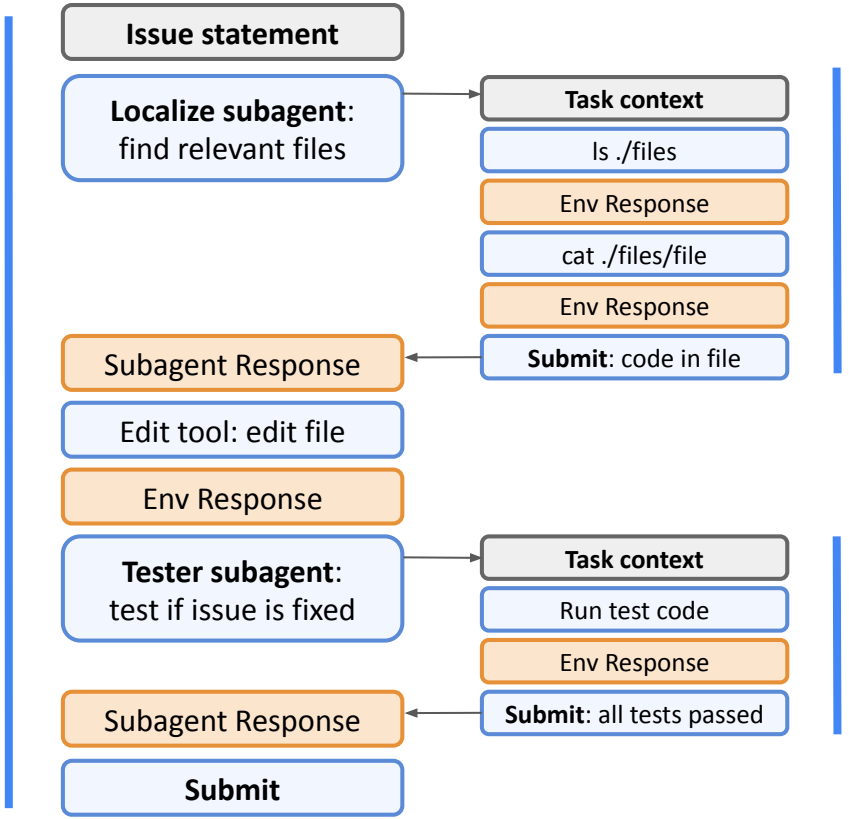


Claude code also uses subagents!

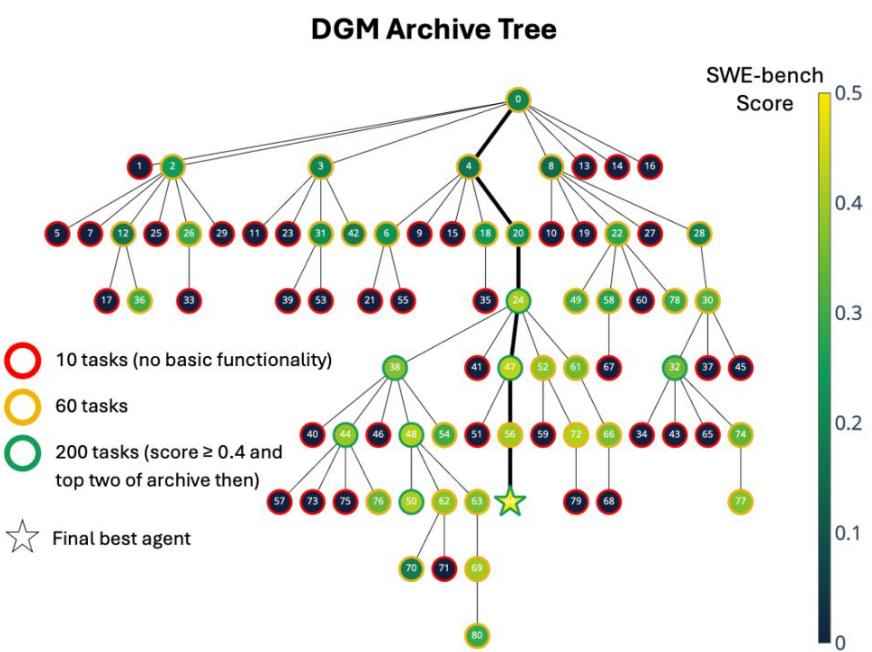
Only main-agent

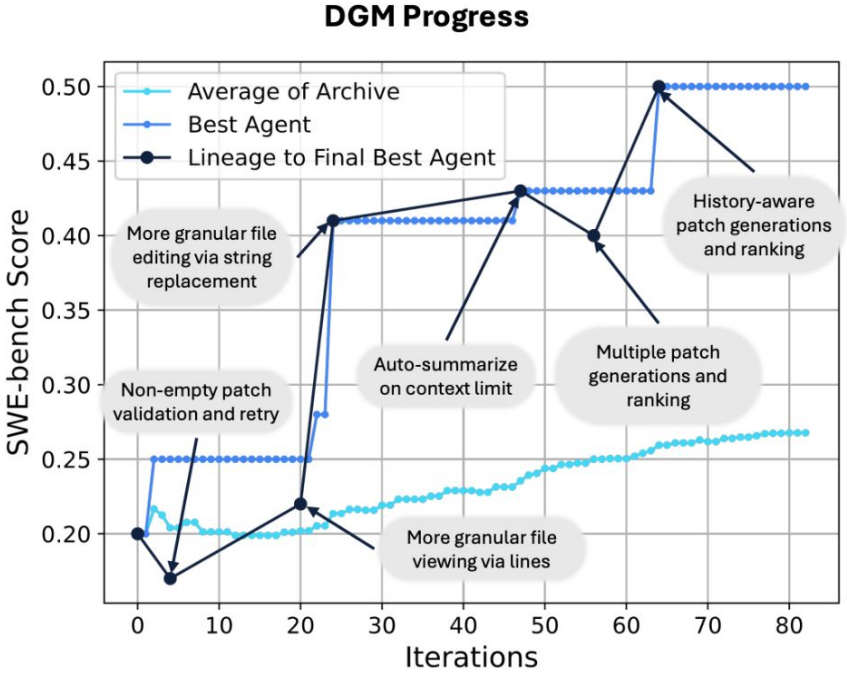


With sub-agents



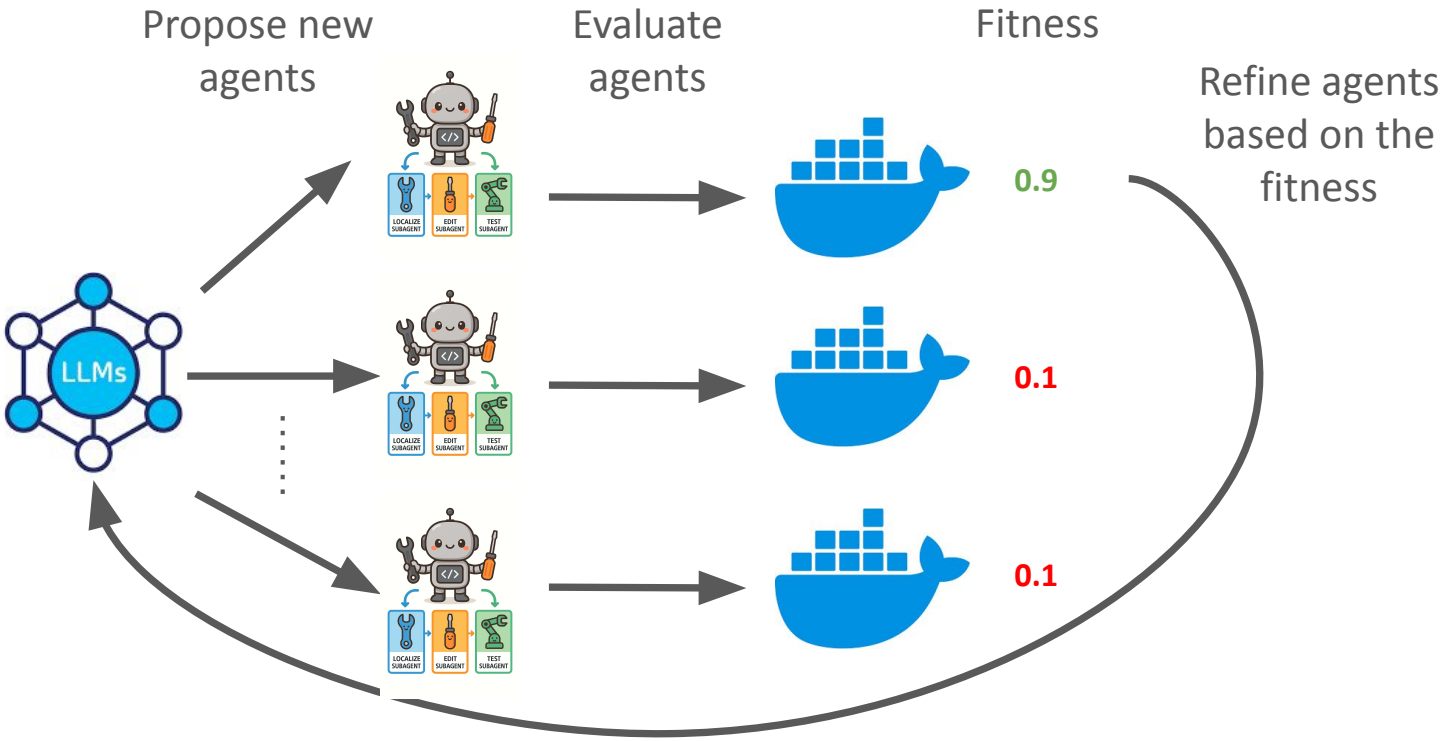
Evolutionary Search for Agent Design- redo diagram



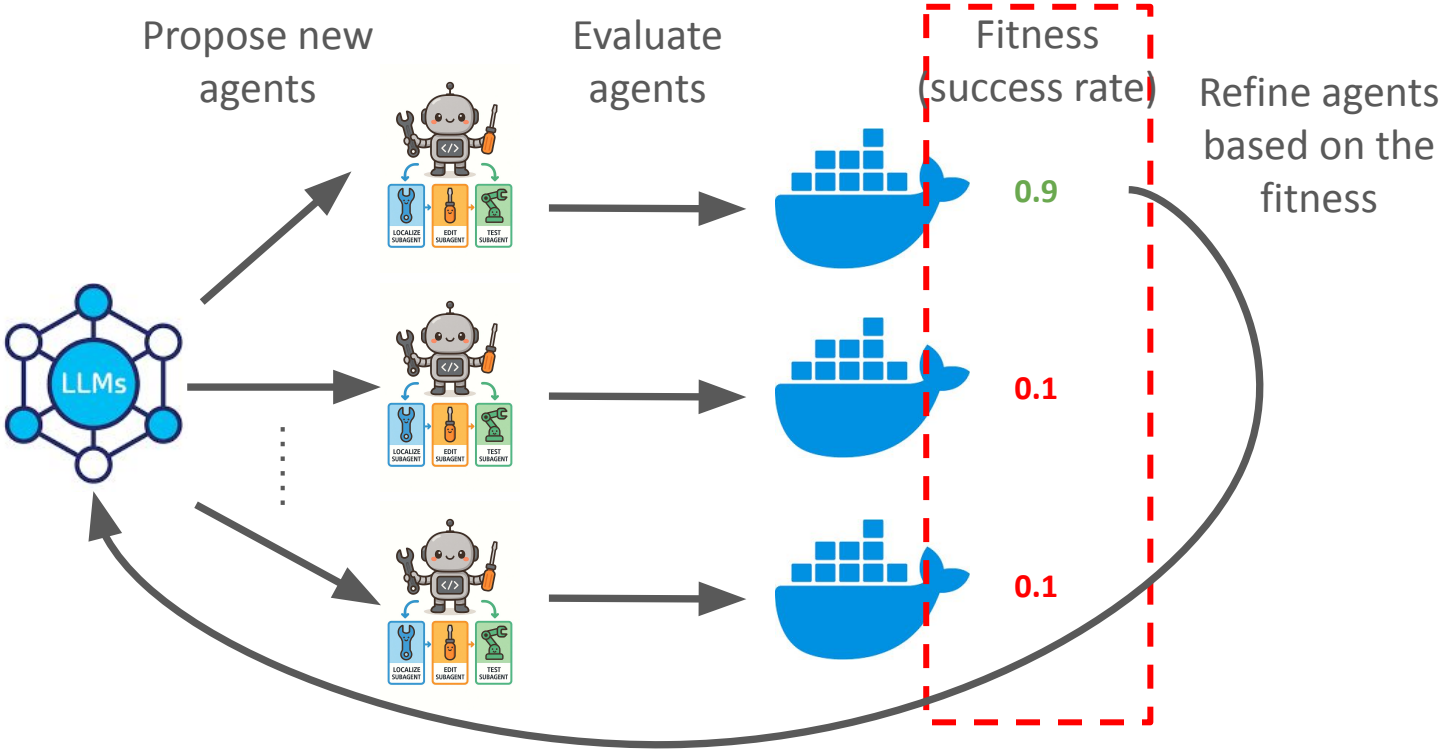


Darwin Gödel Machine: Open-Ended Evolution of Self-Improving Agents (Zhang et al., 2025)

Prior works: Evolutionary search / Self-refinement

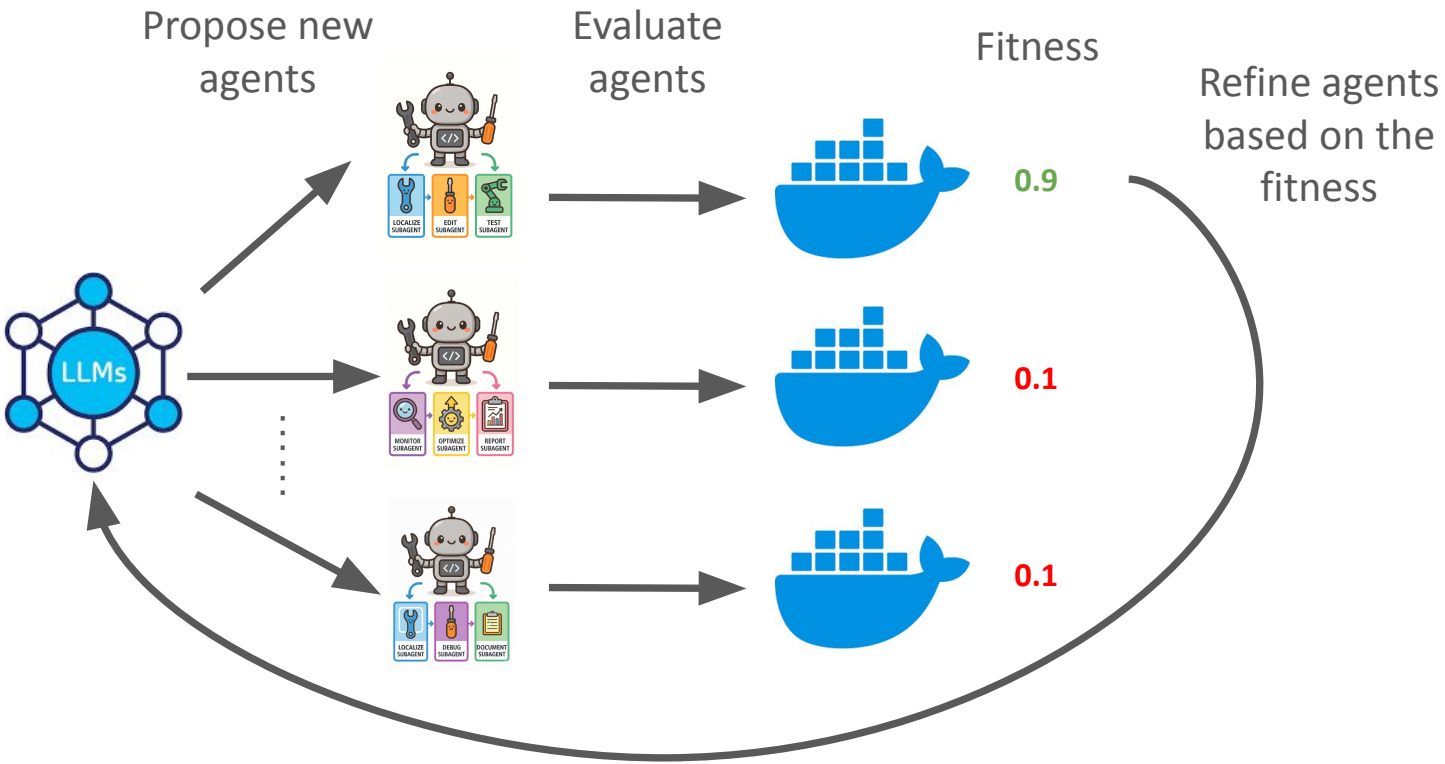


Limitation: Need large number of samples to find good sub-agents



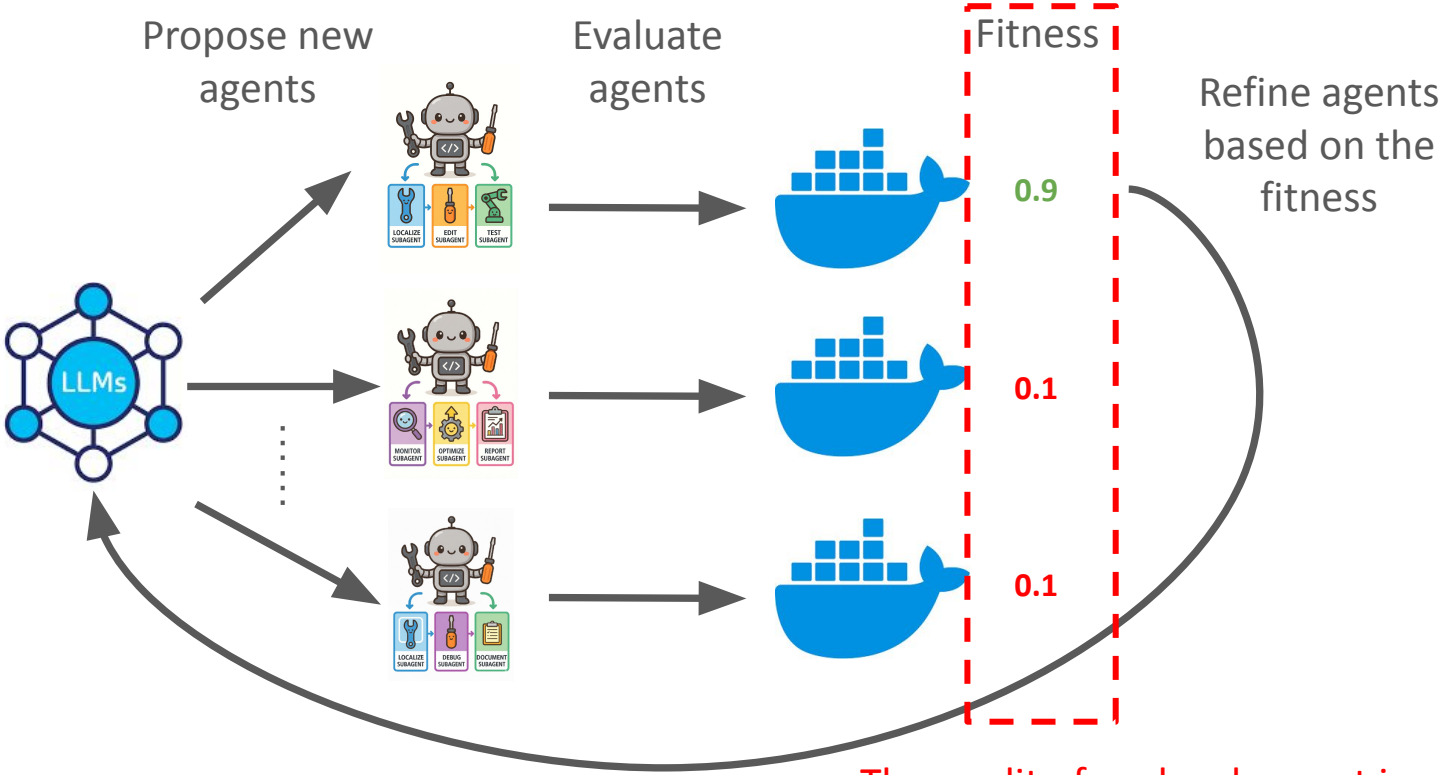
The credit of each sub-agent is noisy

Prior works: Evolutionary search / Self-refinement



Darwin Gödel Machine: Open-Ended Evolution of Self-Improving Agents (Zhang et al., 2025)
Automated design of agentic systems (Hu et al., 2024)

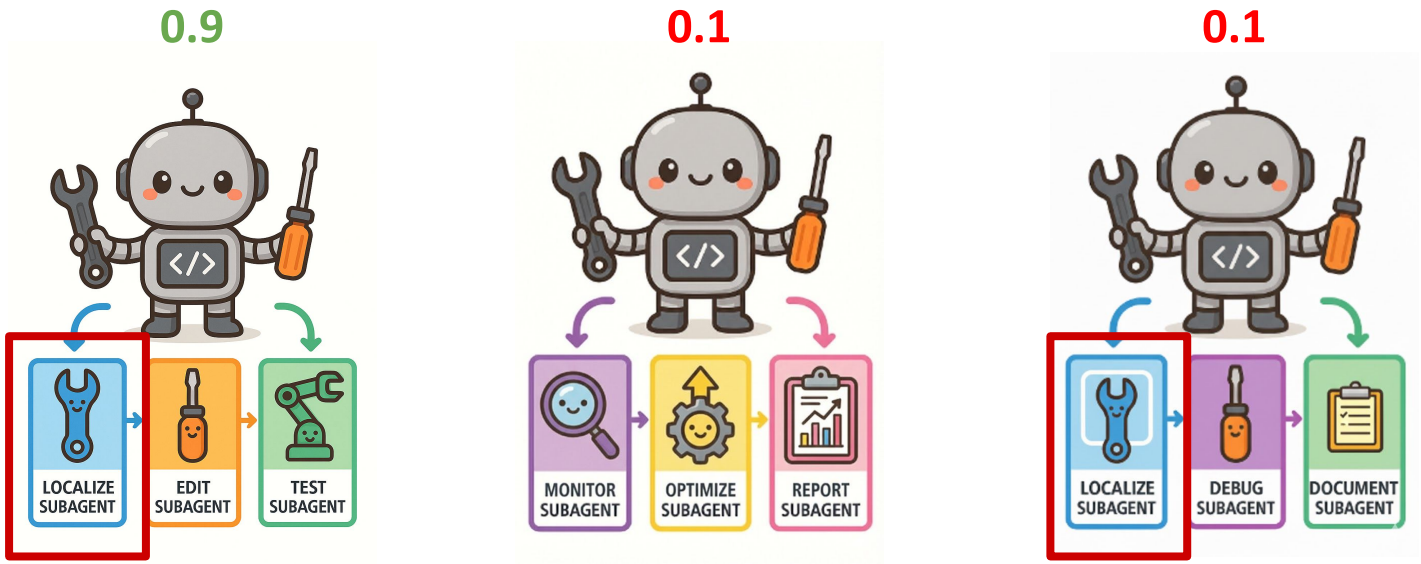
Prior works: Evolutionary search / Self-refinement



The credit of each sub-agent is noisy

Problem: Credit assignment is noisy

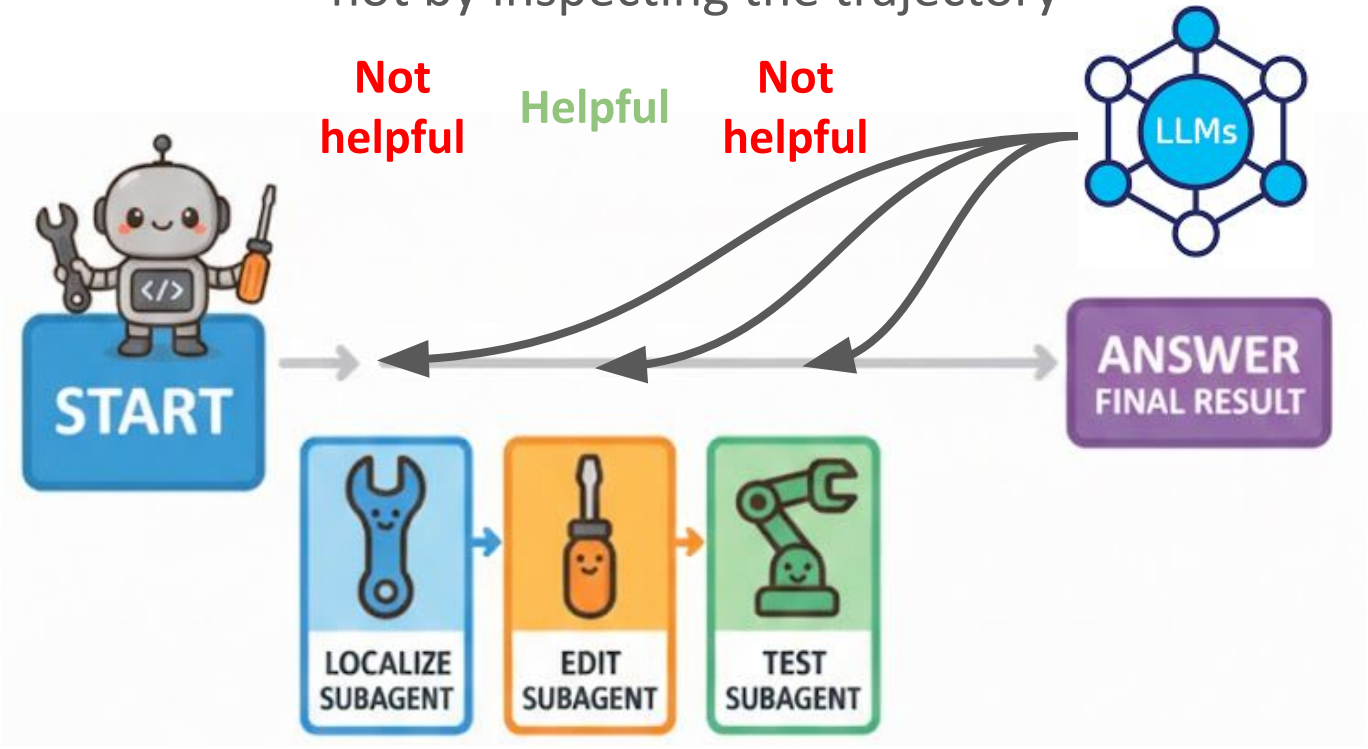
The same sub-agent can appear in both good and bad teams.



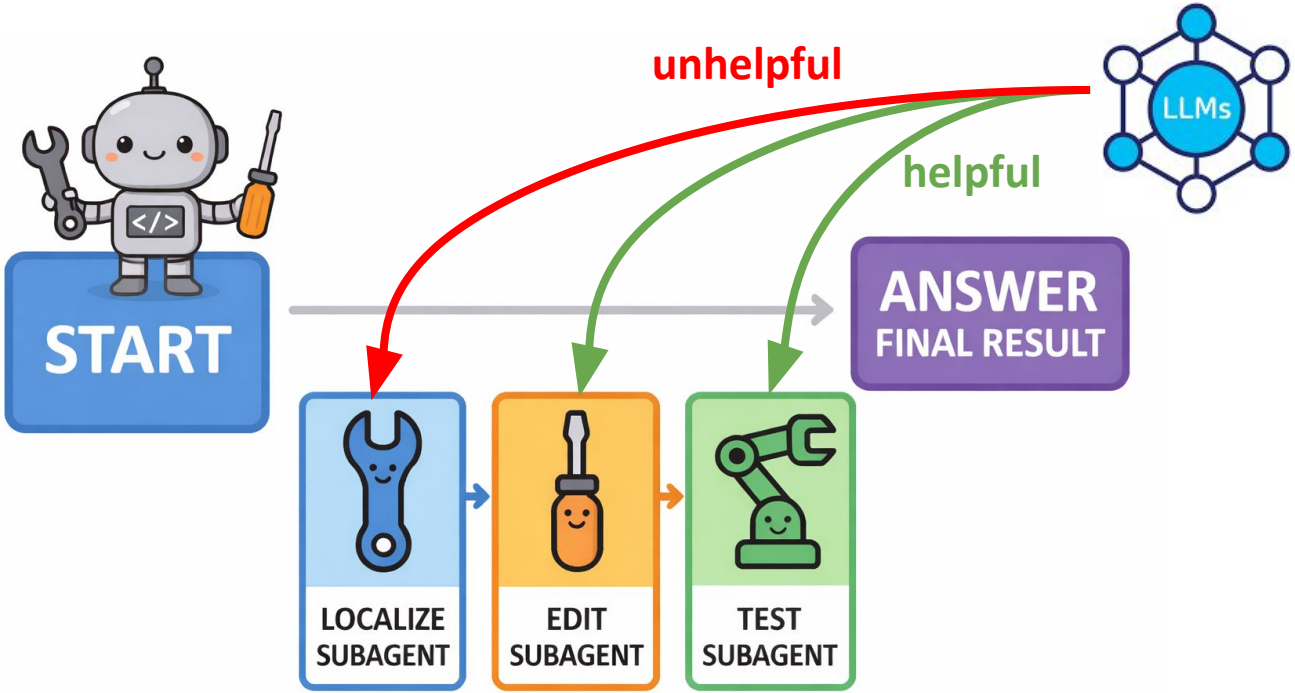
Reducing noise needs a large sample size

Our insight: Hindsight credit assignment

Judge if a sub-agent contributes to the result or not by inspecting the trajectory



Our Insight: Hindsight Credit Assignment



Judge if a sub-agent contributes to the result or not by inspecting the trajectory

How to select sub-agents

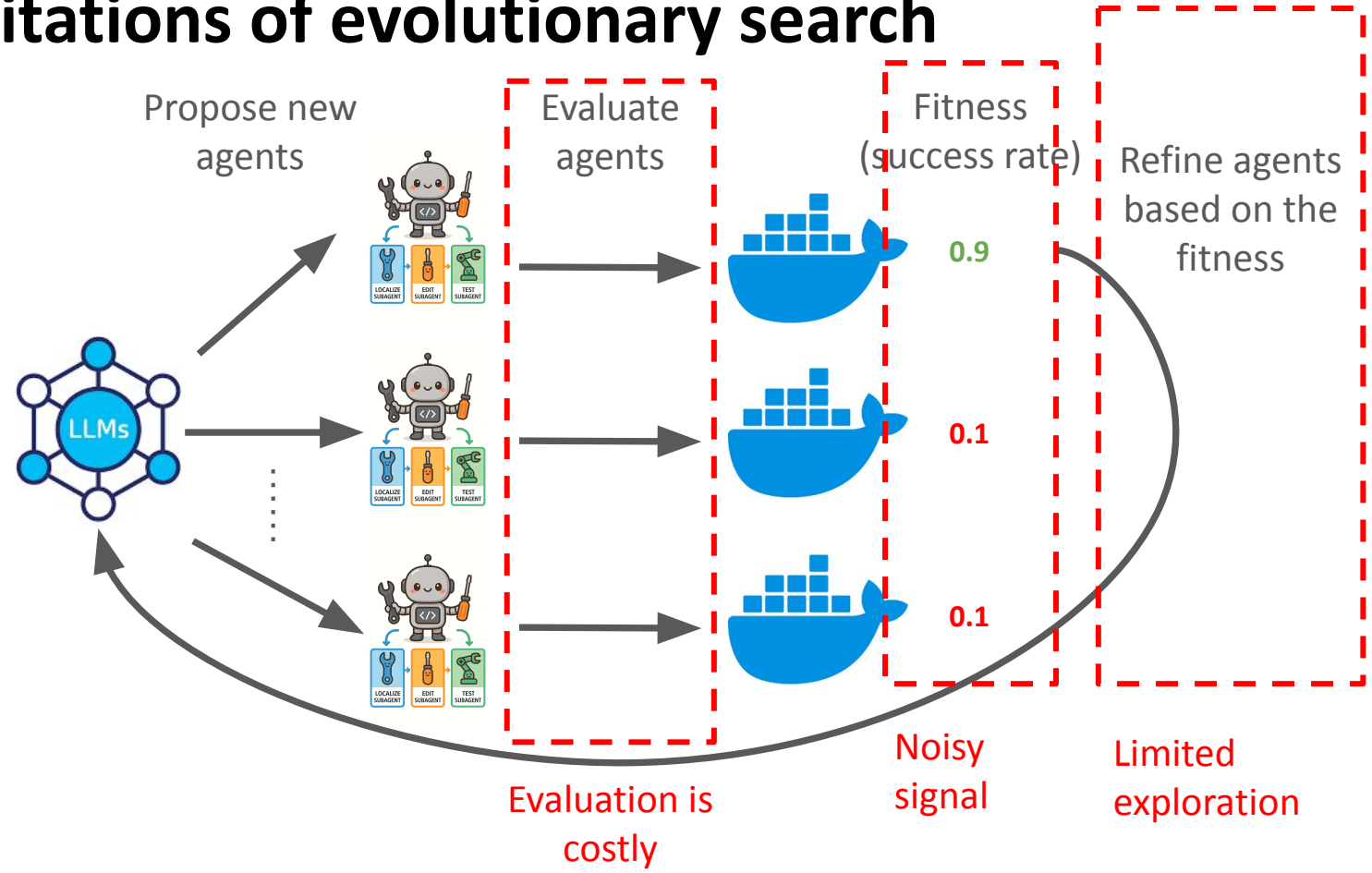
- Choosing a sub-agent to equip ⇔ action in RL
- Each sub-agent ⇔ “arm” (RL bandit problem)
- Exploration-exploitation tradeoff
  - Upper Confidence Bound (UCB):

$$\mathrm{UCB}_\omega(t) = \hat{\mu}_\omega(t) + \sqrt{\frac{2\ln t}{n_\omega(t)}}.$$

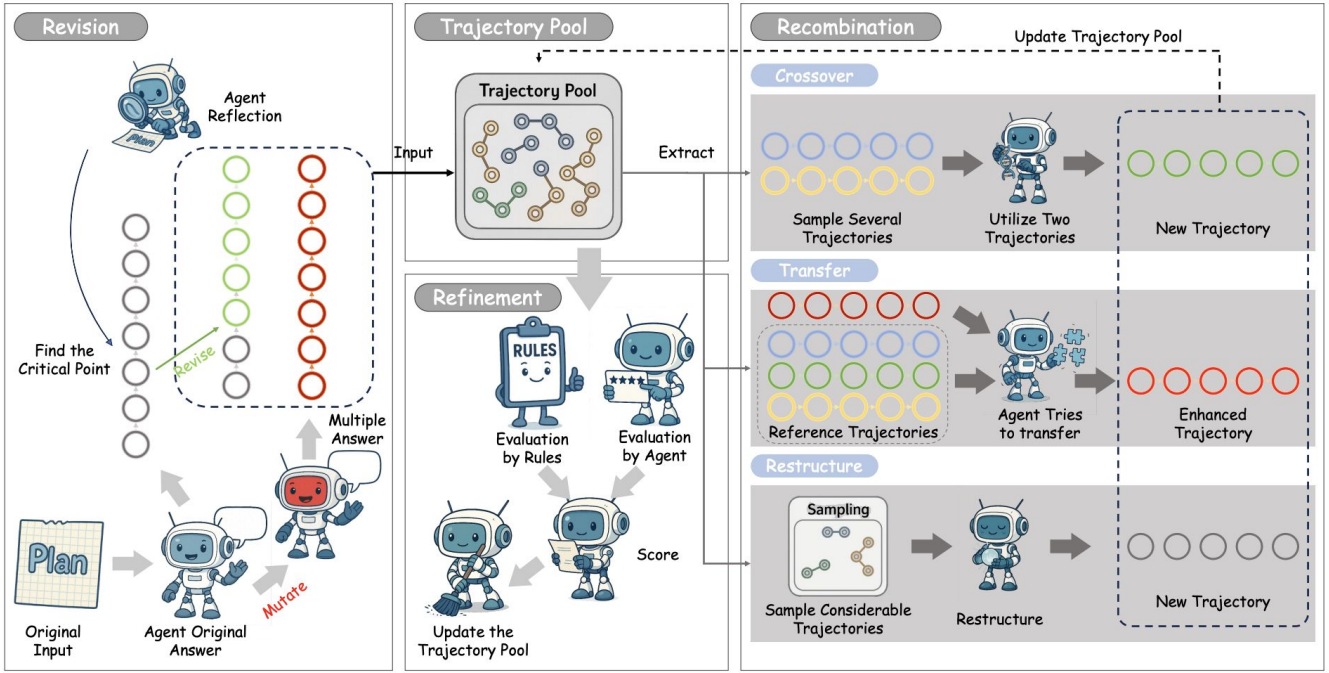
exploitation

exploration

Limitations of evolutionary search



Evolutionary Search for Agent Design



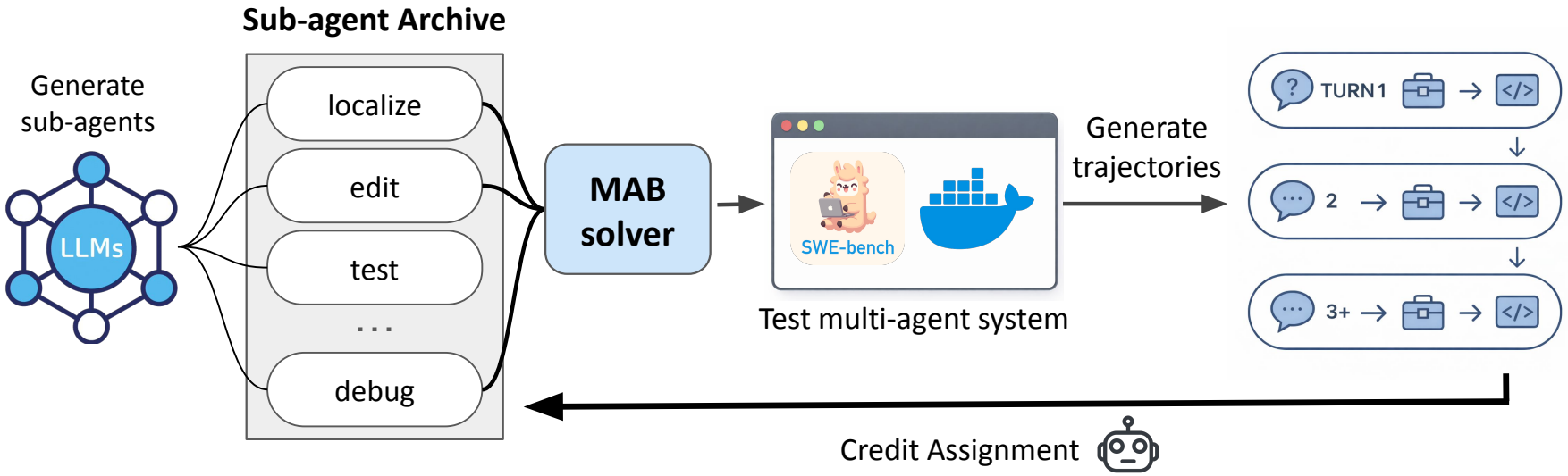
SE-Agent: Self-Evolution Trajectory Optimization in Multi-Step Reasoning with LLM-Based Agents (Lin et al., 2025)

Motivation

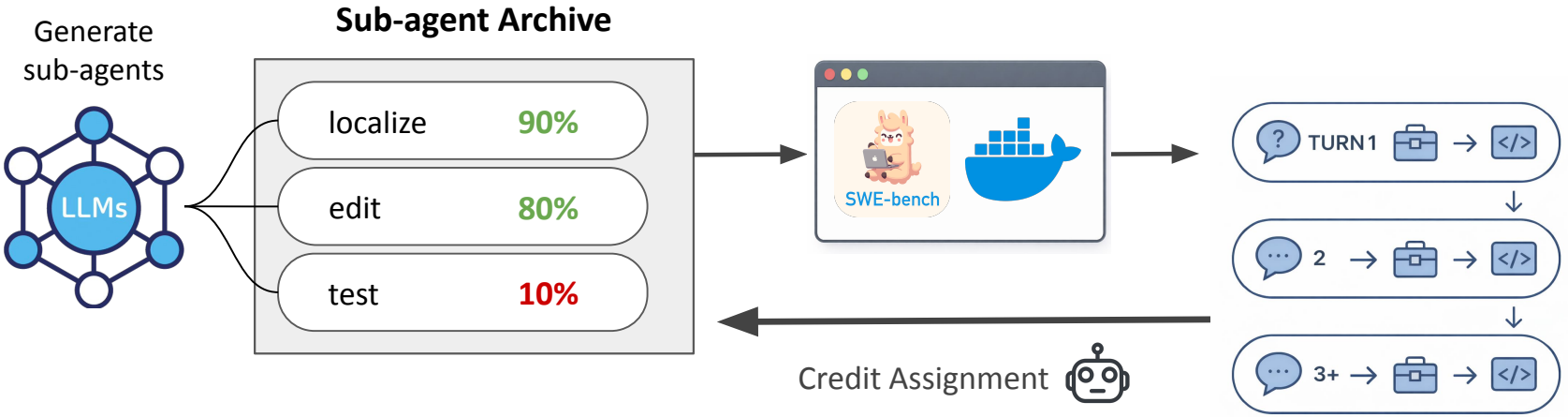
- Need more efficient way to explore/exploit
- Need to be able to do better hindsight credit assignment

Idea 2: Agent design as a multi-armed bandit, with each sub-agent as an arm

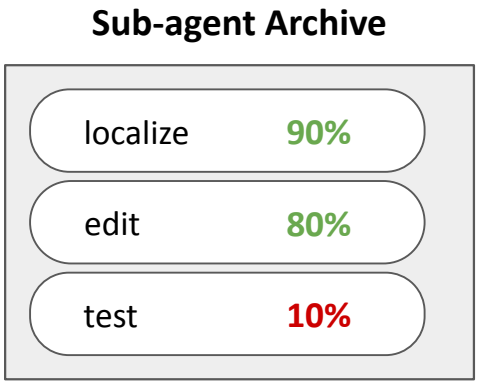
Bandit Optimization for Agent Design (BOAD)



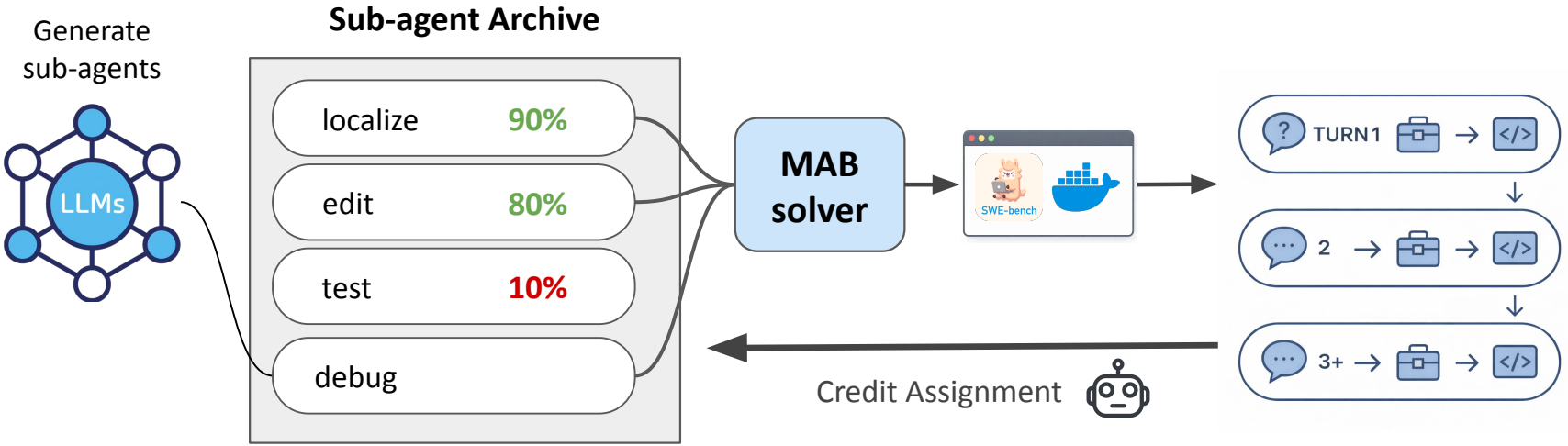
Implementation



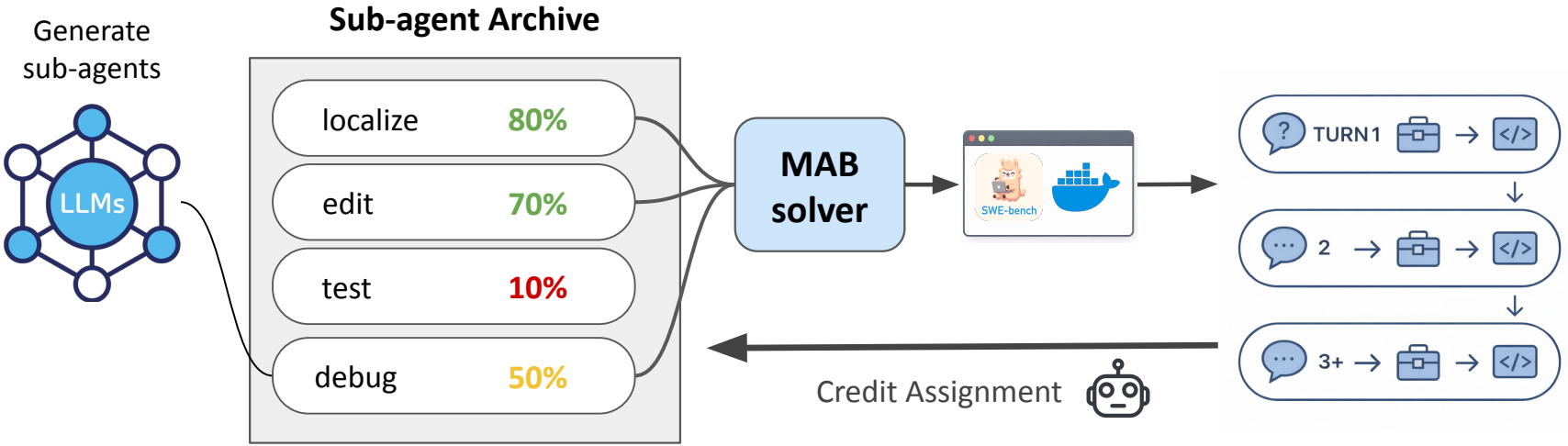
Implementation



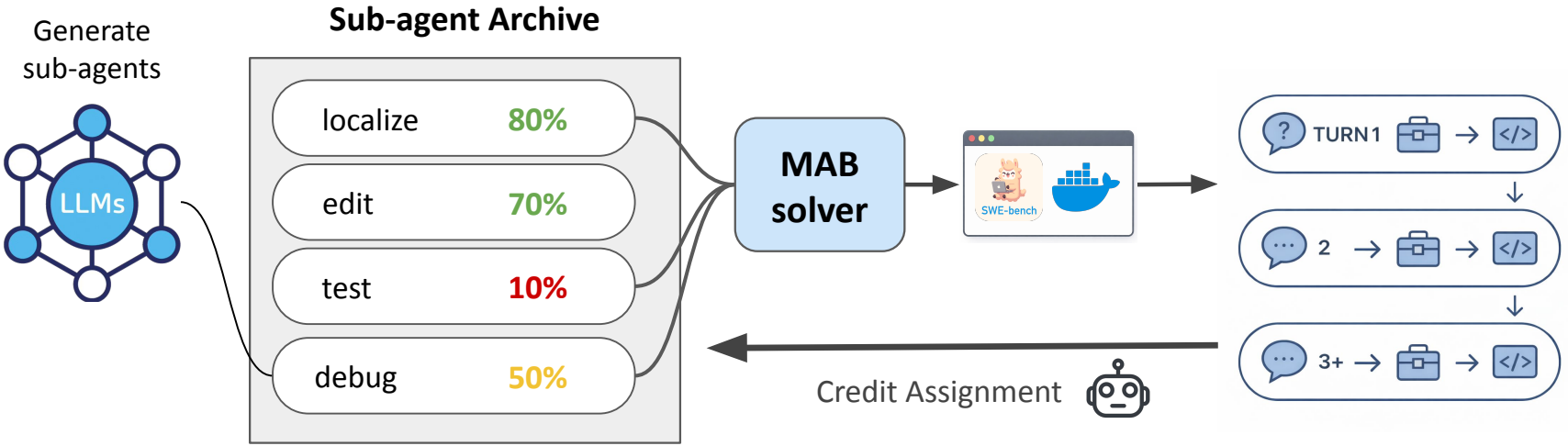
Implementation



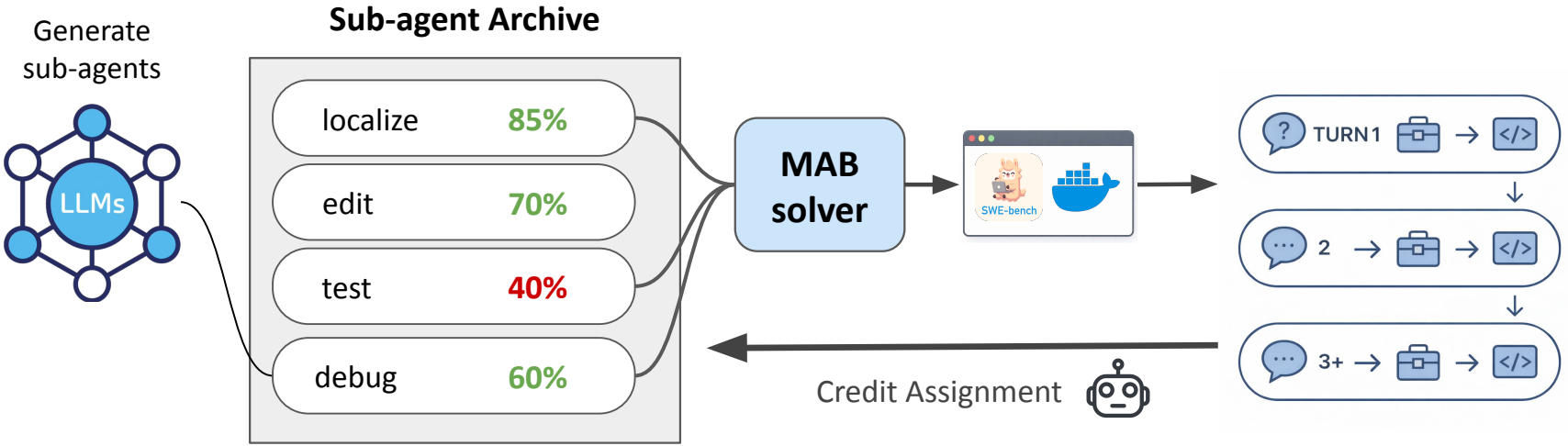
Implementation



Implementation



Implementation



When to add new sub-agents?

- Chinese Restaurant Process to create sub-agents:

$$\Pr(\text{new at } t) = \frac{\theta}{\theta + |\Gamma_{t-1}|}$$

size of archive

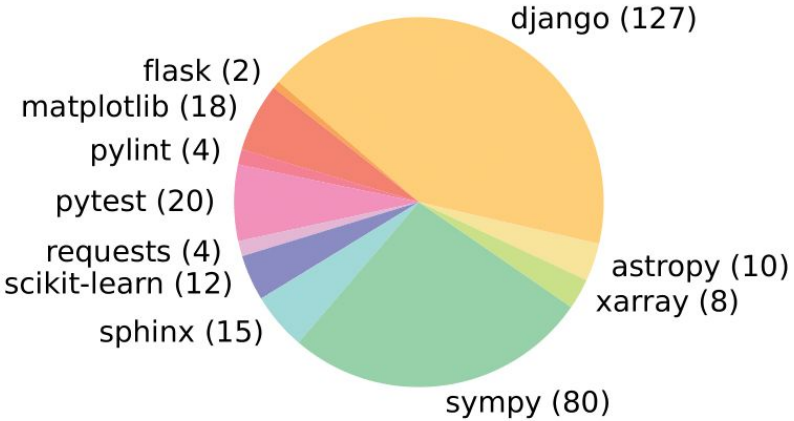
- As archive size increases: less chance to add new sub-agent

Results

Scale	Model	Scaffold	Verified Resolved (%)
Large	**GPT-4o** [50]	SWE-agent	23.0
	GPT-4o [49]	Agentless	38.8
	Claude 3.5 Sonnet [5]	Agentless	50.8
	Claude 3.5 Sonnet [5]	OpenHands	53.0
	Claude 3.7 Sonnet [4]	SWE-agent	62.4
	Claude 4.0 Sonnet [6]	SWE-agent	66.8
	Claude 4.0 Sonnet [6]	OpenHands	70.4
	DeepSeek-R1 [18]	Agentless	49.2
	DeepSeek-V3 [15]	Agentless	42.0
	GLM-4.5-Air [43]	OpenHands	57.6
	GLM-4.5-Air [43]	SWE-Agent	–
	Qwen3-Coder 480B/A35B Instruct [45]	OpenHands	69.6
Small	**Qwen3-Coder-30B-A3B-Instruct** [45]	SWE-agent	–
	Qwen3-Coder-30B-A3B-Instruct [45]	OpenHands	51.6
	Devstral-Small-2505 [7]	OpenHands	46.8
	Seed-OSS-36B-Instruct [44]	SWE-agent (baseline)	49.8
	Seed-OSS-36B-Instruct [44]	SWE-agent + Manual Subagent	47.4
	Seed-OSS-36B-Instruct [44]	SWE-agent + **BOAD**	**53.2**[2]

Limitations of SWE-bench

- Static issues
- New models may have “seen” the repository
- Or even the solutions!



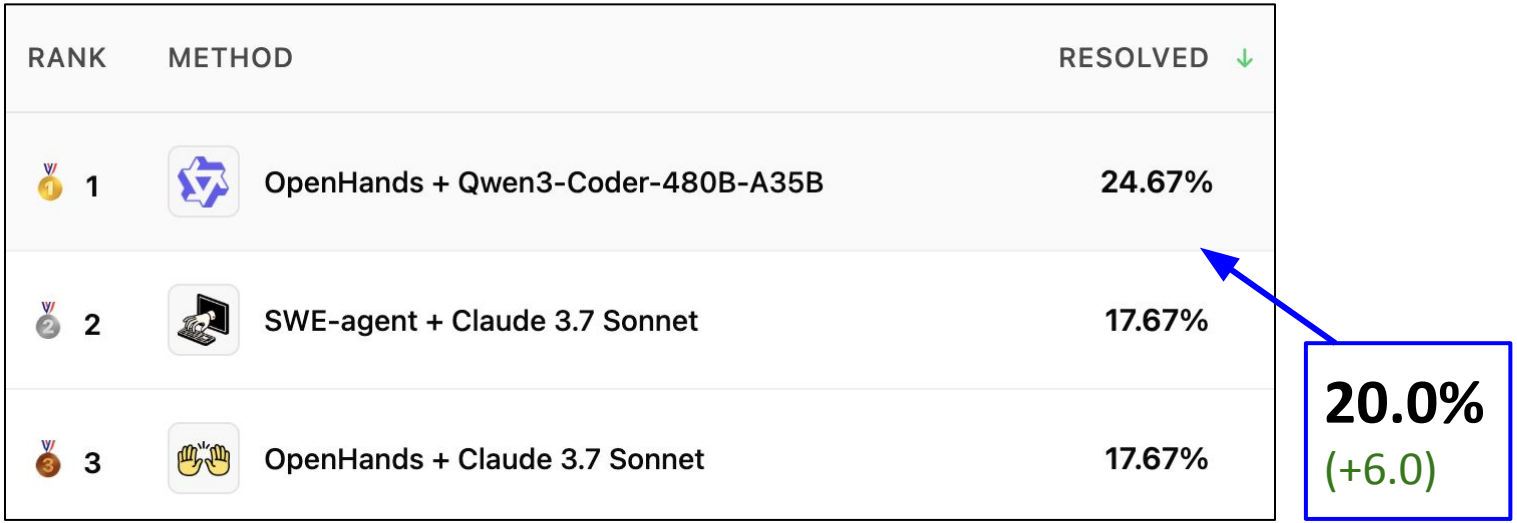
SWE-bench: Can Language Models Resolve Real-World GitHub Issues? (Yang et al., 2023)

SWE-bench-Live

- Newer dataset
- Updated every month with new issues
- Out-of-distribution from our training instances

SWE Bench Goes Live! (Zhang et al., 2025)

Scale	Model	Scaffold	Verified Resolved (%)	Live Resolved (%)
Large	**GPT-4o** [50]	SWE-agent	23.0	10.0
	GPT-4o [49]	Agentless	38.8	11.7
	Claude 3.5 Sonnet [5]	Agentless	50.8	–
	Claude 3.5 Sonnet [5]	OpenHands	53.0	–
	Claude 3.7 Sonnet [4]	SWE-agent	62.4	13.7[1]
	Claude 4.0 Sonnet [6]	SWE-agent	66.8	–
	Claude 4.0 Sonnet [6]	OpenHands	70.4	–
	DeepSeek-R1 [18]	Agentless	49.2	–
	DeepSeek-V3 [15]	Agentless	42.0	13.3
	GLM-4.5-Air [43]	OpenHands	57.6	–
	GLM-4.5-Air [43]	SWE-Agent	–	17.7
	Qwen3-Coder 480B/A35B Instruct [45]	OpenHands	69.6	24.7
Small	**Qwen3-Coder-30B-A3B-Instruct** [45]	SWE-agent	–	17.0
	Qwen3-Coder-30B-A3B-Instruct [45]	OpenHands	51.6	–
	Devstral-Small-2505 [7]	OpenHands	46.8	–
	Seed-OSS-36B-Instruct [44]	SWE-agent (baseline)	49.8	12.3
	Seed-OSS-36B-Instruct [44]	SWE-agent + Manual Subagent	47.4	14.0
	Seed-OSS-36B-Instruct [44]	SWE-agent + **BOAD**	**53.2**[2]	**20.0**



SWE-Bench-Live Leaderboard

Token Usage Analysis

Metric	Setting	Verified	Live
Total tokens (M)	SWE-agent	0.92	1.49
	SWE-agent + BOAD	0.93 (+0.7%)	1.13 (-23.8%)
Max input tokens	SWE-agent	34.6k	49.0k
	SWE-agent+BOAD	30.5k (-11.6%)	36.7k (-25.0%)

Next Steps?

- Jointly train subagent proposer model with end-to-end RL
- Can we discover good sub-agents that are instance-specific?

Qualitative Analysis

Advantages:

- Over-editing
- Multi-site fixes and coverage