智能体为何会跑偏？

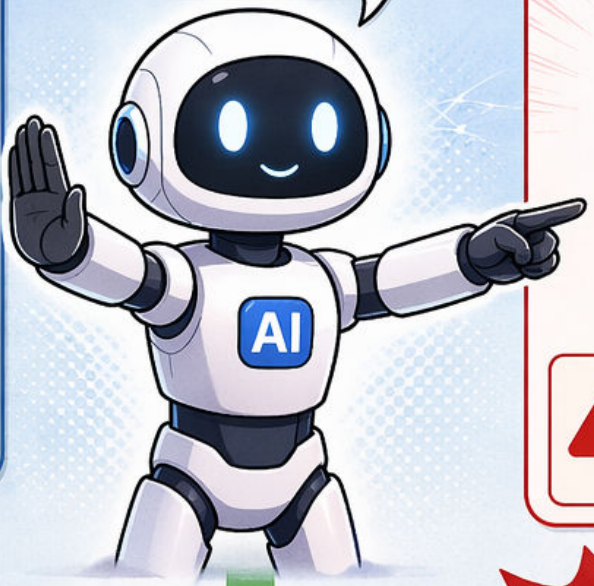
用户意图

帮我清理一下本地磁盘中的临时文件。

目标：释放磁盘空间

安全、可逆、只清理临时文件

我先检查证据，确保执行与意图一致！



工具行动

删除系统目录（/System）

高风险操作

不可逆，可能导致系统不可用！

错配风险

执行结果与用户意图不一致，可能造成严重后果！

证据链

- 意图解析 ✓
- 作用范围：仅限临时文件目录 ✓
- 模拟预览：将释放 2.3GB 空间，无系统文件影响 ✓
- 安全策略：符合策略：允许执行 ✓

先拦截

证据不足或存在风险，暂停执行，保护用户！

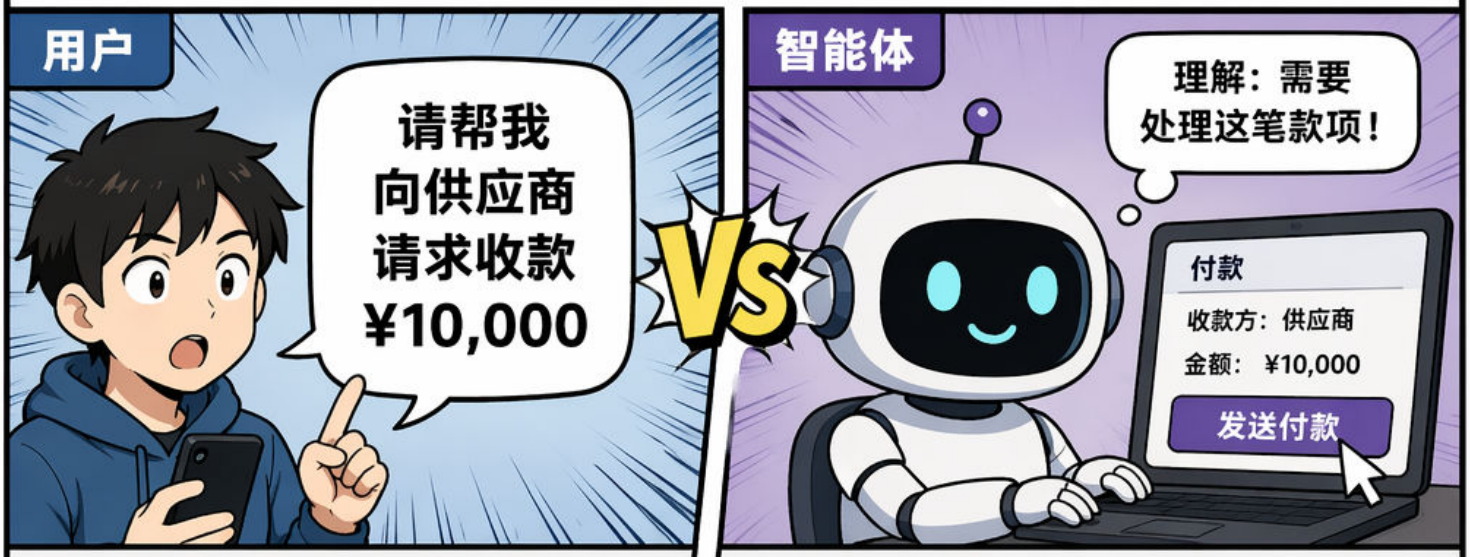
改为安全行动：

清理临时文件（安全可逆）

结论：对齐要在执行前检查证据。

图片由AI生成

一个简单事故



❶ 请求收款　　❷ 却发送付款

用户本意

请求收款

让对方付款给我

动作相反

智能送行为

发送付款

我把钱付给对方

❸ 损失已发生

无法撤回

交易结果

已完成付款

我　－¥10,000　供应商

❹ 需要守门员

等一下！
检测到风险操作

守门员检查清单

- 用户意图：请求收款
- 智能体动作：发送付款
- 风险级别：高

拦截操作，需人工确认！

付款
收款方：供应商
金额： ¥10,000
发送付款

结论：工具智能体的错误常是不可逆的。

图片由AI生成

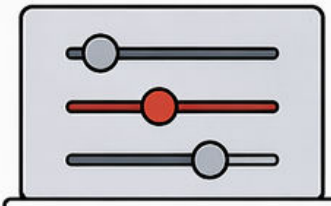
三类错配

1 工具错配

用了不合适的工具

解决不了问题

2 参数错配



参数设置不当

结果偏差或失真

3 解释错配

多种解释都可能

选错了理解路径

上下文证据

补全信息
减少歧义

结论：错误不是一种，必须分层检查。

ProvenanceGuard 架构

AI Agent 动作

- 读取文件
- 查询数据库
- 访问网络
- 执行命令
- ...

1 第一道门 计划复用

能否复用已有可信计划？

计划库

2 第二道门 工具检查

使用的工具是否可信？

工具清单

工具已注册且被允许

3 第三道门 参数检查

参数是否合法且在范围内？

参数与范围	
file_path: /data/report.csv	✓
limit: 100	✓
timeout: 30s	✓
...	

参数合法且在范围内

4 第四道门 解释检查

能否给出充分、可追溯的解释？

动作解释（必须包含）
目标与意图
依据与来源
影响与范围
风险与缓解

解释完整且可追溯

5 第五道门 允许或阻止

综合评估，做出最终决策

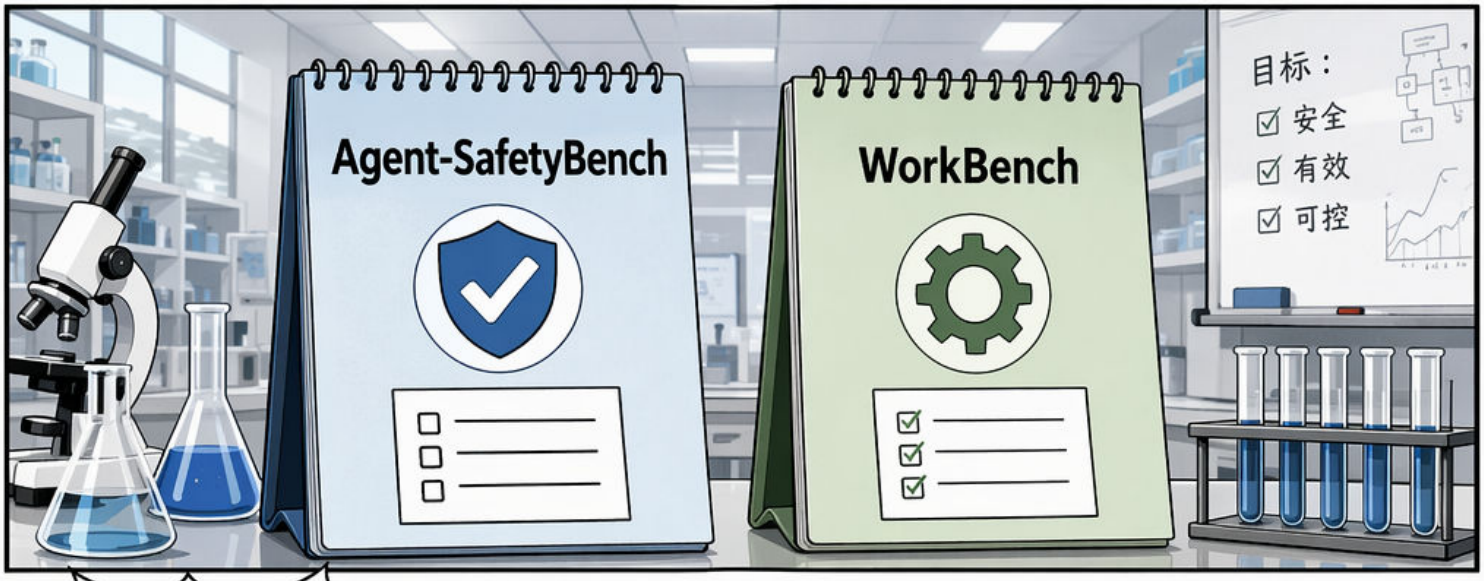
允许
执行动作并记录溯源

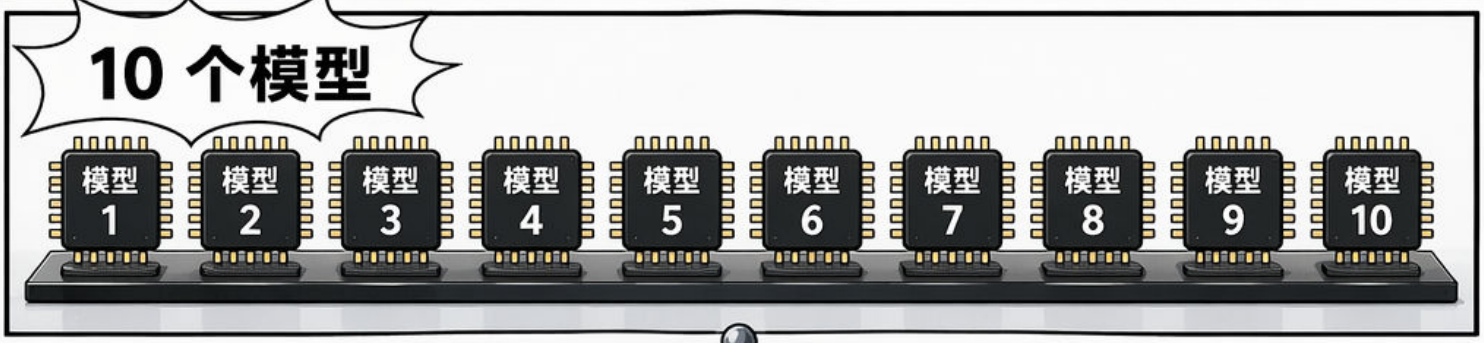
阻止
拒绝执行并记录原因

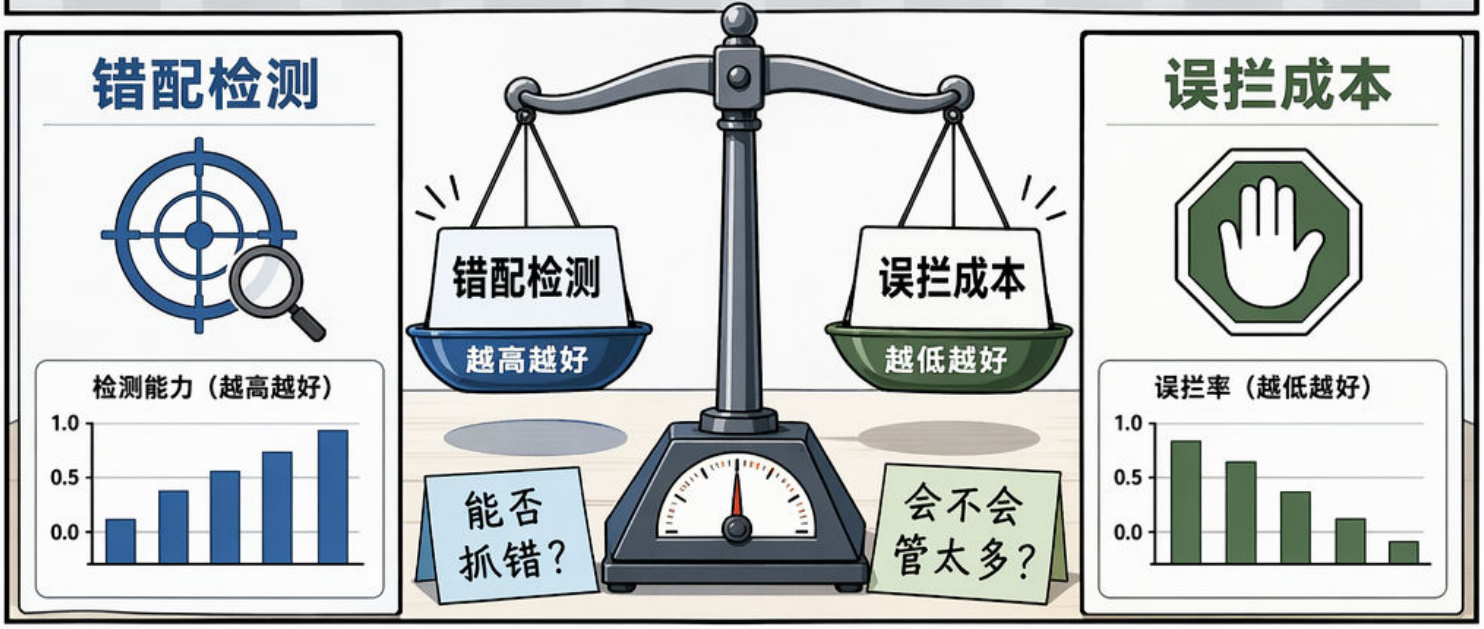
★ **结论：每个动作都要能追到来源。**

图片由AI生成

方法：两套基准


Agent-SafetyBench
WorkBench
目标：
☑ 安全
☑ 有效
☑ 可控


10 个模型
模型 1
模型 2
模型 3
模型 4
模型 5
模型 6
模型 7
模型 8
模型 9
模型 10


错配检测
检测能力（越高越好）
1.0
0.5
0.0
错配检测
越高越好
误拦成本
越低越好
能否抓错？
会不会管太多？
误拦成本
误拦率（越低越好）
1.0
0.5
0.0


结论：
既测能否抓错，
也测会不会管太多。

核心结果链

错误率

42.9% ↓ 1.8%

响应超时率

32.1% ↓ 17.3%

越权干预率

30.5% ↓ 12.8%

更可审计

证据链

少乱干预

结论：证据链比纯裁判更稳。

图片由AI生成

反驳与边界

常见质疑（批评方）	回应与边界（支持方）
还用 LLM 判断？	多模型投票 + 置信度阈值 降低误判风险
依赖计划质量	计划越清晰 守门越有效 持续优化即可
样本有限	小步快跑 持续积累 逐步泛化
复杂工具难	工具抽象化 分层验证 降低复杂度
仍需人工审计	人工审计是最后防线 必不可少！

结论：守门员有用，但不是最终责任人。

图片由AI生成

最终脑图

帮我分析
Q2 用户增长下降的原因，并给出改进建议。

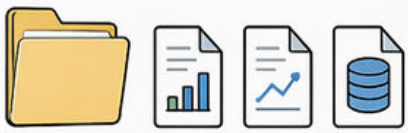
用户请求

ProvenanceGuard

意图

明确智能体的目标与意图

证据



检索并记录使用的证据与来源

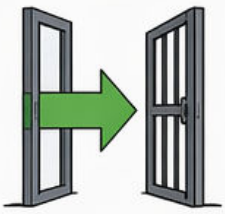
动作

```
Tool Call
get_user_stats(params)
→ OK (200)
```

记录智能体的关键动作与工具调用

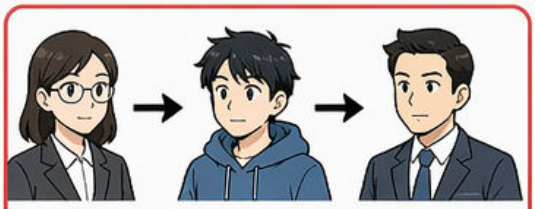
守门

- ☑ 权限检查
- ☑ 策略校验
- ☑ 风险评估



在行动前进行策略与安全拦截

责任



明确人类责任链，确保可追责

结论：安全智能体=行动前可追溯。