AiDD 6th 2025

2025 AI+研发数字峰会

AI+ Development Digital Summit

拥抱AI 重塑研发

05/23-24 | 上海站

AiDD 6th 2025 | 05/23-24 | 上海站

2025 AI+ Development Digital Summit

AI+研发数字峰会

拥抱AI 重塑研发

面向复杂软件工程任务的 Coding Agent

高鹏飞 | 字节跳动

高鹏飞

字节跳动 Trae Research Team

上海科技大学计算机科学与技术工学博士学位，导师宋富研究员。曾访问新加坡管理大学，导师孙军教授。主要研究方向为利用形式化方法保障软件的安全性和可靠性。博士期间共发表高水平论文10篇，包括一作身份发表ACM TOSEM 3篇，IEEE TSE 1篇，OOPSLA 2024 1篇，荣获教育部博士研究生国家奖学金，上海市优秀毕业生，2023年CCF 形式化专委优秀博士论文激励计划等荣誉。在字节跳动MarsCode Research Team，负责 Coding Agent 的前沿探索和落地

- 开源 & 闭源模型特性 / SWE Agent 模型训练
- Agent 算法的优化
- Vibe Coding 下的软件工程新范式

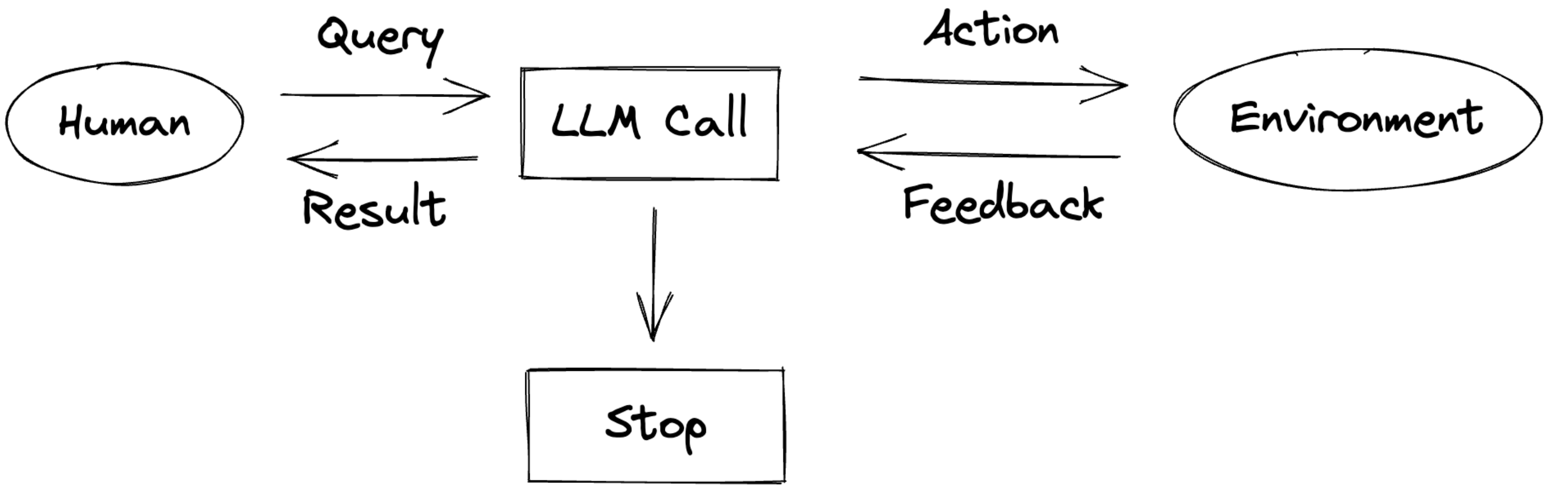
目录
CONTENTS

PART 01

背景

Agent

- LLM-based Agent
  - 进行 tool call
  - 环境交互
  - 复杂任务
- 与 workflow 的区别
  - 是否进行自主决策

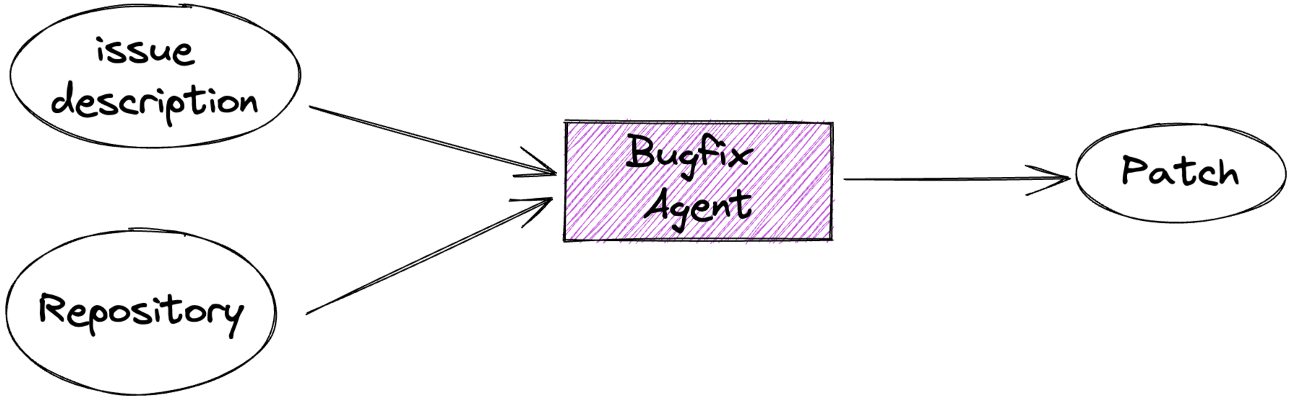


"Agents are models using tools in a loop"

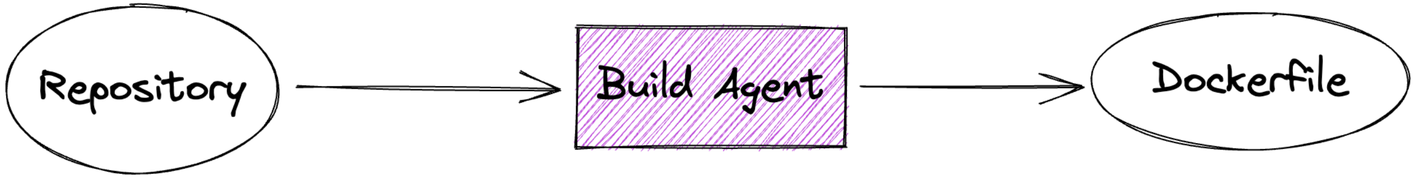
--------- *Hannah Moran, Anthropic*

面向复杂软件工程任务的 Coding Agent

- Bugfix Agent

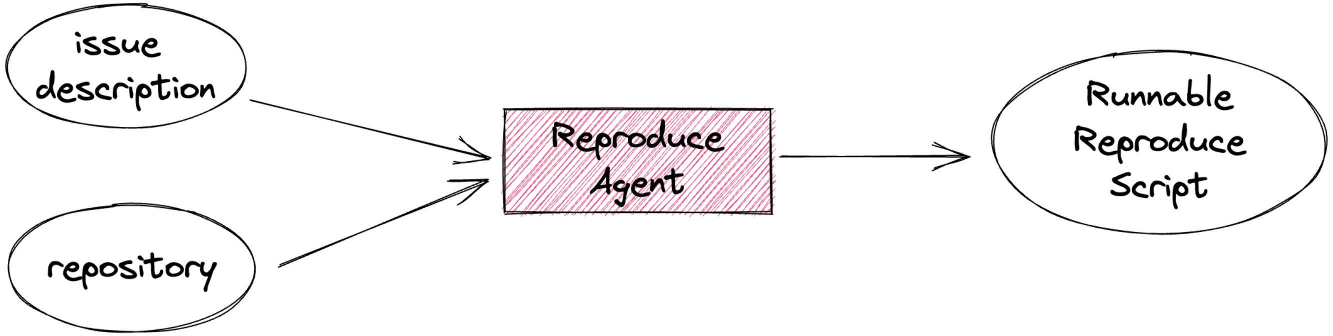


- Build Agent

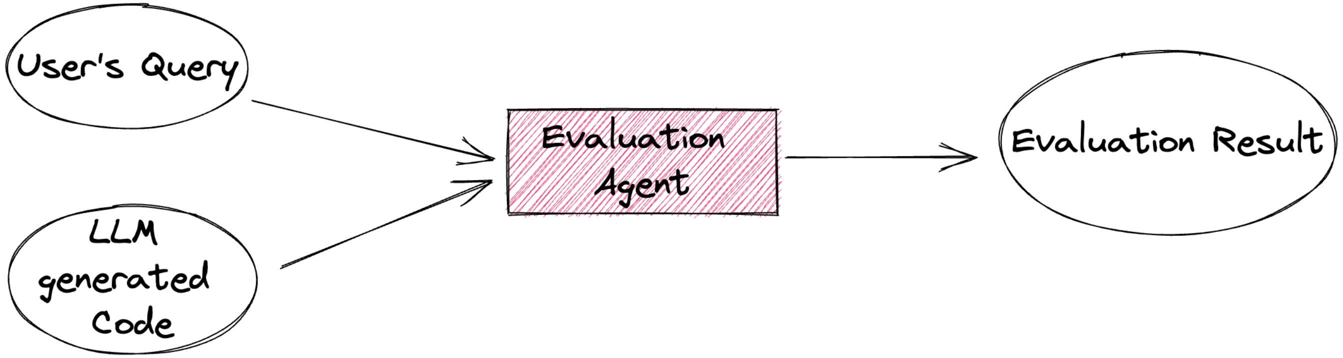


面向复杂软件工程任务的Coding Agent

- Reproduce Agent



- Evaluation Agent

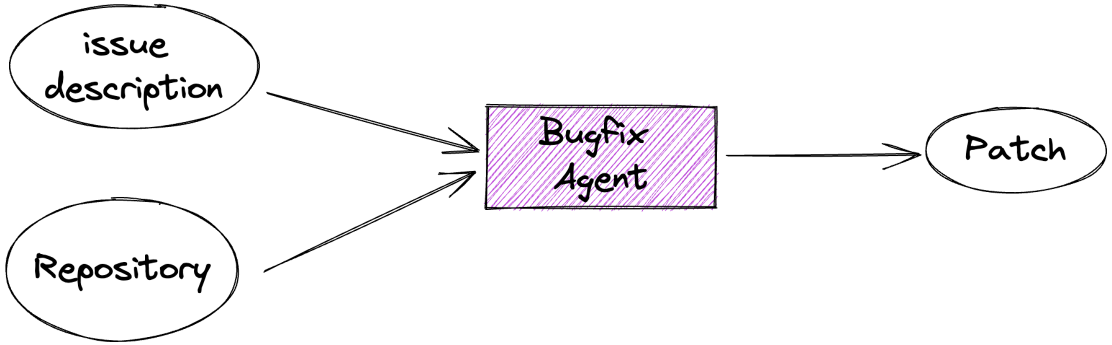


PART 02

Bugfix Agent

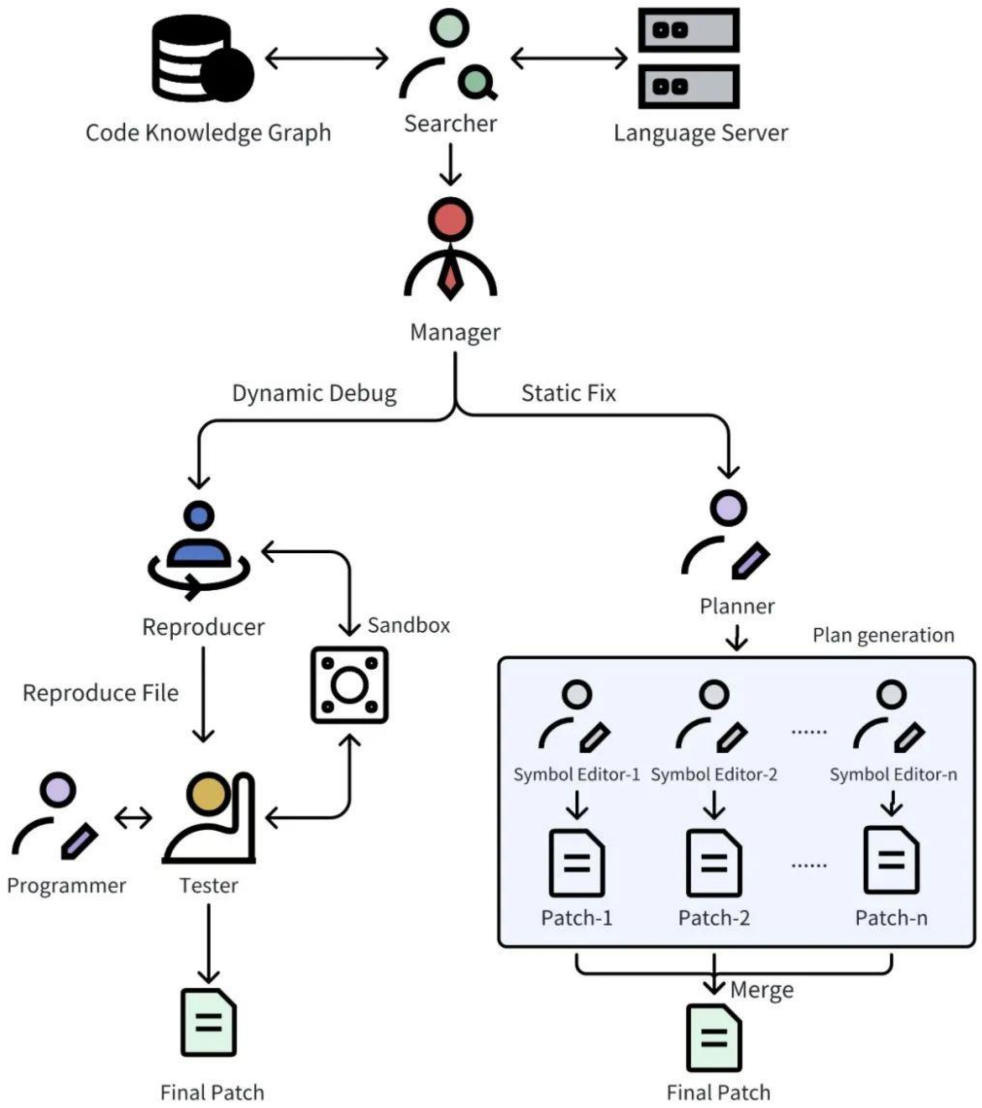
Bugfix Agent

- 自动求解 GitHub issue 的 Agent
- bug fix 是复杂软件工程任务的代表
  - 错误定位 和 缺陷修复
  - 代码仓粒度的代码理解
  - 代码生成：reproduce script 生成 & patch生成
  - 命令执行：reproduce script 执行



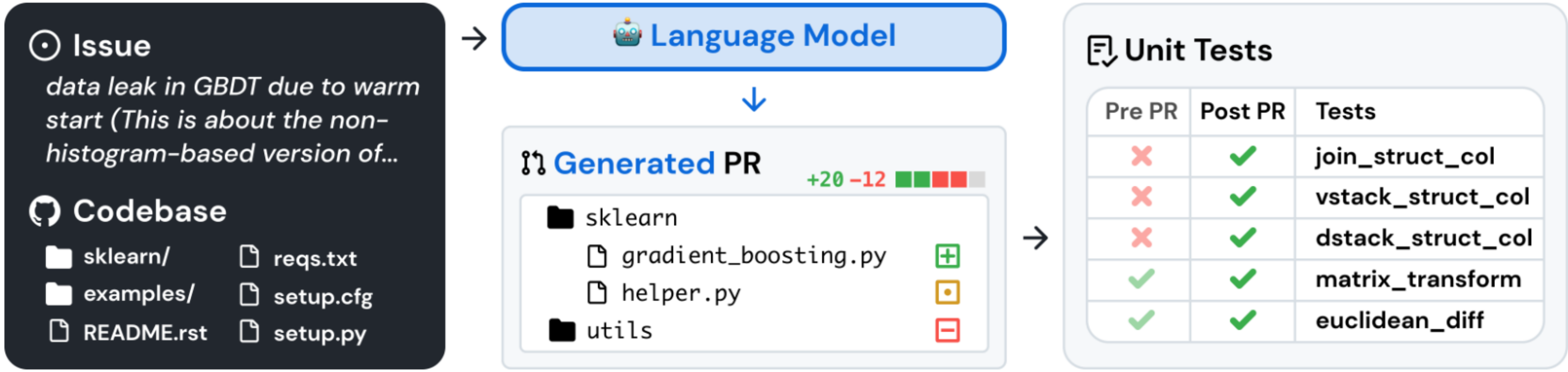
Bugfix Agent

- 代码检索
  - Code Knowledge Graph & LSP
- 动态调试
  - Reproduce Script 的生成和运行
  - 依赖代码仓运行环境
- 静态修复
  - 生成修复计划，Symbol 编辑
  - 不依赖代码仓运行环境



Evaluation on SWE-bench

- 12 个 Top Star Python 代码仓
- 收集 issue description / golden patch / test cases
- 检验 Agent 生成的 patch 是否能通过 test cases



Evaluation on SWE-bench

- Tech Giant
- AI Startup
- Academia

Lite | Verified | Full | Multimodal

☐ Open Weight Model ☐ Open Source System ☐ Checked (All Tags Selected)

TRAE @ 70.6%

Model	% Resolved	Org	Date	Logs	Trajs	Site
🆕 CORTEXA	68.20		2025-05-16	✓	✓	
🆕 Aime-coder v1 + Anthopic Claude 3.7 Sonnet	66.40		2025-05-14	✓	✓	-
🆕✅ OpenHands	65.80		2025-04-15	✓	✓	
Augment Agent v0	65.40		2025-03-16	✓	✓	
🆕 Amazon Q Developer Agent (v20250405-dev)	65.40		2025-04-05	✓	✓	
W&B Programmer O1 crosscheck5	64.60		2025-01-17	✓	✓	
🆕 PatchPilot-v1.1	64.60		2025-05-03	✓	✓	-
AgentScope	63.40		2025-02-06	✓	✓	
Tools + Claude 3.7 Sonnet (2025-02-24)	63.20		2025-02-24	✓	✓	
Blackbox AI Agent	62.80	-	2025-01-10	✓	✓	

How TRAE Achieve #1 on SWE-bench Verified

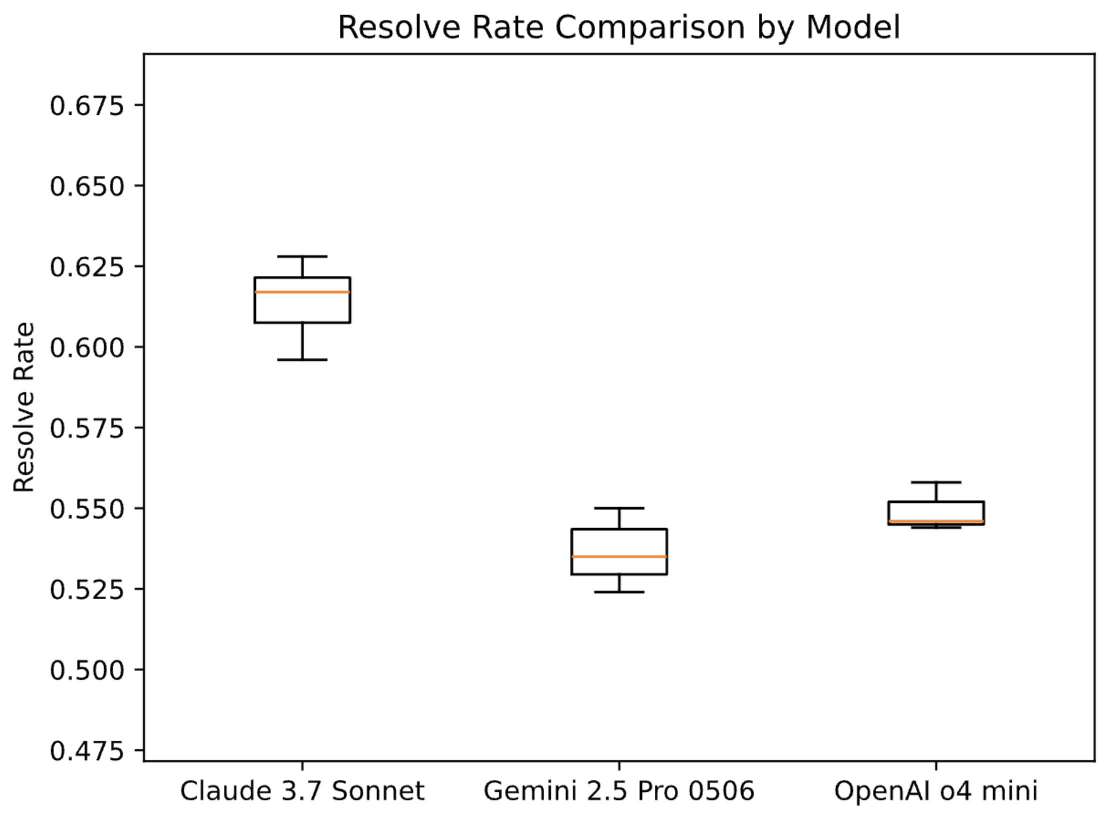


Figure 1: Resolve Rate Comparison by Model

How TRAE Achieve #1 on SWE-bench Verified

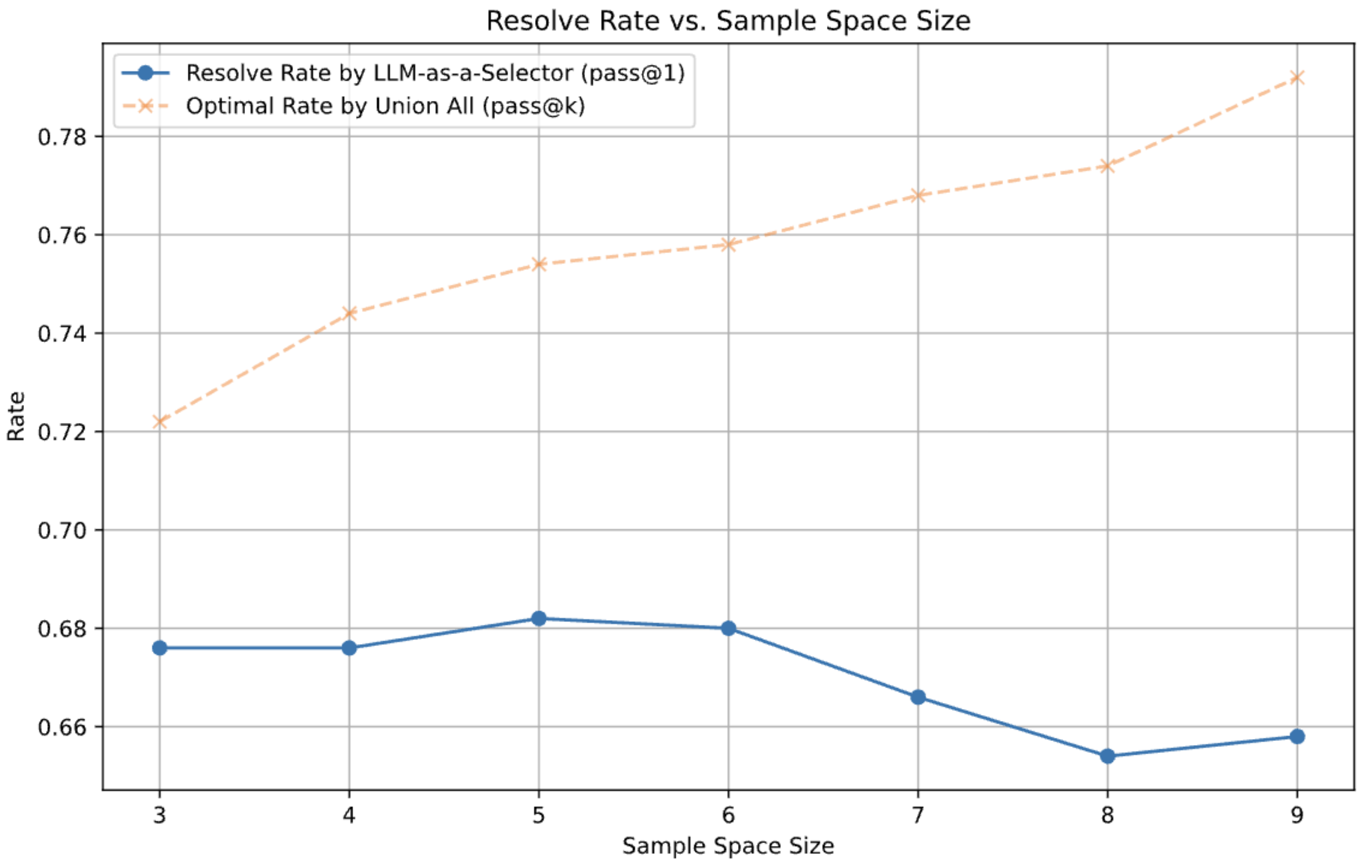


Figure 2: Resolve Rate vs. Sample Space Size

How TRAE Achieve #1 on SWE-bench Verified

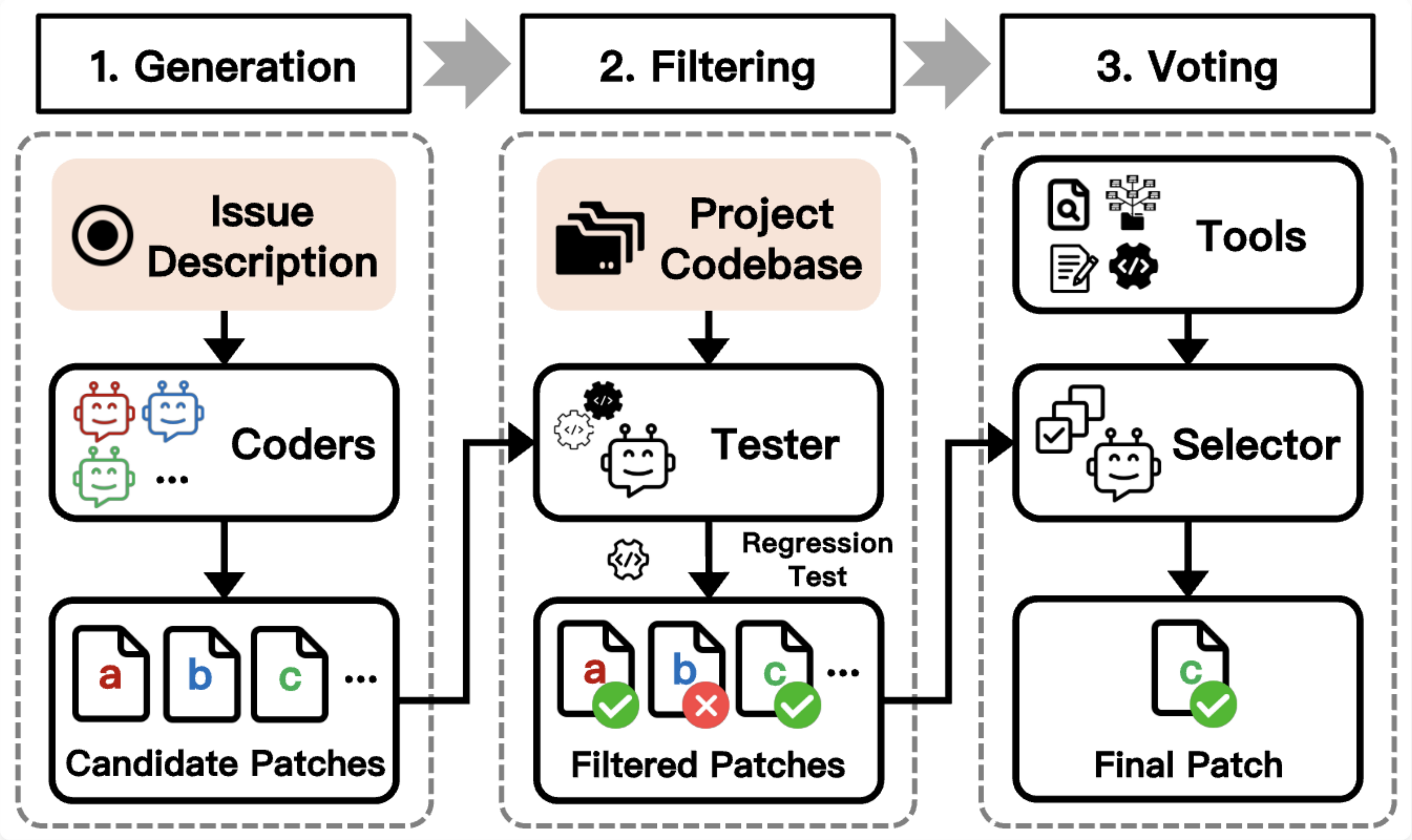


Figure 3: Selector Agent Overview

[1] https://se-research.bytedance.com/blogs/trae-on-swe-bench-verified-71/

Bugfix Agent – Lessons Learned

- 高质量 issue description 仍然求解失败
  - Stack Trace 较深
  - 无法正确识别关联修改
  - issue 描述中存在误导性求解方案
- Agent 系统和 Non-Agent 系统求解的差异
  - Agent 系统探索了过多 fault 位置，导致最终选择错误
  - Agent 缺乏采样，导致多样性较低
  - Non-Agent 缺乏多轮对话和工具调用能力，补丁语法错误

[1] An Empirical Study on LLM-based Agents for Automated Bug Fixing. Xiangxin Meng, Zexiong Ma, Pengfei Gao, Chao Peng, arXiv:2411.10213, 2024

Bugfix Agent 落地实践

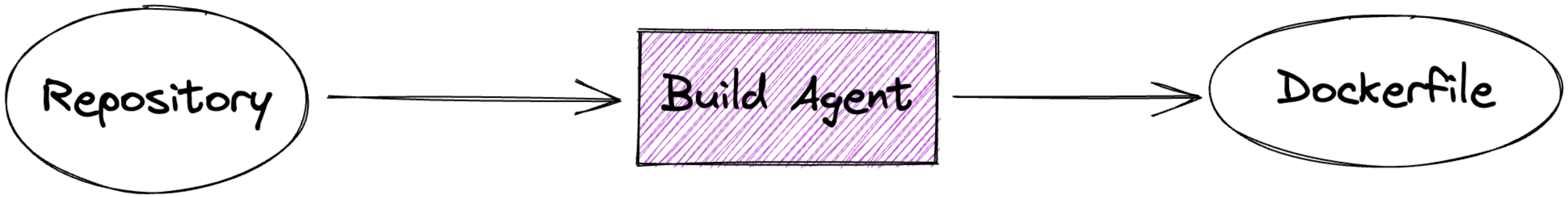
- 问题修复类型
  - 空指针异常 / 代码风格 / Clean Code 等
  - 线上 crash
- Patch 接受率达 70%
- 静态求解方案仍是主要手段

PART 03

Build Agent

Build Agent

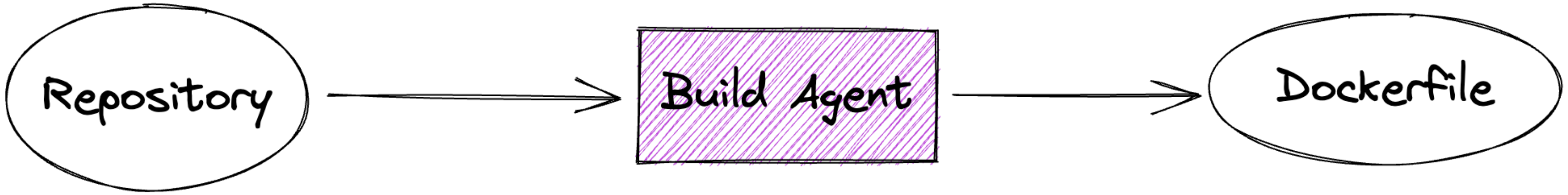
- 一个自动构建代码仓运行环境的 Agent
- 自动构建代码仓运行环境的重要性
  - Verifier & Feedback
  - RL Training



[1] https://github.com/bytedance/Repo2Run
[2] Hu, Ruida, Chao Peng, Xinchen Wang, and Cuiyun Gao. "An llm-based agent for reliable docker environment configuration." *arXiv preprint arXiv:2502.13681* (2025).

Build Agent

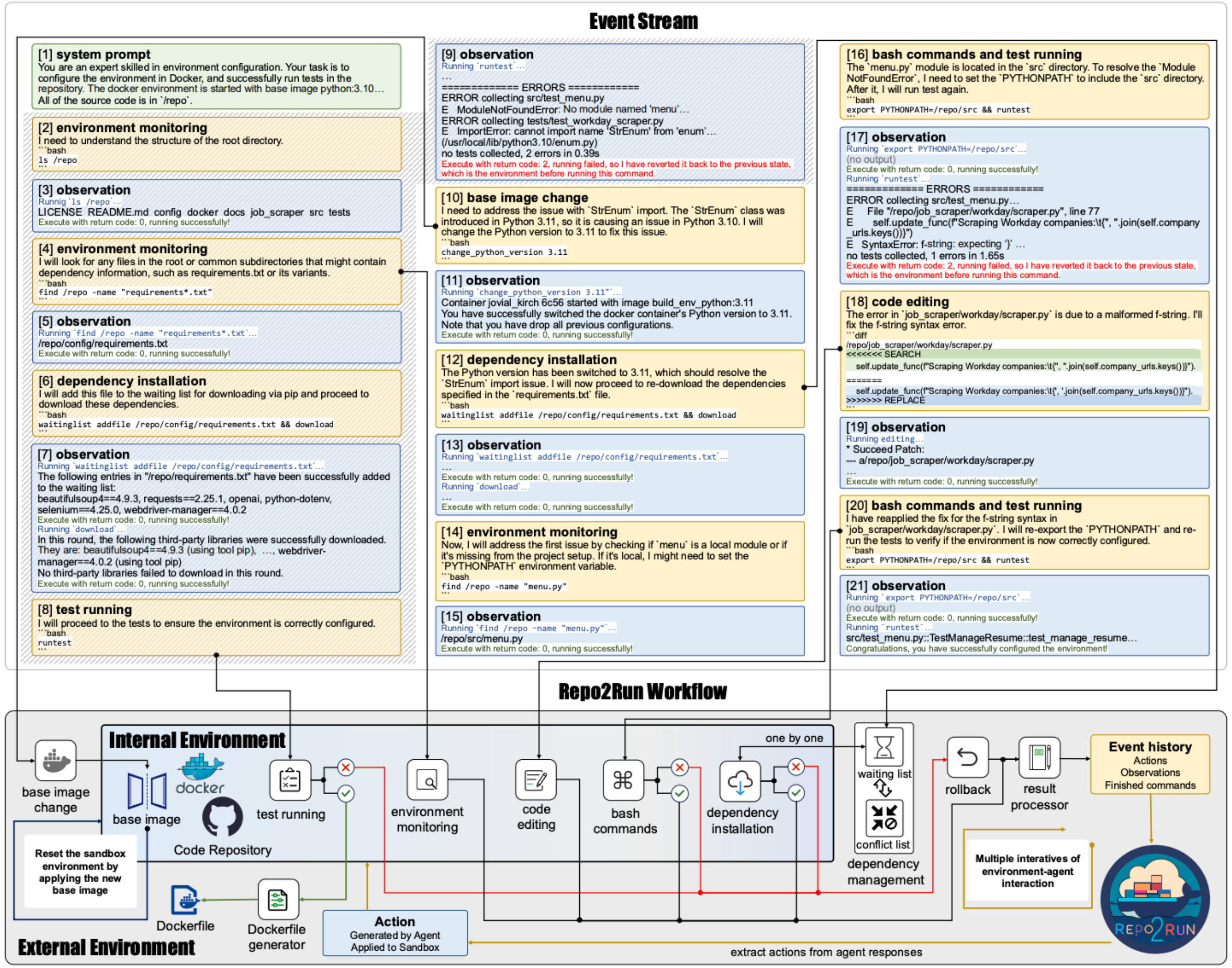
- 主要解决两个挑战
  - 挑战 1：在docker container中配置环境
    - 合适的 base image
    - 依赖安装
    - 依赖冲突
  - 挑战 2：配置过程转为 dockerfile



[1] https://github.com/bytedance/Repo2Run
[2] Hu, Ruida, Chao Peng, Xinchen Wang, and Cuiyun Gao. "An llm-based agent for reliable docker environment configuration." *arXiv preprint arXiv:2502.13681* (2025).

Build Agent

- 挑战 1: 环境配置
  - 基于反馈的 Rollback 操作
    - 测试运行失败
    - Bash 命令执行失败
    - 依赖安装失败
  - 合适的成功标志
    - 测试命令成功运行



[1] https://github.com/bytedance/Repo2Run
[2] Hu, Ruida, Chao Peng, Xinchen Wang, and Cuiyun Gao. "An llm-based agent for reliable docker environment configuration." *arXiv preprint arXiv:2502.13681* (2025).

Build Agent

- 挑战 2: Dockerfile 生成
  - 记录成功运行的命令
  - 记录状态改变操作

(default base image: python:3.10)

	internal command	return code	dir
1	ls /repo	0	/
2	cat /repo/README.md	0	/
3	pip install A	0	/
4	pip install -e /repo	1	/
5	change_python_version 3.11	0	/
6	cd /repo	0	/
7	cat setup.py	0	/repo
8	pip install -e .	0	/repo
9	python code_edit.py 0.patch	0	/repo
10	poetry install	1	/repo
11	pip install B>=1.0, <2.0	0	/
12	runtest	1	/
13	export TOKEN=123	0	/
14	runtest	0	/
15	pipdeptree –json-tree	0	/

(a) Commands running in the internal environment.

0. We begin with the default base image python:3.10.

```
FROM python:3.10
```

1-4. The commands “ls”, “cat” are safe commands, so they won't be added into Dockerfile. Instead, we save the command “pip install A”. The command “pip install –e .” failed, so rolling back.

```
FROM python:3.10
RUN pip install A
```

5. The base image is successfully changed, so all previous configurations are discarded, leaving only the new base image python:3.11.

```
FROM python:3.11
```

6-8. The commands “cd”, “cat” are safe commands, so they will not be added into the Dockerfile. Instead, we save the command “cd /repo && pip install –e .”

```
FROM python:3.11
RUN cd /repo && pip install –e .
```

9. Successfully edited the file. Apply a patch in the external environment, and save the "python code_edit.py" statement.

```
FROM python:3.11
RUN cd /repo && pip install –e .
RUN cd /repo && python /code_edit.py /patch/0.patch
```

10-12. The command “poetry install” failed, so rolling back. Save the successful execution of “pip install B>=1.0, <2.0”. Ignore all preset commands “runtest”.

```
FROM python:3.11
RUN cd /repo && pip install –e .
RUN cd /repo && python /code_edit.py /patch/0.patch
RUN pip install B>=1.0, <2.0
```

13. The export statement is converted into an ENV instruction for persistent retention.

```
FROM python:3.11
RUN cd /repo && pip install –e .
RUN cd /repo && python /code_edit.py /patch/0.patch
RUN pip install B>=1.0, <2.0
ENV TOKEN=123
```

14-15. After passing the tests, use “pipdeptree” to record the actual installed versions of the current Python third-party libraries, and replace non-specific constraints in the Dockerfile with these actual versions. Additionally, since code editing is required, copy the “code_edit.py” script and patch file(s) into the Docker container.

```
FROM python:3.11
COPY code_edit.py /code_edit.py
COPY patch /patch
RUN cd /repo && pip install –e .
RUN cd /repo && python /code_edit.py /patch/0.patch
RUN pip install B==1.5.1
ENV TOKEN=123
```

(b) The process of generating the Dockerfile by Dockerfile generator.

Figure 4: An example of the Dockerfile generator to transfer the commands into a runnable Dockerfile.

[1] https://github.com/bytedance/Repo2Run
[2] Hu, Ruida, Chao Peng, Xinchen Wang, and Cuiyun Gao. "An llm-based agent for reliable docker environment configuration." *arXiv preprint arXiv:2502.13681* (2025).

Build Agent - Experiment

- DGSR
  - Dockerfile Generation Success Rate
- ECSR
  - Environment Configuration Success Rate

Table 2: The results of ablation experiments.

Metric	DGSR	ECSR
w/o dual-environment	92.4% (388)	41.7% (175)
w/o Dockerfile generator	19.5% (82)	13.8% (58)
Repo2Run	100% (420)	86.0% (361)

Table 1: The results of different baselines.

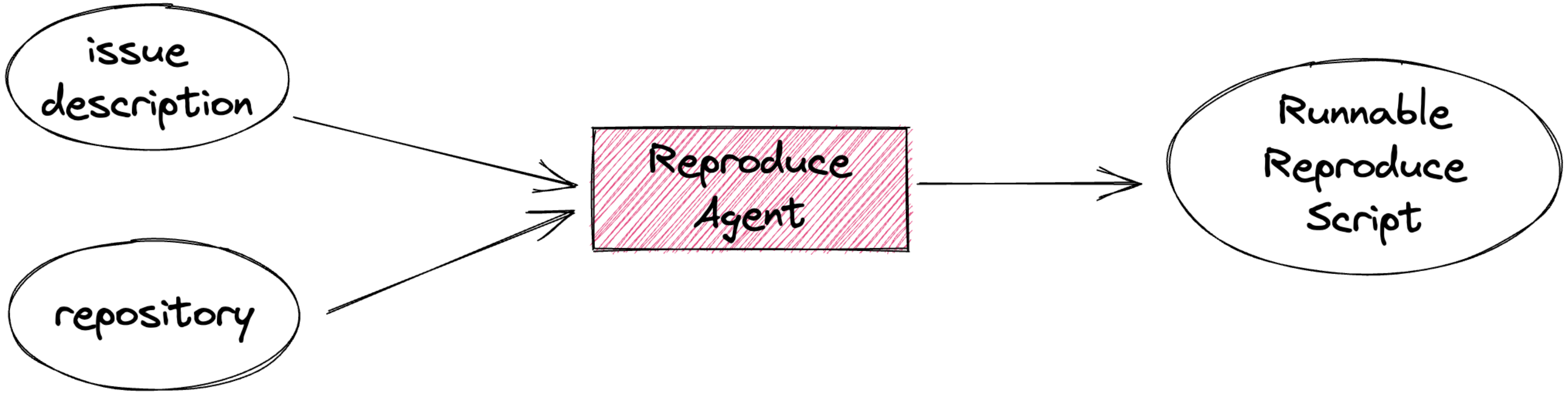
Metric	DGSR	ECSR
pipreqs	29.8% (125)	6.0% (25)
LLM generator	47.6% (200)	22.1% (93)
SWE-agent	26.9% (113)	9.0% (38)
Repo2Run	100% (420)	86.0% (361)

PART 04

Reproduce Agent

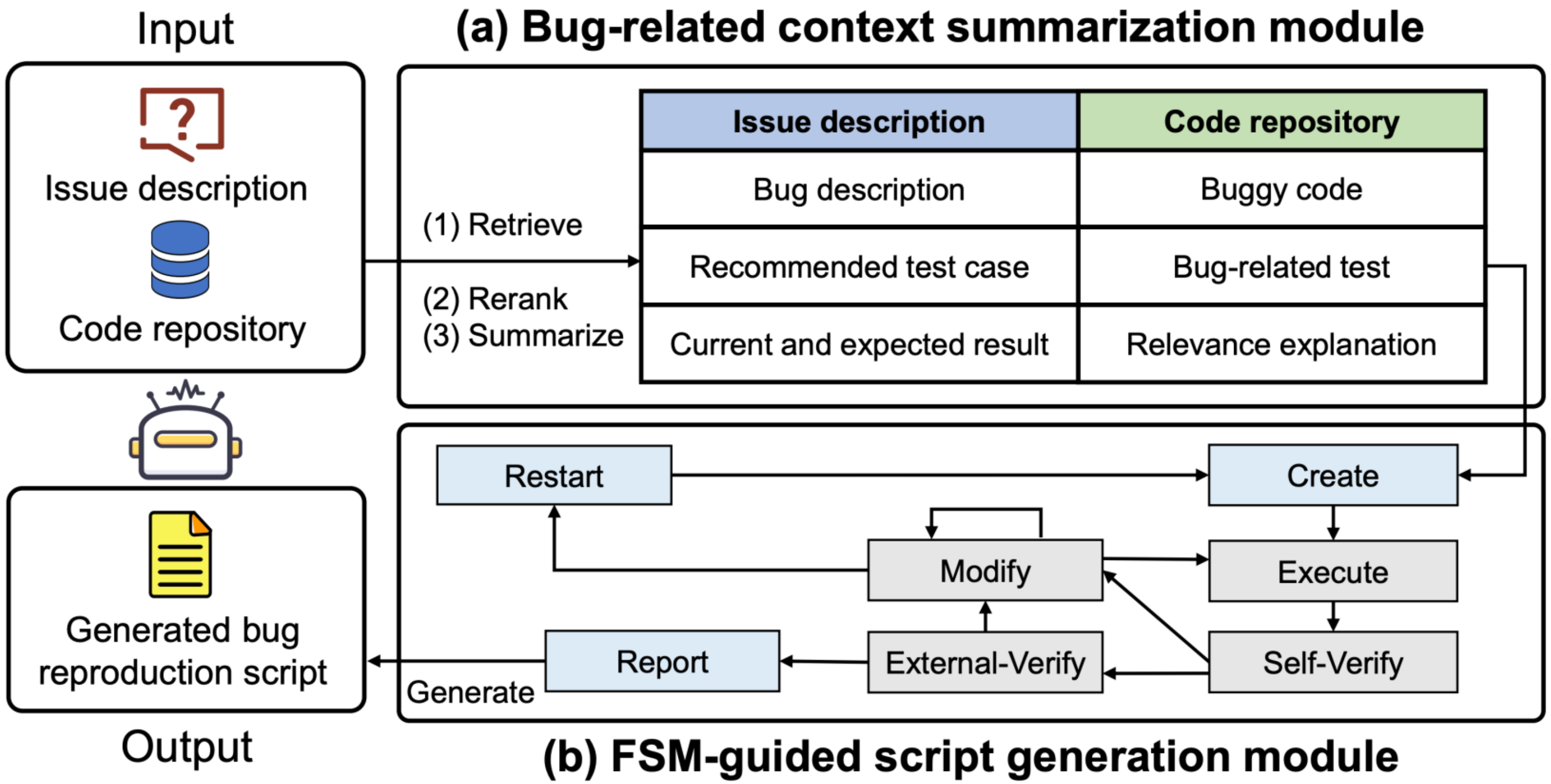
Reproduce Agent

- 动态求解的成功率远高于静态求解
- Reproduce 的成功率越高，修复效果越好
- test case generation



[1] Wang, Xinchen, Pengfei Gao, Xiangxin Meng, Chao Peng, Ruida Hu, Yun Lin, and Cuiyun Gao.
AEGIS: An Agent-based Framework for General Bug Reproduction from Issue Descriptions. *FSE 2025*

Reproduce Agent



[1] Wang, Xinchen, Pengfei Gao, Xiangxin Meng, Chao Peng, Ruida Hu, Yun Lin, and Cuiyun Gao. AEGIS: An Agent-based Framework for General Bug Reproduction from Issue Descriptions. *FSE 2025*

Reproduce Agent

- F -> P
    - golden patch 前失败
    - golden patch 后成功

Table 1: Comparison results between AEGIS, the LLM-based methods, and the agent-based methods.

	Method	$F \to \times$	$P \to P$	$F \to P$
LLM-based	ZEROSHOT	38.8	3.6	5.8
	ZEROSHOTPLUS	55.4	7.2	10.1
	LIBRO	60.1	7.2	15.2
Agent-based	AUTOCODEROVER	43.8	7.6	9.1
	AIDER	57.6	8.7	17.0
	SWE-AGENT	48.2	9.8	16.7
	AEGIS	90.0	9.0	**36.0**

[1] Wang, Xinchen, Pengfei Gao, Xiangxin Meng, Chao Peng, Ruida Hu, Yun Lin, and Cuiyun Gao. AEGIS: An Agent-based Framework for General Bug Reproduction from Issue Descriptions. *FSE 2025*

Reproduce Agent

Lite Verified

Script Mode ‡ Unit Test Mode

Model	Org.	$\mathcal{S}$ (↓)	$\Delta\mathcal{C}$	Date	Traj.
🥇 AEGIS‡		47.8%	26.0%	2025/2/17	🔗
🥇 🆕 Amazon Q Developer Agent v20250405-dev	aws	37.7%	52.7%	2025/4/10	🔗
🥈 OpenHands CI setup#	🙌	28.3%	52.4%	2025/2/18	🔗
🥉 OpenHands vanilla	🙌	22.8%	43.6%	2025/2/18	🔗
SWE-Agent+		18.5%	27.6%	2024/5/22	🔗
SWE-Agent Mistral Large 2		16.3%	23.0%	2024/5/22	🔗
SWE-Agent GPT-4		15.9%	26.5%	2024/5/22	🔗

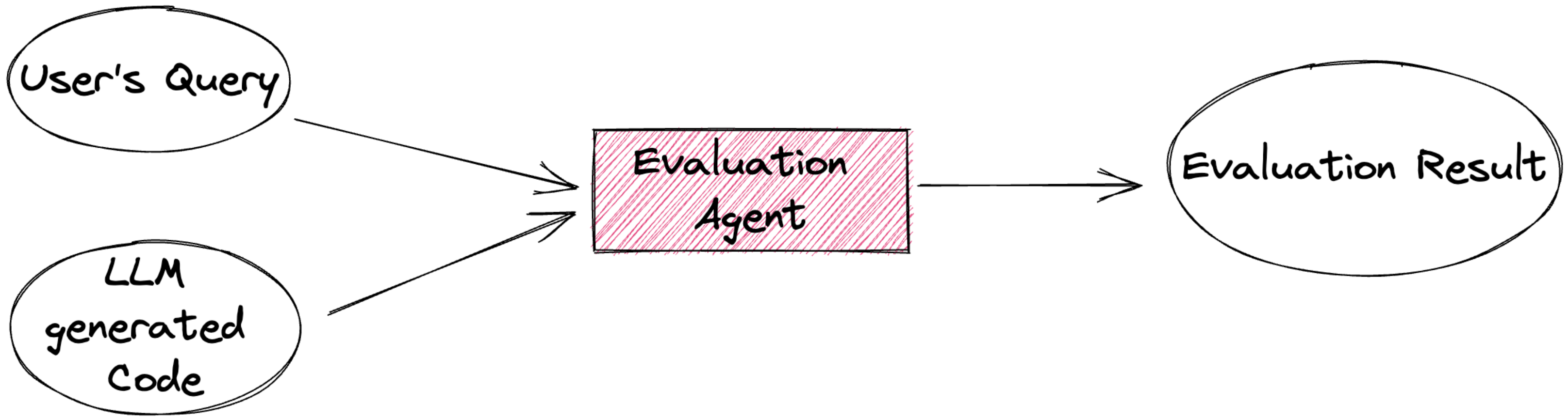
[1] Wang, Xinchen, Pengfei Gao, Xiangxin Meng, Chao Peng, Ruida Hu, Yun Lin, and Cuiyun Gao. AEGIS: An Agent-based Framework for General Bug Reproduction from Issue Descriptions. *FSE 2025*

PART 05

Evaluation Agent

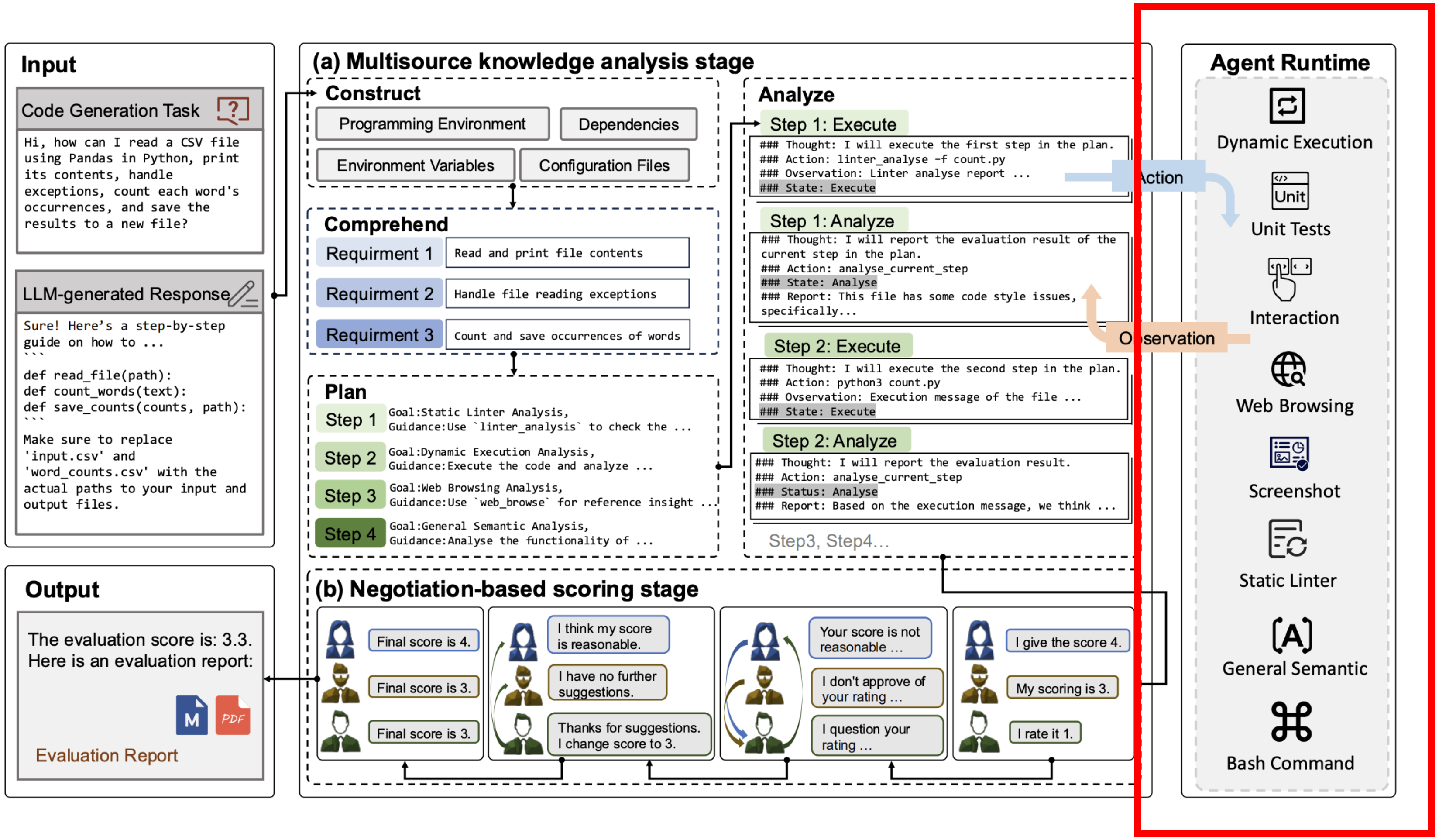
Evaluation Agent

- 评估 LLM 代码生成的质量
  - 人工评分
  - LLM-as-a-Judge
- LLM-as-a-Judge
  - 缺乏多源领域知识
  - 复杂代码理解不足



[1] Wang, Xinchen, Pengfei Gao, Chao Peng, Ruida Hu, and Cuiyun Gao.
CodeVisionary: An Agent-based Framework for Evaluating Large Language Models in Code Generation." *arXiv preprint arXiv:2504.13472* (2025).

Evaluation Agent



[1] Wang, Xinchen, Pengfei Gao, Chao Peng, Ruida Hu, and Cuiyun Gao.
CodeVisionary: An Agent-based Framework for Evaluating Large Language Models in Code Generation." *arXiv preprint arXiv:2504.13472* (2025).

Evaluation Agent

- Benchmark 构建

Table 1: Statistics of the constructed benchmark.

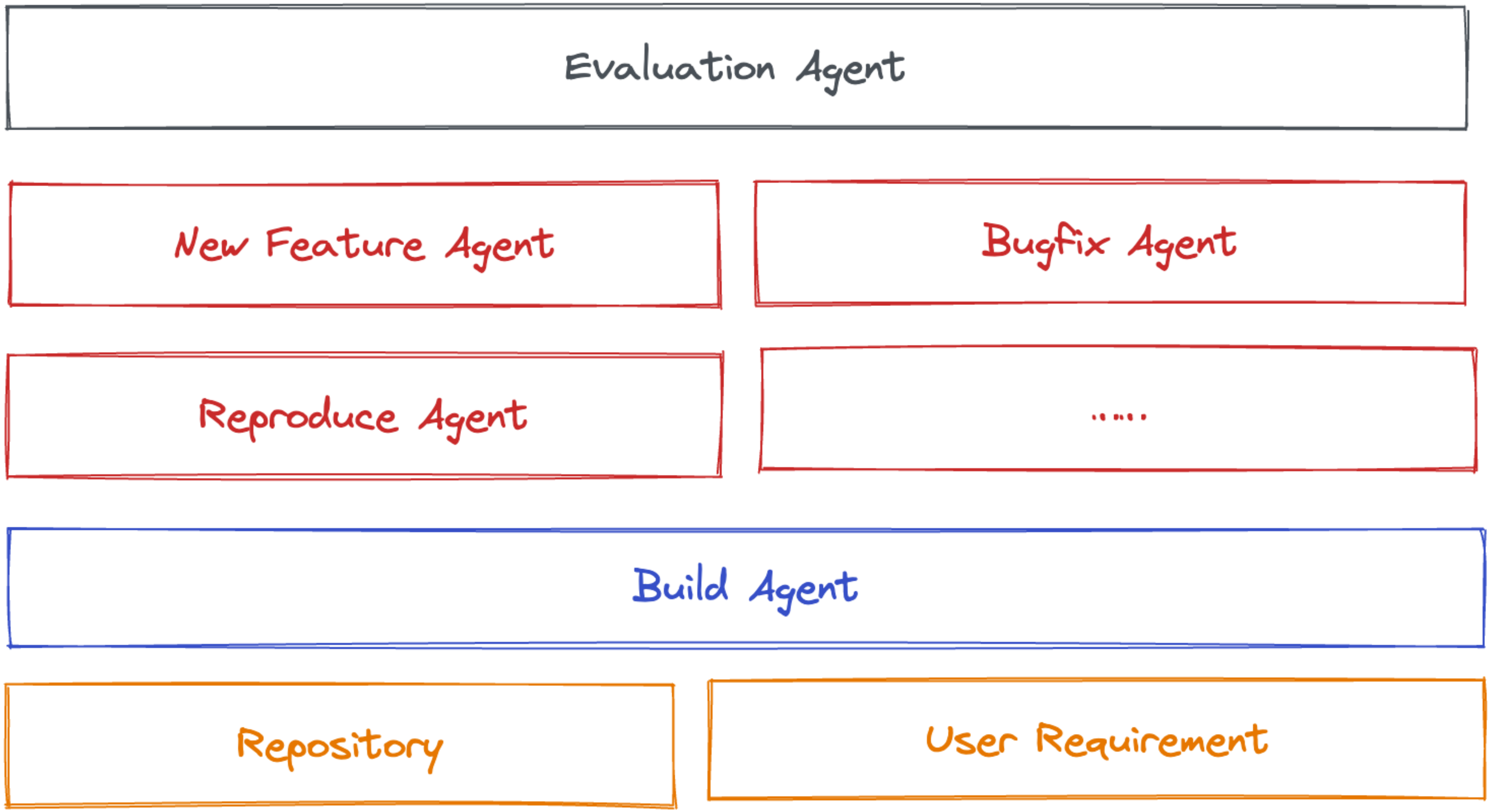
Statistics	Number
Samples	363
Coding Scenarios	37
Programming Languages	23
Average String Length	
Code Generation Task	1,906
LLM-generated Response	2,650

- 在多个相关性指标上领先 baseline

Table 2: Comparison results between CodeVisionary and baseline methods.

Models	GPT-3.5-turbo			Claude-3.5-Sonnet			GPT-4o		
Metrics	r_p	r_s	τ	r_p	r_s	τ	r_p	r_s	τ
VANILLA	0.083	0.129	0.117	0.033	0.098	0.094	0.010	0.047	0.045
ICE-SCORE	0.100	0.177	0.157	0.029	0.096	0.092	-0.011	0.054	0.051
CODEJUDGE	0.130	0.130	0.120	0.097	0.074	0.072	0.025	0.078	0.076
CodeVisionary	**0.329**	**0.342**	**0.288**	**0.312**	**0.245**	**0.222**	**0.218**	**0.156**	**0.140**

总结



Evaluation Agent
New Feature Agent
Bugfix Agent
Reproduce Agent
......
Build Agent
Repository
User Requirement

未来展望

TRAE
IDE
插件
文档
社区
活动 NEW
登录
下载 IDE
智能无限，协作无间
Trae，致力于成为真正的 AI 工程师（The Real AI Engineer）。Trae 旗下的 AI IDE 产品，以智能生产力为核心，无缝融入你的开发流程，与你默契配合，更高质量、高效率完成每一个任务。
立即获取 Trae IDE
查看所有下载选项
myproject
搜索
文件
react-todolist
node_modules
public
src
assets
components
TodoList.jsx
styles
TodoList.css
TodoList.jsx
App.css
App.jsx
TodoList.css
Trae
与 @Builder With MCP 协作

AiDD 6th 2025

参与调研您将优先获得

AiDD定制版
《AI+软件研发精选案例》

专属学习顾问
1对1需求对接

AiDD会后小调研

AiDD峰会致力于协助企业利用AI技术深化计算机对现实世界的理解，推动研发进入智能化和数字化的新时代。作为峰会的重要共建者，您的真知灼见对我们至关重要。衷心感谢您的参与支持！

扫码参与调研

2025 AI+研发数字峰会

拥抱AI 重塑研发

科技生态圈峰会 + 深度研习

——1000 + 技术团队的共同选择

KEYLINK ing

K+峰会

- 敦煌站｜K+ 思考周®研习社｜时间：2025.08.29-30
- 上海站｜K+ 金融专场｜时间：2025.09.26-27
- 香港站｜K+思考周®研习社｜时间：2025.11.17-18

K+峰会详情

AiDD峰会

- 上海站｜AI+研发数字峰会｜时间：2025.05.23-24
- 北京站｜AI+研发数字峰会｜时间：2025.08.08-09
- 深圳站｜AI+研发数字峰会｜时间：2025.11.14-15

AiDD峰会详情

感谢聆听！

扫码领取会议PPT资料