OWASP
Open Worldwide Application Security Project

探索重塑 AI与大模型技术引领的安全革新实践

2024 OWASP中国安全技术论坛

大模型驱动的应用安全治理智能化

一家安全公司如何直面AI新浪潮

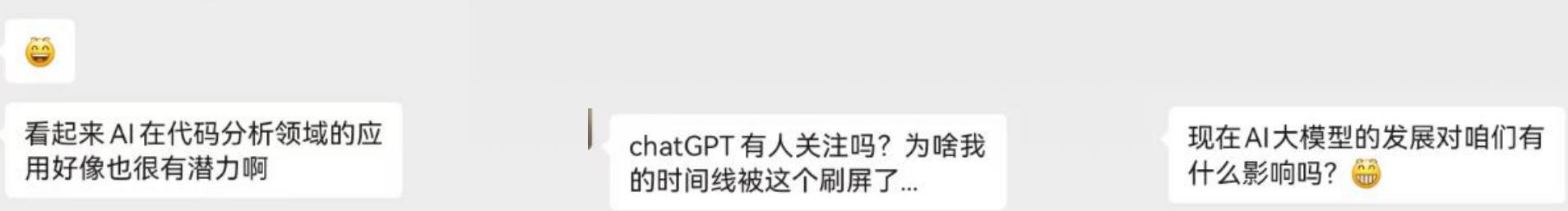
蜚语科技 束骏亮

01.

你们公司是不是要倒闭了?

大模型惊人的泛化能力

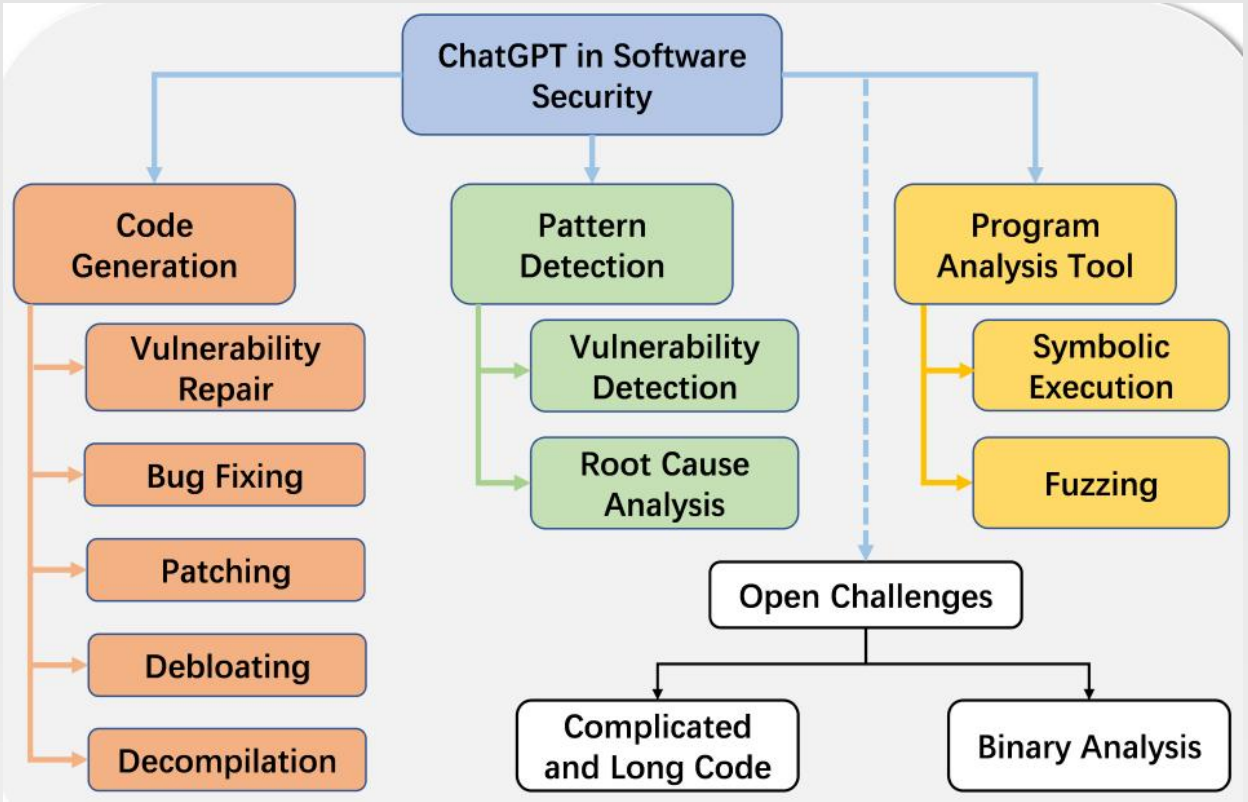
- 2022年11月30日 OpenAI发布以GPT3.5为基座模型的ChatGPT
- 在文本理解、对话、代码阅读等领域展现出惊人的泛化能力
- 随之而来的便是。。。



大模型的能力边界

- 早期判断：成本太高、精确度太差、速度太慢

- 大模型本身也在持续迭代，因此我们决定系统性的研究一下大模型在代码安全领域的能力边界



大模型的能力边界

请帮我找出以下代码中的bug：

```
int testARR30()
{
    int x = 9936;
    int j;
    int t[9936] = {0};
    int i;
    for(i = 0; i<=x; i++)
    {
        j = t[i];
    }
    return j;
}
```

这段代码存在数组越界的漏洞。

具体来说，数组 `t` 的大小为 9936，但是在循环中使用 `i <= x` 的条件，当 `i` 的值等于 9936 时，会访问 `t` 数组的第 9937 个元素，从而导致数组越界。这可能会导致程序崩溃或者产生不可预期的结果。

为了避免这个问题，应该将循环的条件改为 `i < x`。

请帮我找出以下代码中的bug：

```
int testARR30()
{
    int x = 9935;
    int j;
    int t[9936] = {0};
    int i;
    for(i = 0; i<=x; i++)
    {
        j = t[i];
    }
    return j;
}
```

这段代码存在一个越界访问的漏洞。

具体来说，数组 `t` 的大小为 9936，但是在循环中使用 `i <= x` 的条件，当 `i` 的值等于 9935 时，会访问 `t` 数组的第 9936 个元素。虽然数组 `t` 的大小足够大，但是当访问超出数组范围时，可能会访问到未定义的内存位置，从而导致程序出现未定义行为，例如崩溃或者产生奇怪的结果。

为了避免这个问题，应该将循环的条件改为 `i < x`，以确保不会访问数组范围之外的内存位置。

大模型的能力边界


```
// Simple ? condition that assigns constant to bar on true condition
int num = 106;

bar = (7 * 18) + num > 200 ? "This_should_always_happen" : param;

String fileName = null;
java.io.FileInputStream fis = null;

try {
   fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + bar;
   fis = new java.io.FileInputStream(new java.io.File(fileName));
```

这段代码存在一个潜在的安全漏洞，即路径遍历攻击（也称为目录遍历攻击）。攻击者可能会利用这个漏洞来访问、读取或修改您系统中不应该暴露的敏感文件。

问题出在这一行代码上：

java Copy code

```
fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + bar;
```

变量`bar`的值可能是用户可控的，因为它的值可能受到`num`和`param`变量的影响。攻击者可以构造一个特殊的`param`值，以使`bar`包含".."等字符序列，从而导致应用程序访问到意料之外的目录和文件。

大模型的能力边界

结果引用自：

Wu F, Zhang Q, Bajaj A P, et al. Exploring the Limits of ChatGPT in Software Security Applications[J]. arXiv preprint arXiv:2312.05275, 2023.

TABLE 1: Vulnerability detection on synthetic code.

ID	#Case	GPT-3.5				GPT-4			
		TP	FP	Precision	Recall	TP	FP	Precision	Recall
CWE-22	10	5	1	83.33%	100%	5	1	83.33%	100%
CWE-78	10	5	3	62.50%	100%	5	1	83.33%	100%
CWE-79	10	5	4	55.56%	100%	5	1	83.33%	100%
CWE-89	10	5	3	62.50%	100%	5	0	100%	100%
CWE-119	10	4	3	57.14%	80%	5	1	83.33%	100%
CWE-125	10	5	2	71.43%	100%	5	2	71.43%	100%
CWE-190	10	3	2	60%	60%	4	1	80%	80%
CWE-416	10	5	2	71.43%	100%	5	0	100%	100%
CWE-476	10	4	1	80%	80%	5	0	100%	100%
CWE-787	10	3	0	100%	60%	5	0	100%	100%
Total	100	44	21	67.69%	88%	49	7	87.50%	98%

TABLE 2: Vulnerability detection on CVEs.

Language	#Case	GPT-3.5					GPT-4				
		TP	FP	Fail	Precision	Recall	TP	FP	Fail	Precision	Recall
C	34	3	1	6	75%	21.43%	9	3	0	75%	52.94%
Cpp	6	0	2	2	0%	0%	1	2	0	33.33%	33.33%
Python	10	0	2	0	0%	0%	1	1	0	50%	20%
Go	8	0	0	0	0%	0%	3	0	0	100%	75%
JavaScript	8	2	2	0	50%	50%	2	0	0	100%	50%
PHP	2	0	0	0	0%	0%	1	0	0	100%	100%
Total	68	5	7	8	41.67%	17.24%	17	6	0	73.91%	50%

大模型的能力边界

准确性

- 无法处理代码中的复杂逻辑/运算
- 处理超长上下文，注意力无法集中

效率

- 受限于上下文，分批处理时效率低
- 分析时间与结果长度成正比，无法满足生产环境要求

可维护性

- 结果存在不可预测性，偶发幻觉
- 无法进行有效的定向迭代优化（代价大）

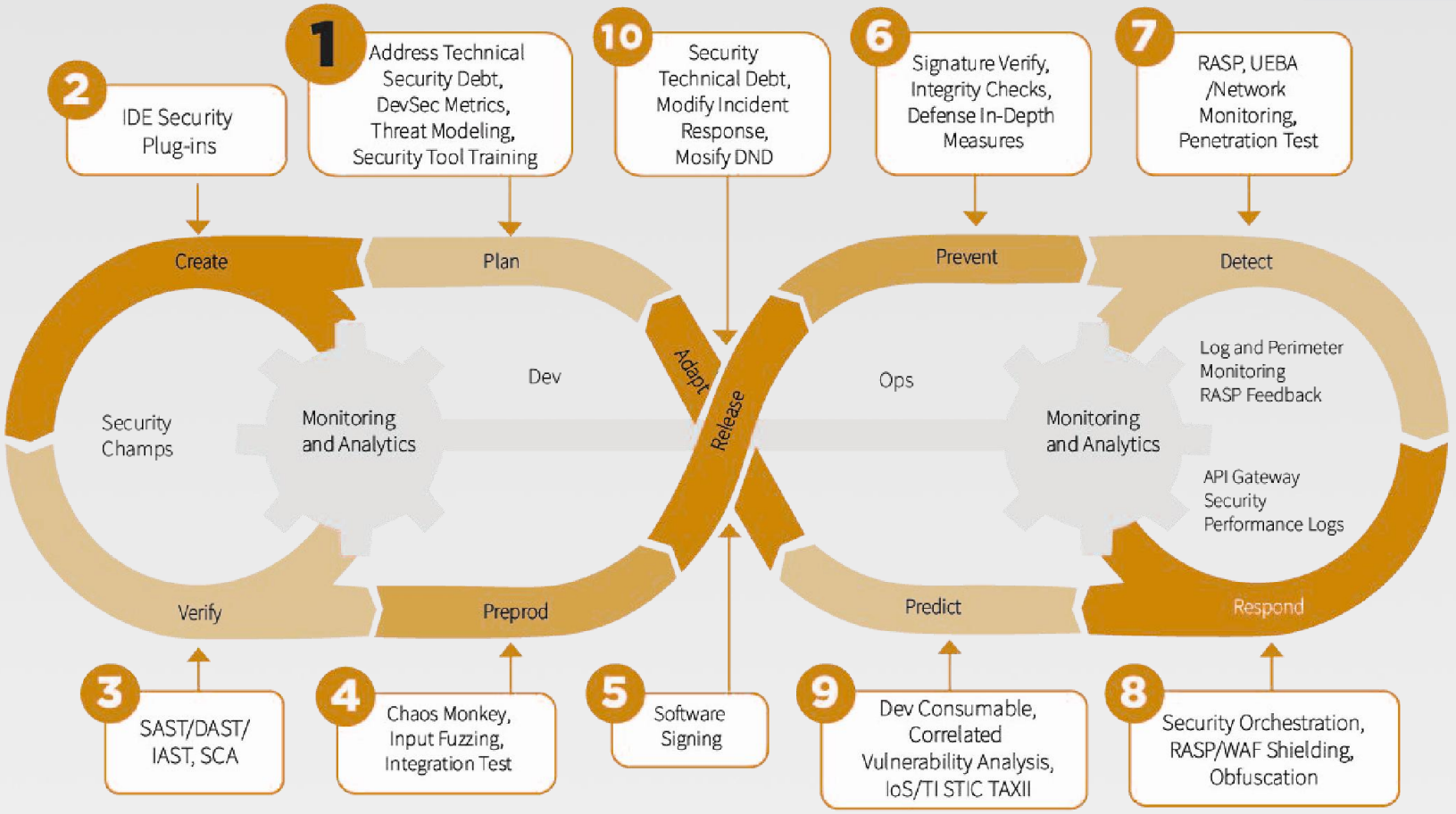
02.

我们能不能蹭到大模型的热度?

从代码安全扩展到应用安全

大模型的泛化能力，降低了不同细分领域之间的技术壁垒

应用安全长久以来的痛点



应用安全长久以来的痛点

01 既要懂安全又要懂研发

02 工具繁多，场景复杂

03 费时费力，不出成绩

Merlin

基于生成式人工智能的
应用安全运营平台

- 对任意的传统应用安全工具进行AI化改造
- 自动实现工具的使用与结果的分析

OWASP
Open Worldwide Application Security Project

探索重塑 AI与大模型技术引领的安全革新实践

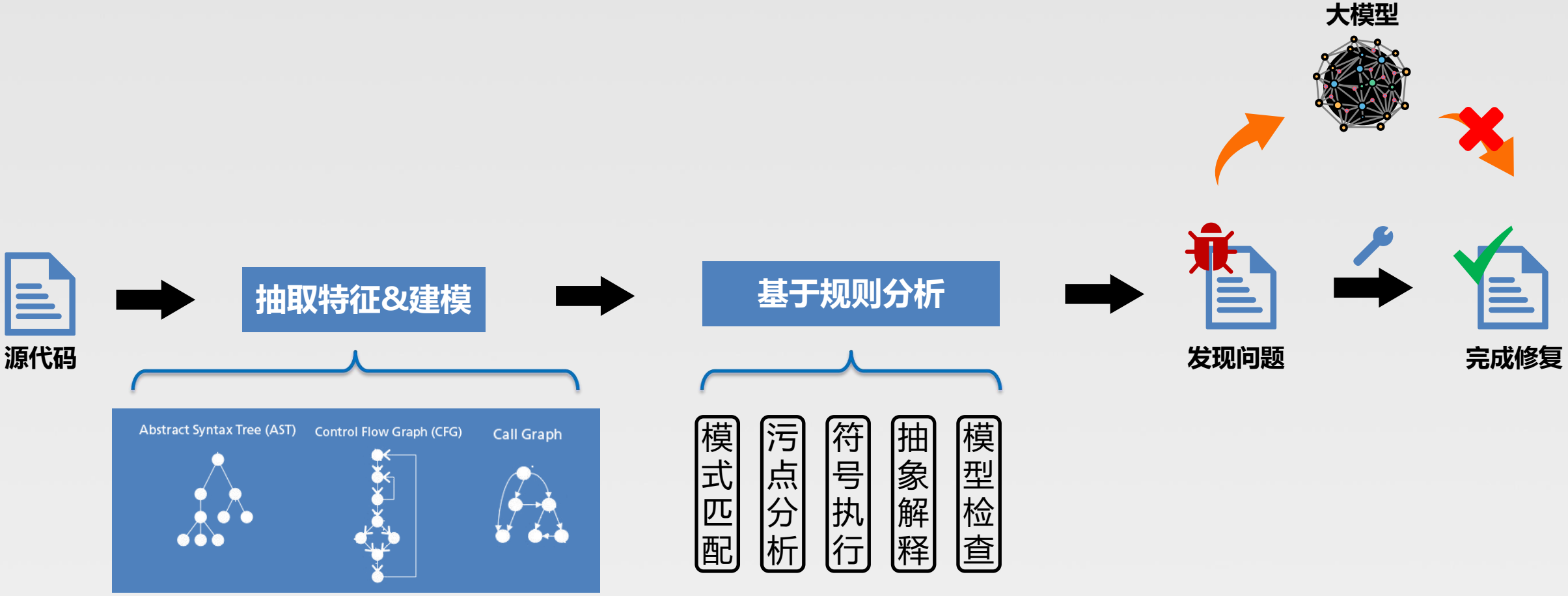
2024 OWASP中国安全技术论坛

03.

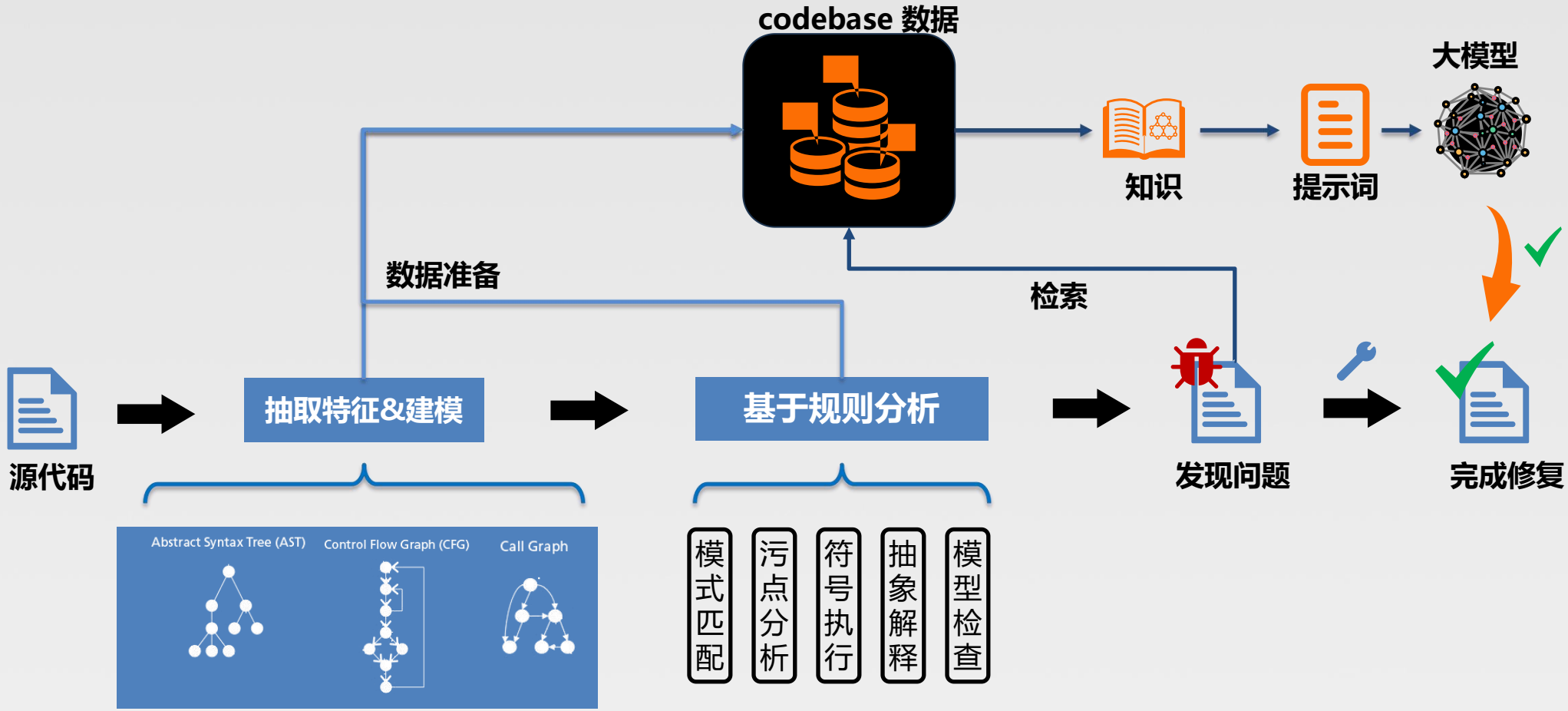
如何突破大模型的能力边界?

如何拯救大模型的短板

如何拯救大模型的短板


大模型
源代码
抽取特征&建模
Abstract Syntax Tree (AST)
Control Flow Graph (CFG)
Call Graph
基于规则分析
模式匹配
污点分析
符号执行
抽象解释
模型检查
发现问题
完成修复

如何拯救大模型的短板

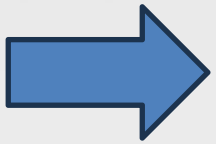

codebase 数据
大模型
知识
提示词
数据准备
检索
源代码
抽取特征&建模
基于规则分析
发现问题
完成修复
Abstract Syntax Tree (AST)
Control Flow Graph (CFG)
Call Graph
模式匹配
污点分析
符号执行
抽象解释
模型检查

04.

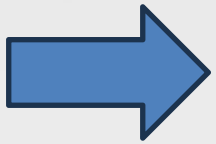
预训练/微调/RAG还是其他?

给大模型导入行业know how

垂直领域，尤其是小众垂直领域



大模型的知识边界



向大模型导入行业/业务知识

给大模型导入行业know how

预训练

- 成本太高，难以实现
- 基座模型迭代快，能力趋于统一

微调/后训练

- 有一定的成本（数据和时间）
- 在基座模型竞争趋于统一的情况下，意义不大
- 可能有副作用
- O1出现以后，基于强化学习的后训练可能能够更好的利用垂类数据

RAG

- 成本低
- 适用于向大模型导入行业背景知识

WorkFlow

- 能够加入验证环节，让大模型的行为更可控
- 适用于向大模型导入任务相关的know how

给大模型导入行业know how

预训练

- 成本太高，难以实现
- 基座模型迭代快，能力趋于统一

微调/后训练

- 有一定的成本（数据和时间）
- 在基座模型竞争趋于统一的情况下，意义不大
- 可能有副作用
- O1出现以后，基于强化学习的后训练可能能够更好的利用垂类数据

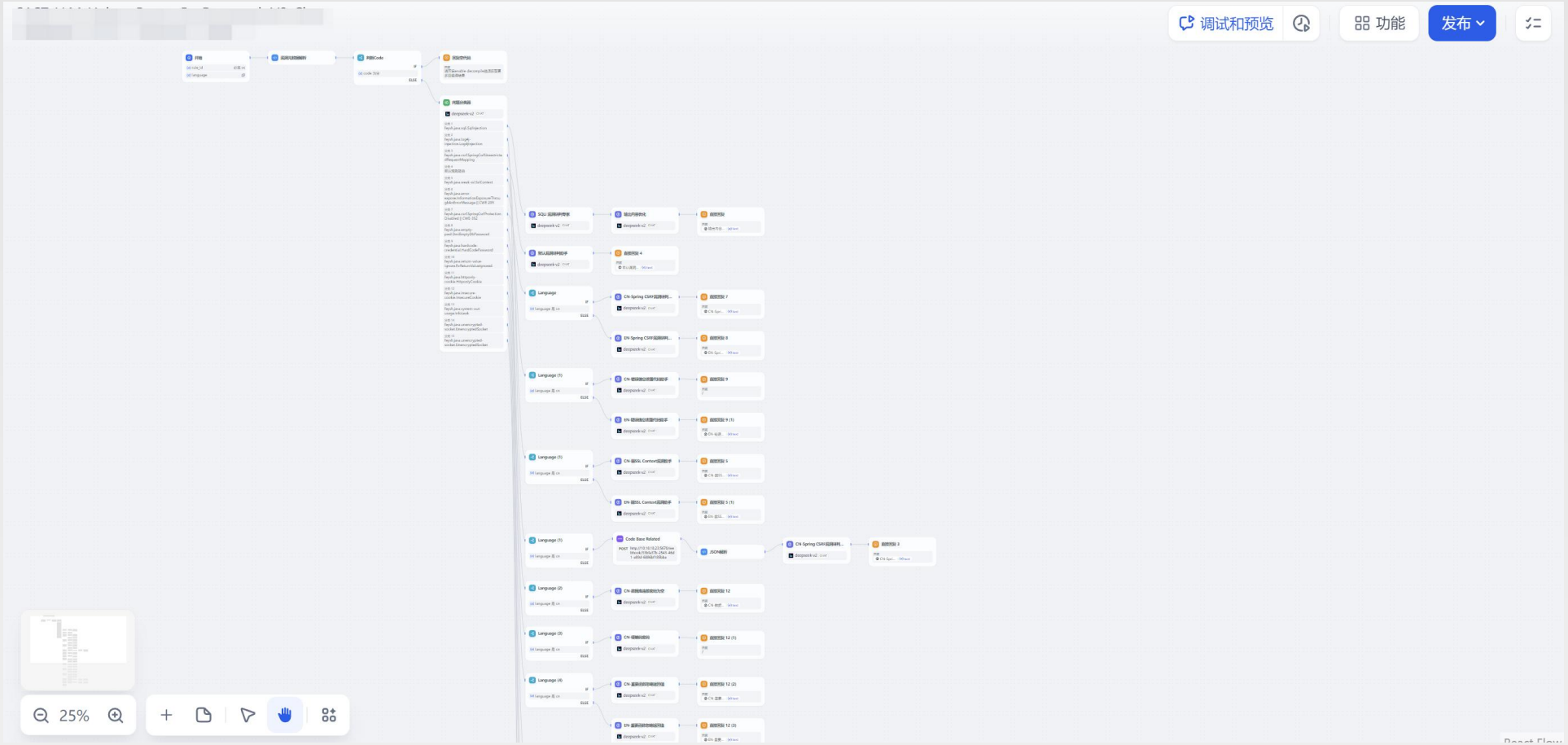
RAG

- 成本低
- 适用于向大模型导入行业背景知识

WorkFlow

- 能够加入验证环节，让大模型的行为更可控
- 适用于向大模型导入任务相关的know how

给大模型导入行业know how



05.

Copilot还是Autopilot?

挂机是人类永恒的追求

Agentic Reasoning Design Patterns

1. Reflection
   - Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
   - Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)
2. Tool use
   - Gorilla: Large Language Model Connected with Massive APIs, Patil et al. (2023)
   - MM-REACT: Prompting ChatGPT for Multimodal Reasoning and Action, Yang et al. (2023)
3. Planning
   - Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Wei et al., (2022)
   - HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face, Shen et al. (2023)
4. Multi-agent collaboration
   - Communicative Agents for Software Development, Qian et al., (2023)
   - AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation, Wu et al. (2023)

Andrew Ng

挂机是人类永恒的追求

大模型技术的最终目标是从交付工具，变成交付结果

06.

思维链还是累积推理?

从简单粗暴到复杂完备

CoT

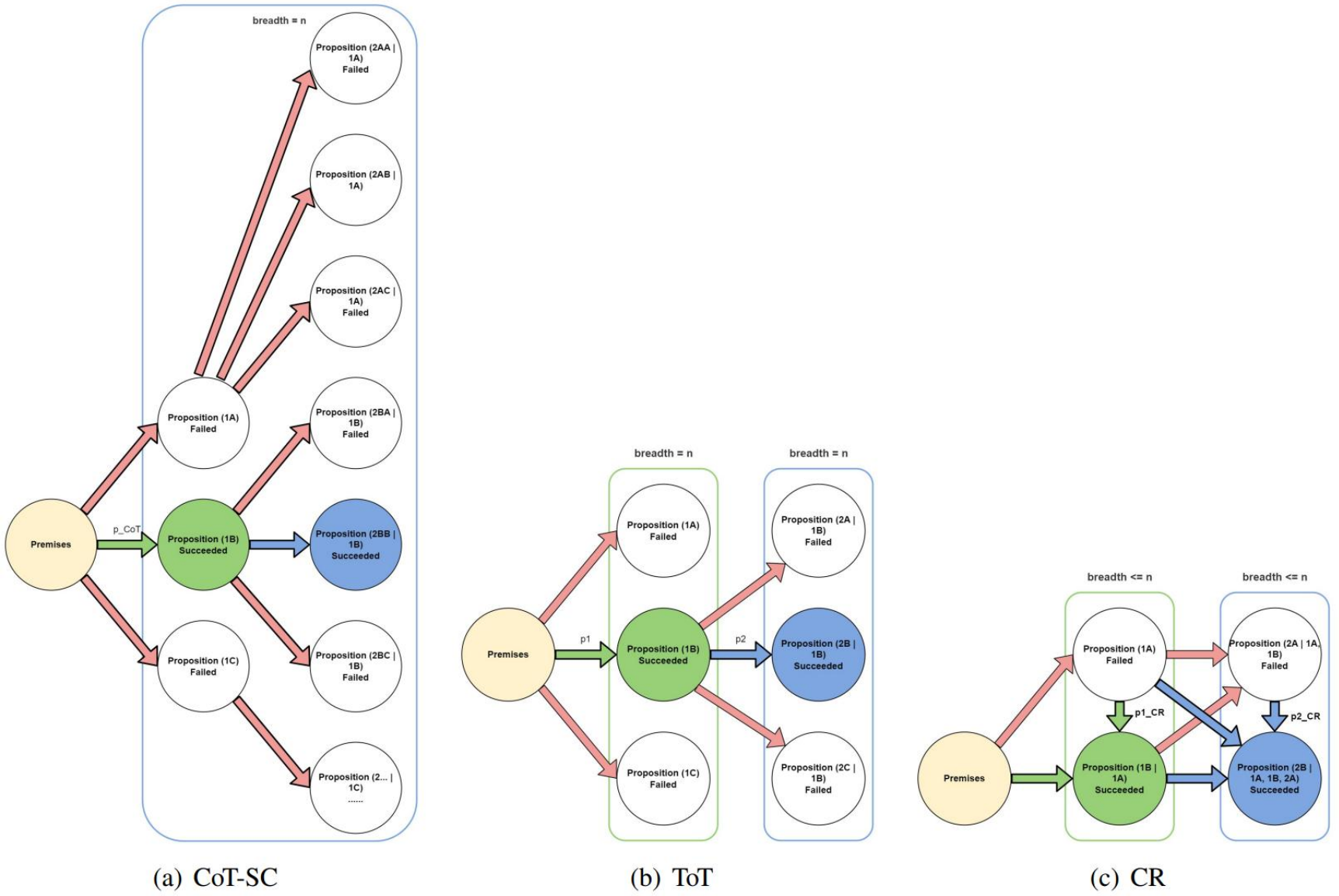
- Chain of Thought，思维链路

ToT

- Tree of Thoughts，思维树

CR

- Cumulative Reasoning，累积推理


breadth = n
Proposition (2AA | 1A) Failed
Proposition (2AB | 1A)
Proposition (2AC | 1A) Failed
Proposition (1A) Failed
Proposition (2BA | 1B) Failed
Premises
p_CoT
Proposition (1B) Succeeded
Proposition (2BB | 1B) Succeeded
Proposition (1C) Failed
Proposition (2BC | 1B) Failed
Proposition (2... | 1C) .....
(a) CoT-SC
breadth = n
breadth = n
Proposition (1A) Failed
Proposition (2A | 1B) Failed
Premises
p1
Proposition (1B) Succeeded
p2
Proposition (2B | 1B) Succeeded
Proposition (1C) Failed
Proposition (2C | 1B) Failed
(b) ToT
breadth <= n
breadth <= n
Proposition (1A) Failed
Proposition (2A | 1A, 1B) Failed
p1_CR
p2_CR
Premises
Proposition (1B | 1A) Succeeded
Proposition (2B | 1A, 1B, 2A) Succeeded
(c) CR

从简单粗暴到复杂完备

Table 4: Results for various approaches on Game of 24 using GPT-4.

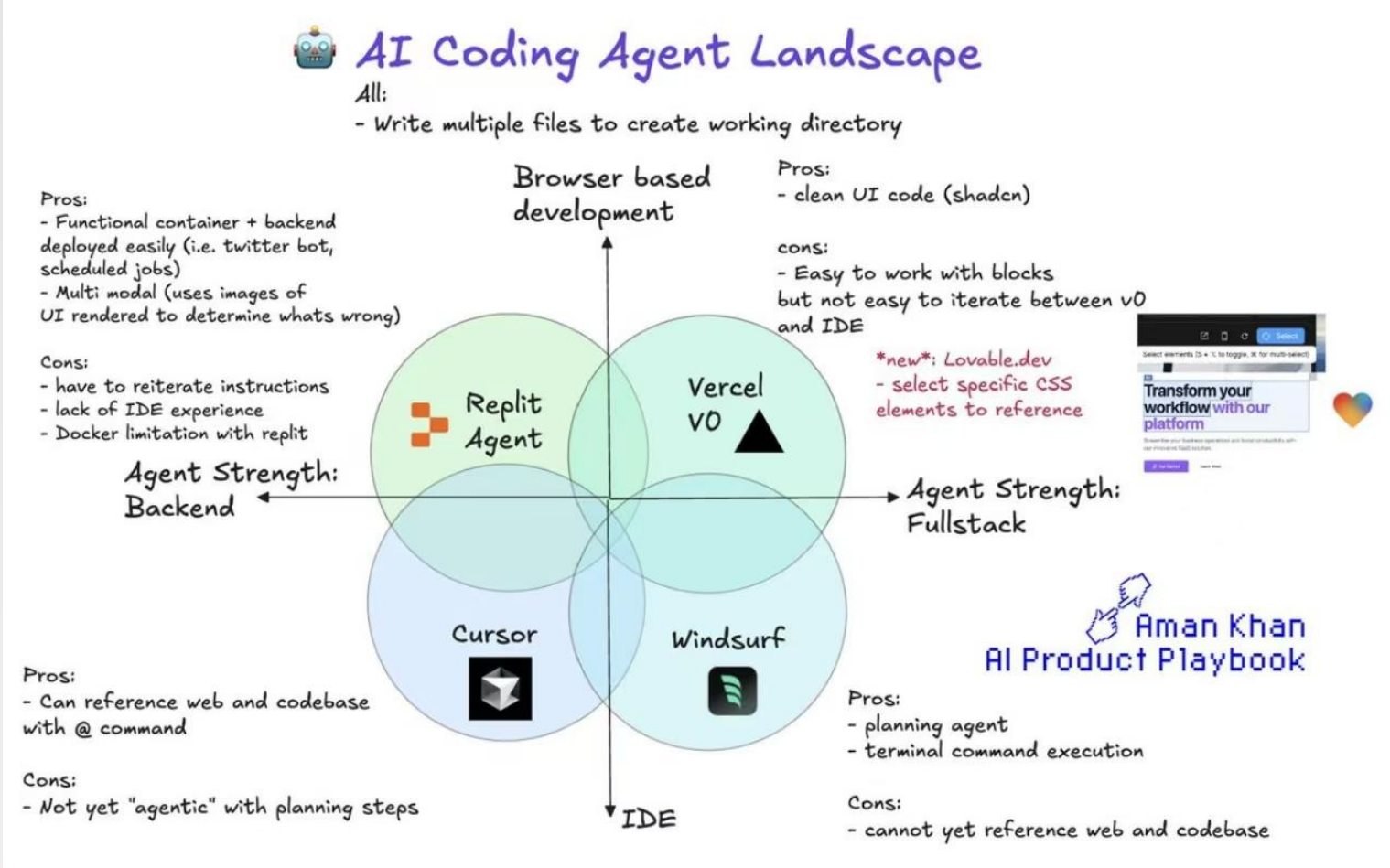
Method	Acc. ↑ (%)	# Visited states ↓
Direct	7.3	1
CoT	4.0	1
CoT-SC (k = 100)	9.0	100
Direct (best of 100)	33	100
CoT (best of 100)	49	100
ToT ($b = 5$)	74	61.72
CR (ours, $b = 1$)	84 (+10)	**11.68 (-50.04)**
CR (ours, $b = 2$)	94 (+20)	13.70 (-48.02)
CR (ours, $b = 3$)	97 (+23)	14.25 (-47.47)
CR (ours, $b = 4$)	97 (+23)	14.77 (-46.95)
CR (ours, $b = 5$)	**98 (+24)**	14.86 (-46.86)

Zhang, Yifan, et al. "Cumulative reasoning with large language models." arXiv preprint arXiv:2308.04371 (2023).

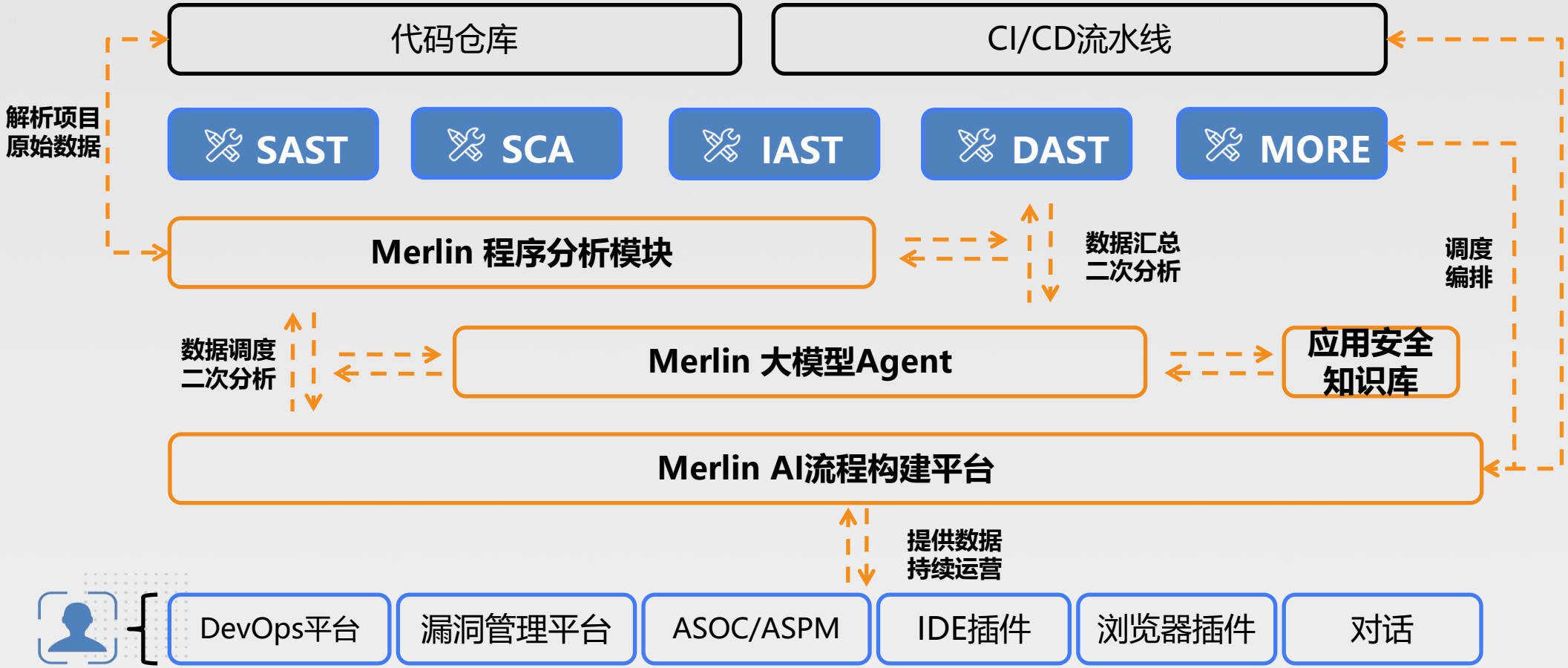
07.

全栈解决方案与边界感

赢家通吃还是融入生态


AI Coding Agent Landscape
All:
- Write multiple files to create working directory
Browser based development
Pros:
- Functional container + backend deployed easily (i.e. twitter bot, scheduled jobs)
- Multi modal (uses images of UI rendered to determine whats wrong)
Cons:
- have to reiterate instructions
- lack of IDE experience
- Docker limitation with replit
Replit Agent
Vercel v0
Pros:
- clean UI code (shadcn)
cons:
- Easy to work with blocks but not easy to iterate between v0 and IDE
new: Lovable.dev
- select specific CSS elements to reference
Transform your workflow with our platform
Agent Strength: Backend
Agent Strength: Fullstack
Cursor
Windsurf
Aman Khan
AI Product Playbook
Pros:
- Can reference web and codebase with @ command
Cons:
- Not yet "agentic" with planning steps
Pros:
- planning agent
- terminal command execution
Cons:
- cannot yet reference web and codebase
IDE

赢家通吃还是融入生态


代码仓库
CI/CD流水线
解析项目
原始数据
SAST
SCA
IAST
DAST
MORE
Merlin 程序分析模块
数据汇总
二次分析
调度
编排
数据调度
二次分析
Merlin 大模型Agent
应用安全
知识库
Merlin AI流程构建平台
提供数据
持续运营
DevOps平台
漏洞管理平台
ASOC/ASPM
IDE插件
浏览器插件
对话

08.

PoC, Prototype与产品化

做产品要克制

PoC

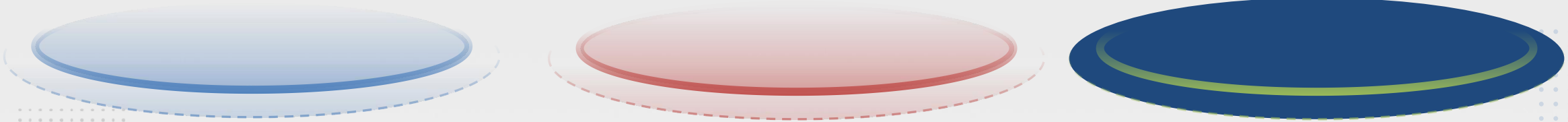
6+11

Prototype

4+6

产品级能力

3+3



AI应用安全运营平台-Merlin

关于蜚语

蜚语科技是一家专注于提供**应用安全**创新解决方案的信息安全企业，孵化自上海交通大学计算机系，创始团队由**数名博士**组成，拥有**十余年**的软件安全的前沿学术研究、产品研发和项目实施经验。

核心产品：AI-Powered 静态应用安全测试工具-Corax

AI-Driven 应用安全运营平台-Merlin

AI-Driven 编程辅助工具-Rosetta

知名投资机构加持：真格基金 相城金控 首发展集团

上海交通大学团队
知名CTF战队0ops，获得多个国内外安全竞赛冠军
上海市科技进步一等奖
数十篇顶会论文
数十项核心专利
多次行业峰会演讲
创业大赛金奖

2011-2018 孵化阶段

- 上海交通大学计算机系，国内最早一批从事应用程序分析、漏洞挖掘的研究团队
- 承接多项国家重点研发计划。
- **上海市科技进步一等奖**

2019-2022初 创阶段

- 蜚语科技成立
- 发布自研下一代代码安全分析平台Corax（2021）
- **Corax达到世界一流水平，完成多次“国产化替代”。**

2023-今 发展阶段

- Corax获得必维ISO 26262认证（国内首个）
- **开启AI产品线，在软件研发场景中引入AI能力，提质增效，对国外产品实现“弯道超车”。**
- Merlin实现商业化落地
- Rosetta通过上海网信办备案

他们选择了蜚语

安全服务

STARBUCKS
PORSCHE
IKEA
BRIDGEWATER
smith&nephew
CapitaLand
THE ASCOTT LIMITED
Abbott
DIOR
SHIMANO
Panasonic
SAINT-GOBAIN
oppo
商汤 sensetime
众安保险

CAICT 中国信通院
miHoYo TECH OTAKUS SAVE THE WORLD
富邦华一银行
tuya
卡亭科技 CHAITIN
CAII 中国工业互联网研究院 China Academy of Industrial Internet

安全产品

ecarX
ZEEKR
XPENG
IM 智己汽车
广汽集团
中国移动 China Mobile
iQIYI 爱奇艺 悦享品质
H3C
中科曙光 Sugon
AVIC
CSSC
NARI 南瑞集团 NARI GROUP CORPORATION

合作伙伴

gitee
极狐 GITLAB
阿里云
蚂蚁金服 ANT FINANCIAL
Open Security Research
DaoCloud
wizlynx group

OWASP
Open Worldwide Application Security Project

探索重塑 AI与大模型技术引领的安全革新实践

2024 OWASP中国安全技术论坛

THANKS