

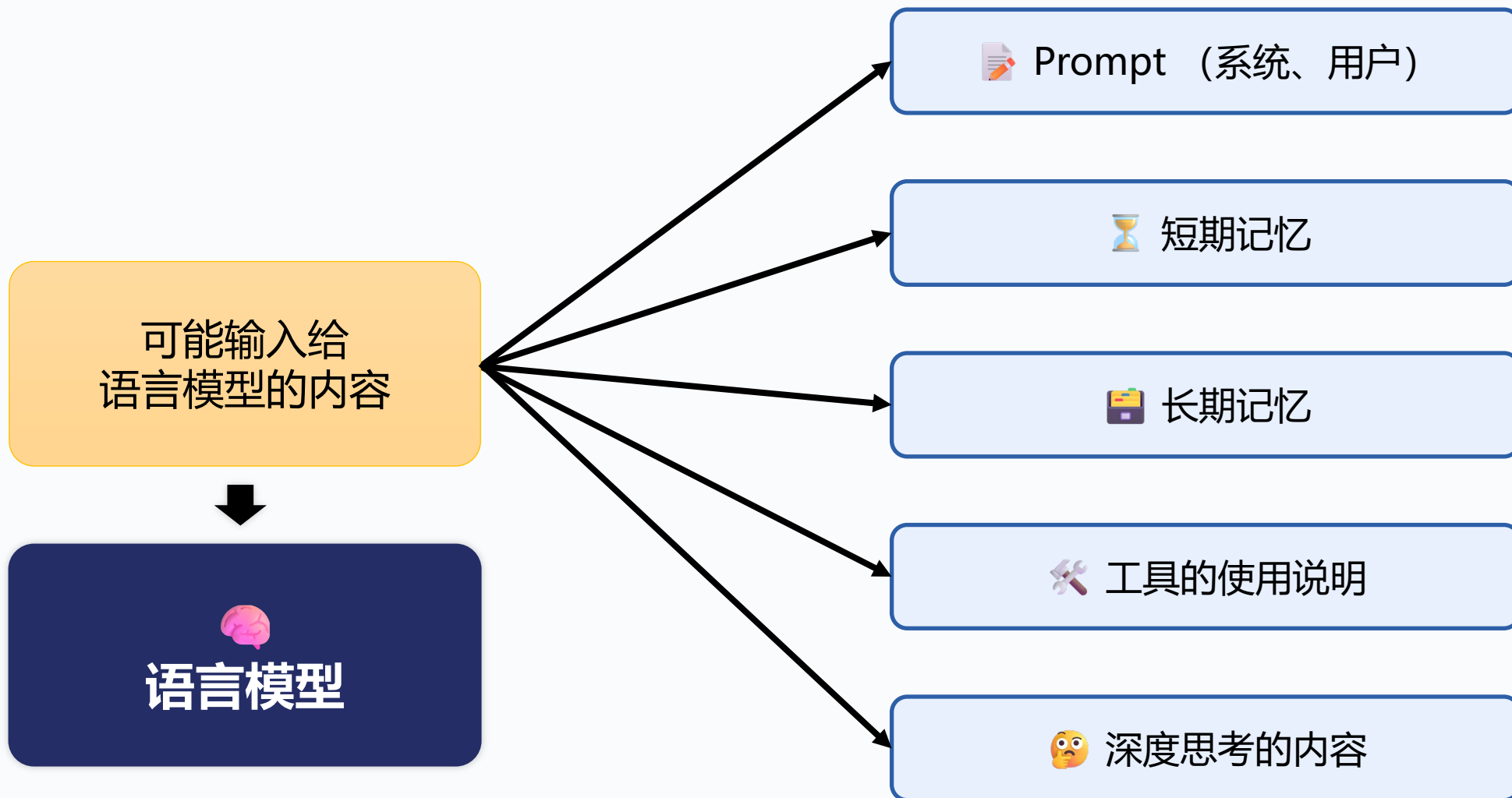
# 神经网络与深度学习 2026

## 第3讲 深入上下文工程

信息科学技术学院 吴瀚霖

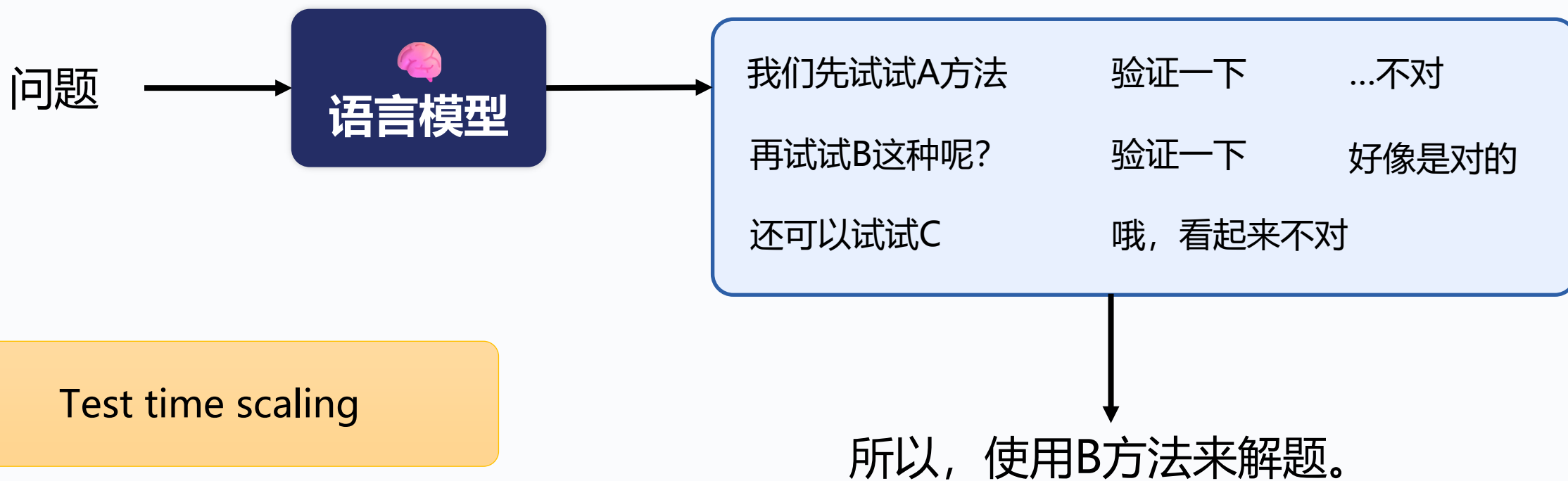
hlwu@bfsu.edu.cn

# 回顾：扩展语言模型的能力

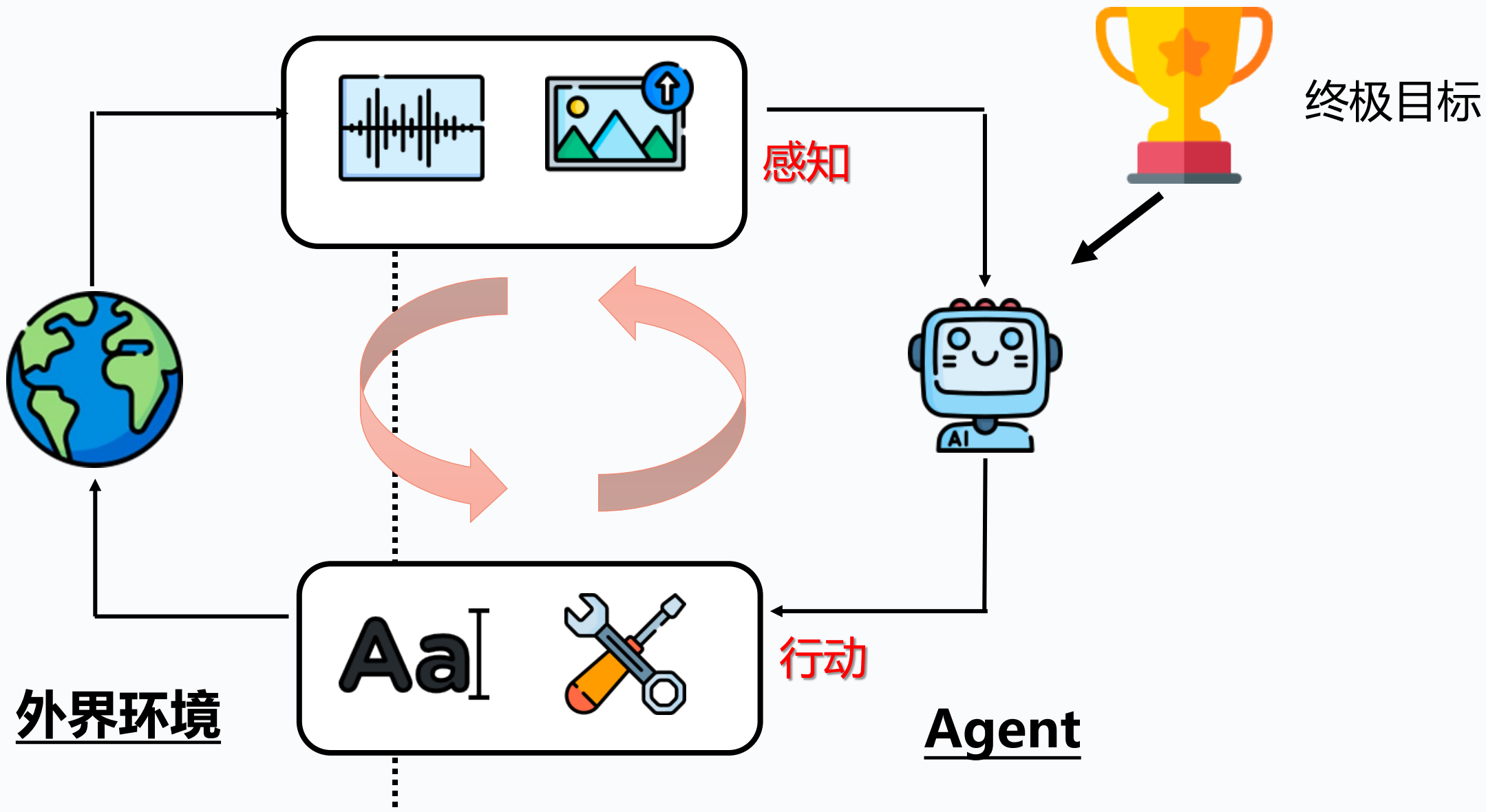


# Context中还可能模型的思考过程

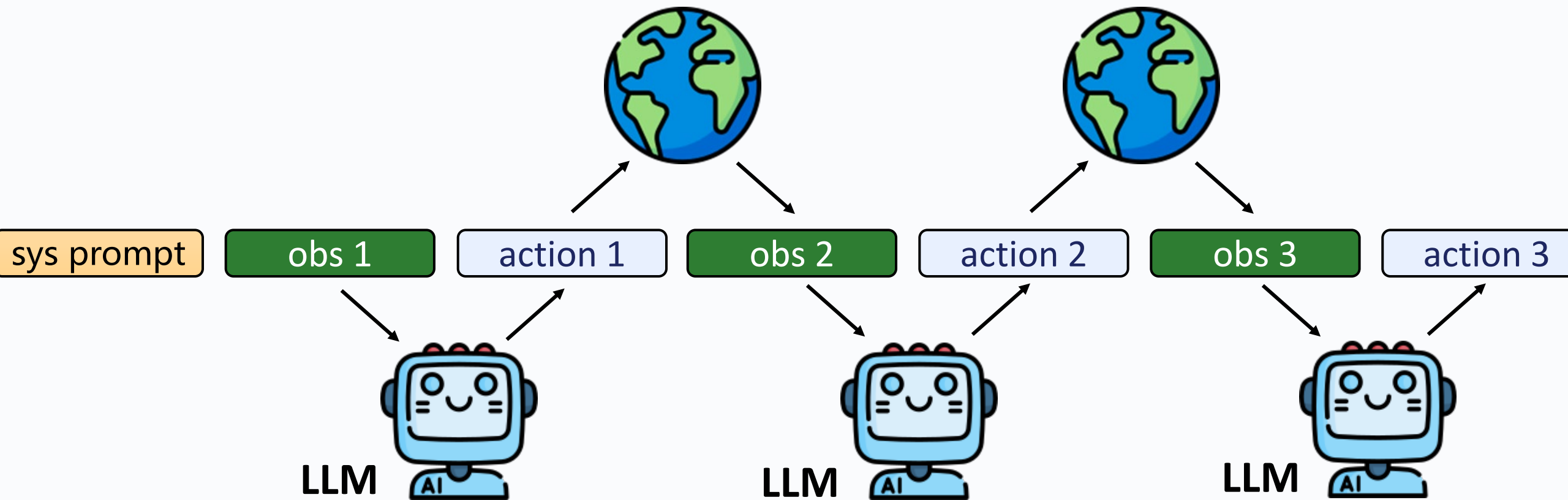
许多模型都支持深度思考模式：Deepseek R系列、Qwen 3.5...



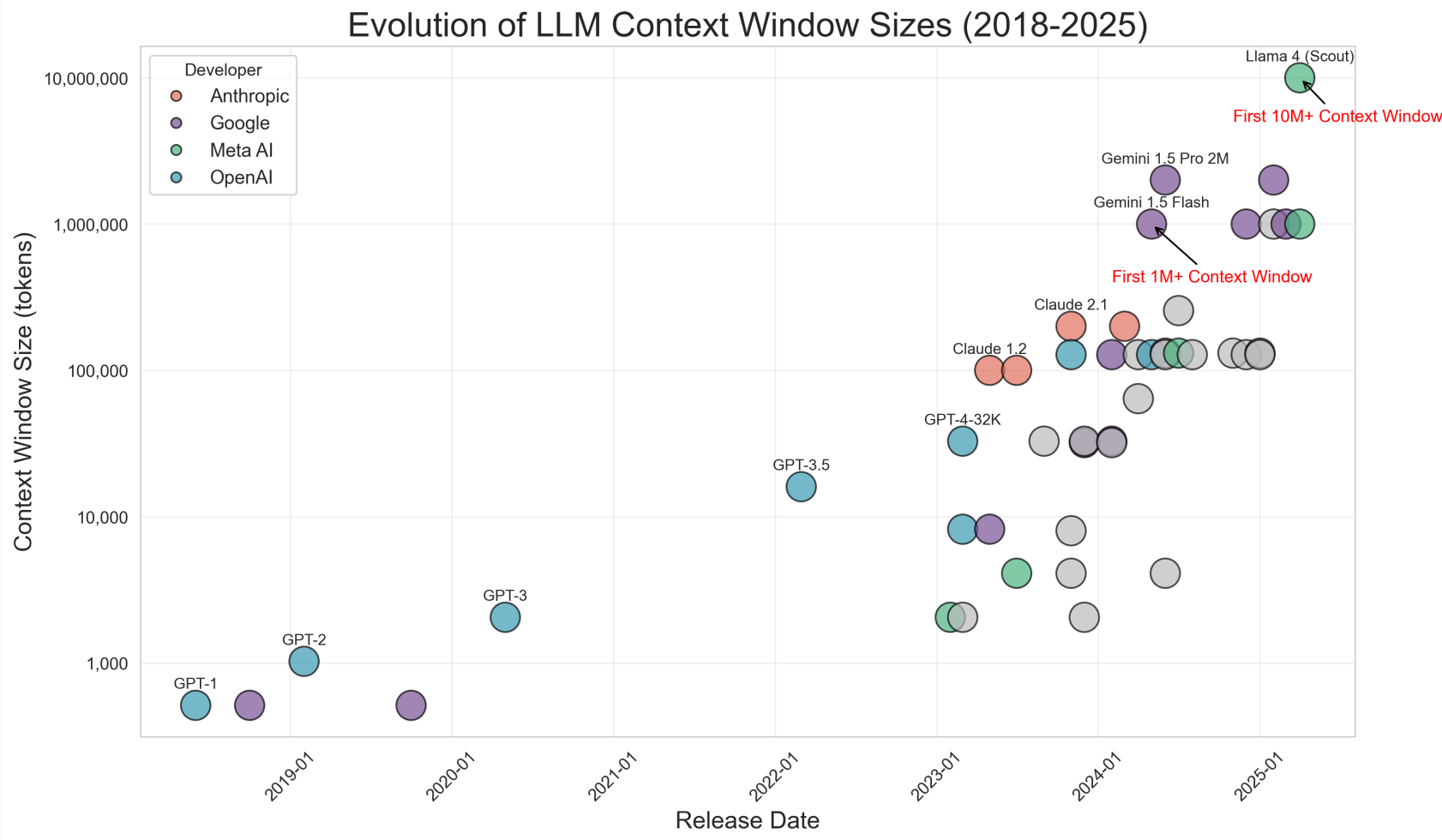
# AI Agent的技术框架



# 为什么需要Context Engineering

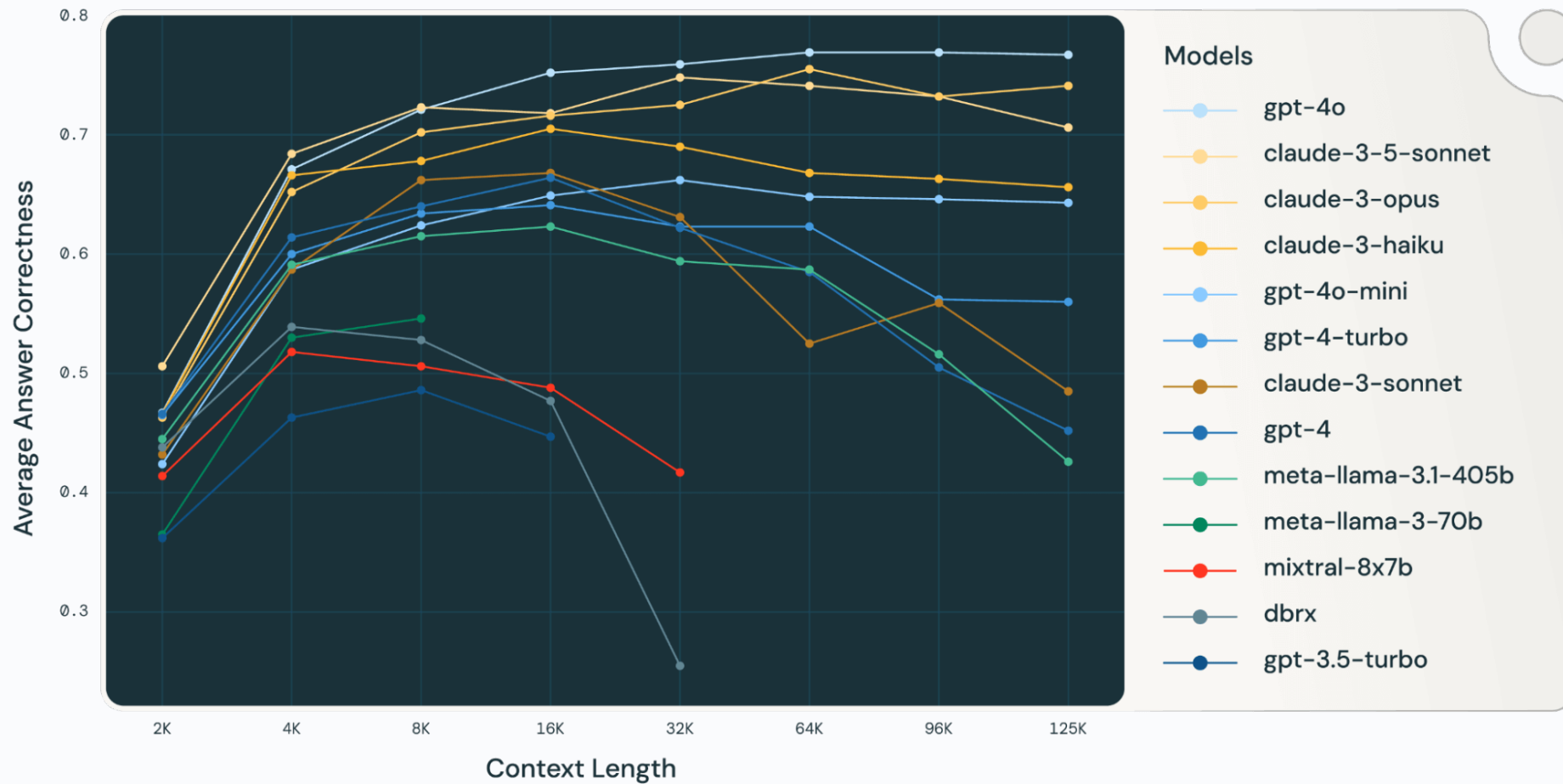


# LLM上下文窗口的进化



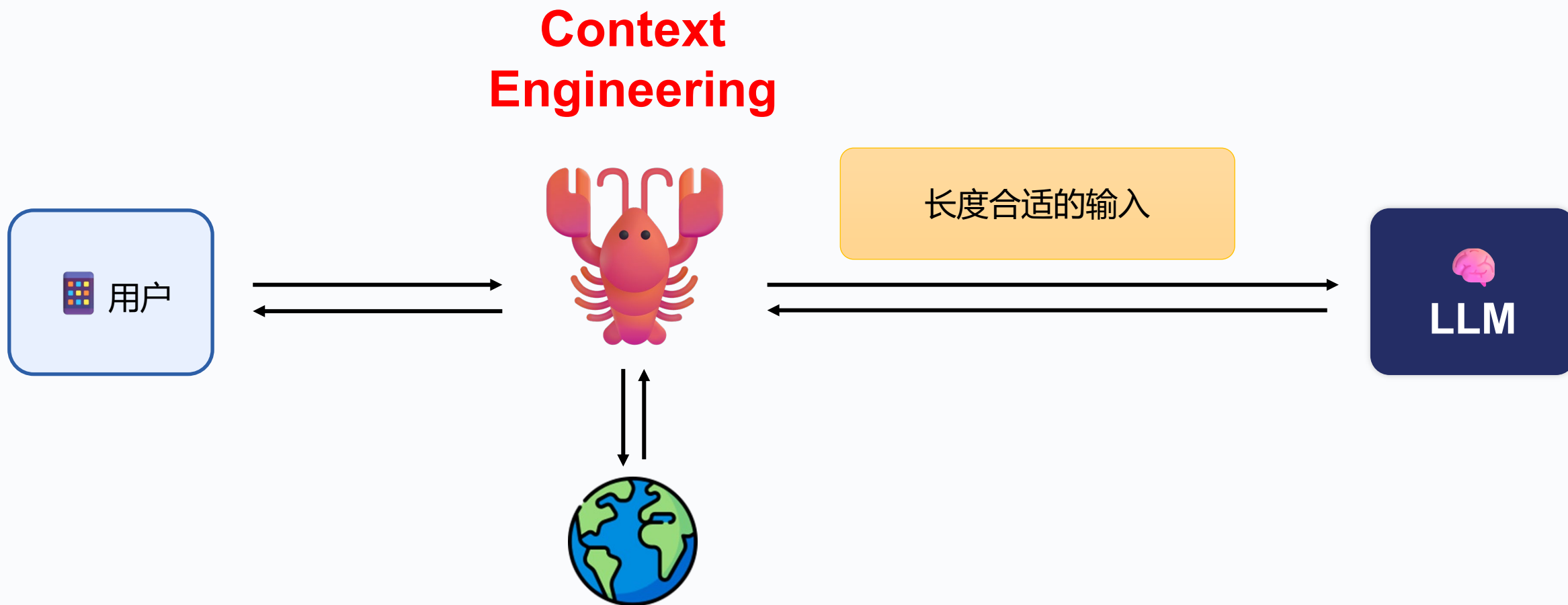
# 上下文：能读 ≠ 能读懂

以RAG为例



# Context Engineering

核心：合理选择让LLM看到什么



# Context Engineering

---

$I_1 \leftarrow$  initial input

$C_1 \leftarrow$  empty

For  $t = 1$  to  $\infty$

$$O_t = LLM(I_t, C_t)$$

$$C_{t+1} \leftarrow C_t | I_t | O_t$$

$I_1 \leftarrow$  initial input

$C_1 \leftarrow$  empty

For  $t = 1$  to  $\infty$

$$O_t = LLM(I_t, C_t)$$

$$C_{t+1} \leftarrow F(C_t, I_t, O_t)$$

# Context Engineering

---

选择

压缩

多智能体

# Context Engineering

---

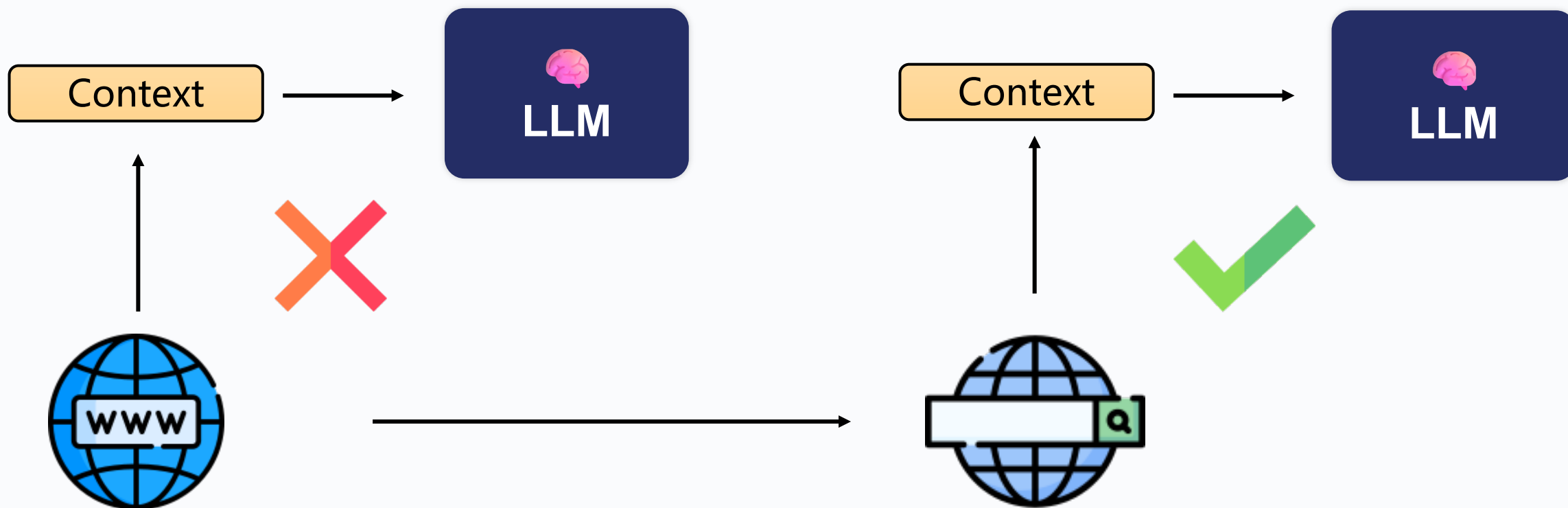
选择

压缩

多智能体

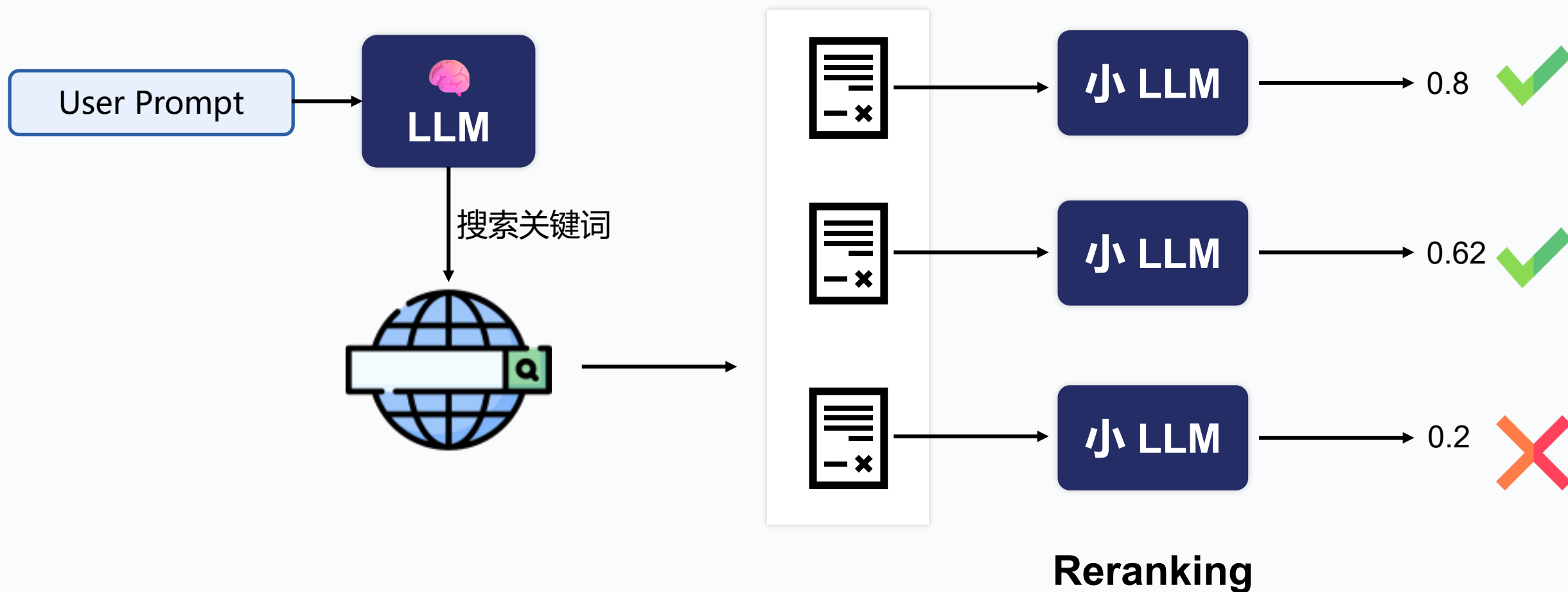
# 选择需要的内容

RAG (Retrieval-Augmented Generation)

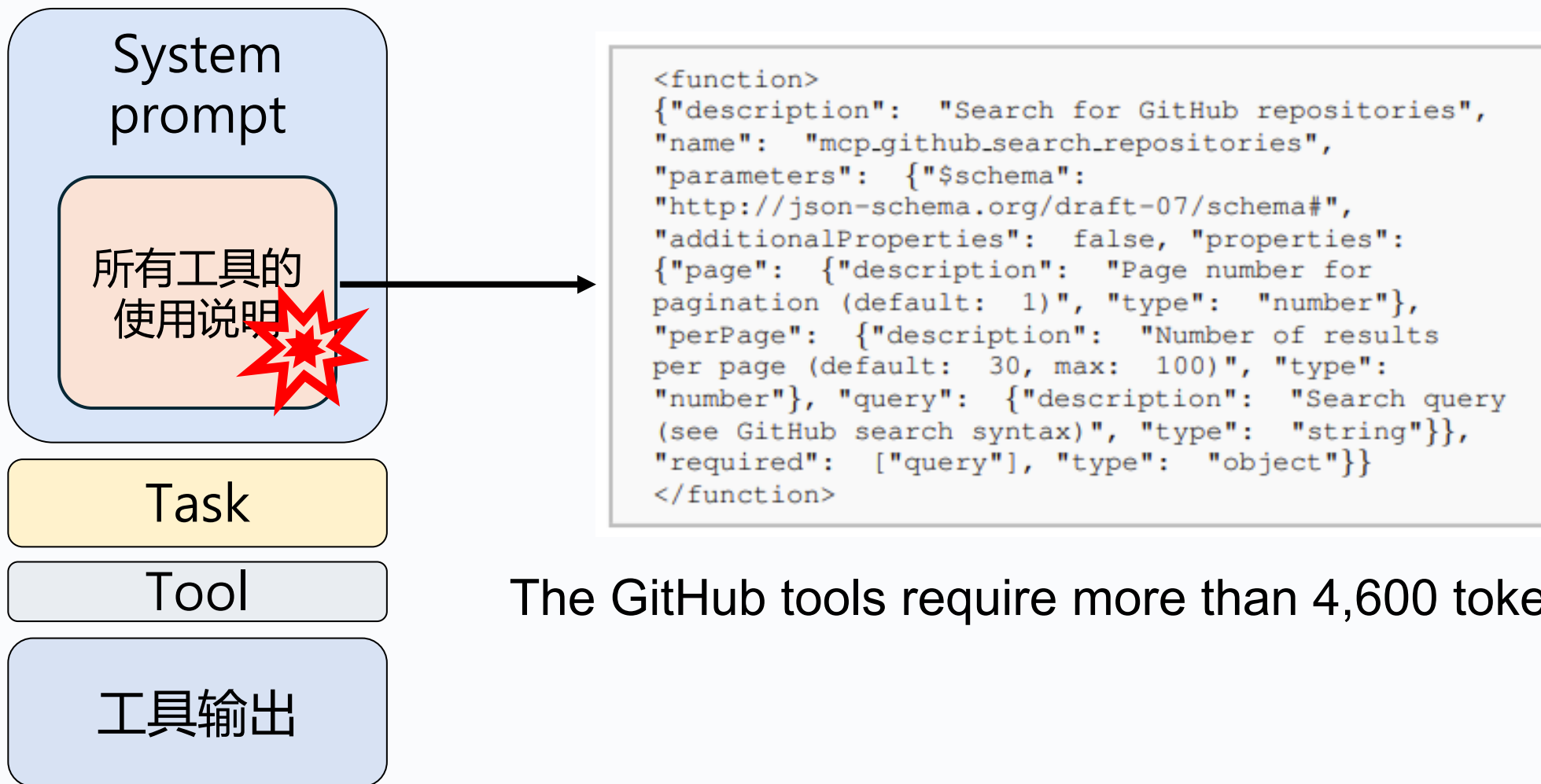


# 选择需要的内容

RAG (Retrieval-Augmented Generation)

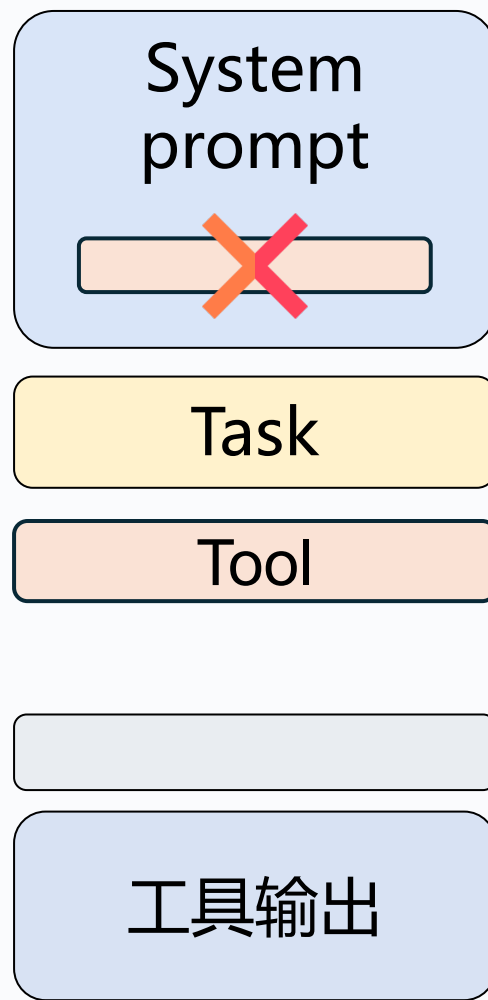
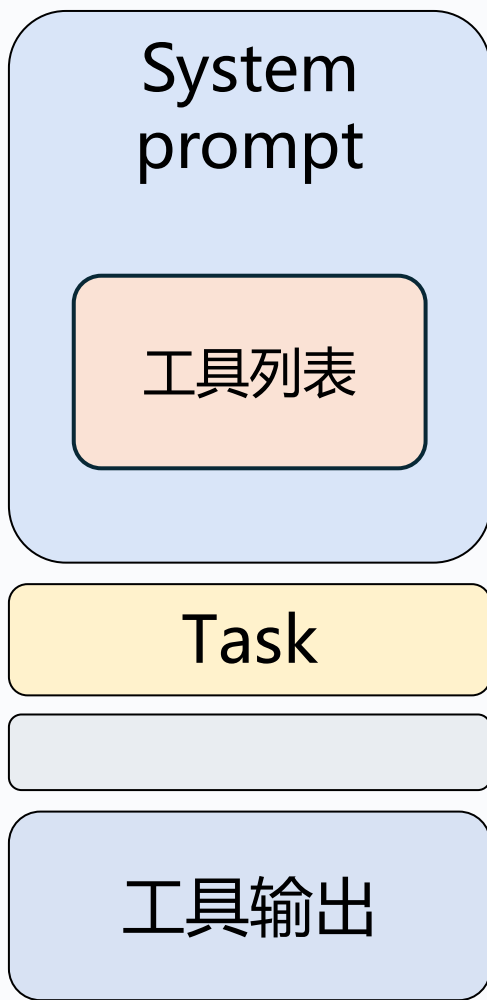


# 选择需要的工具

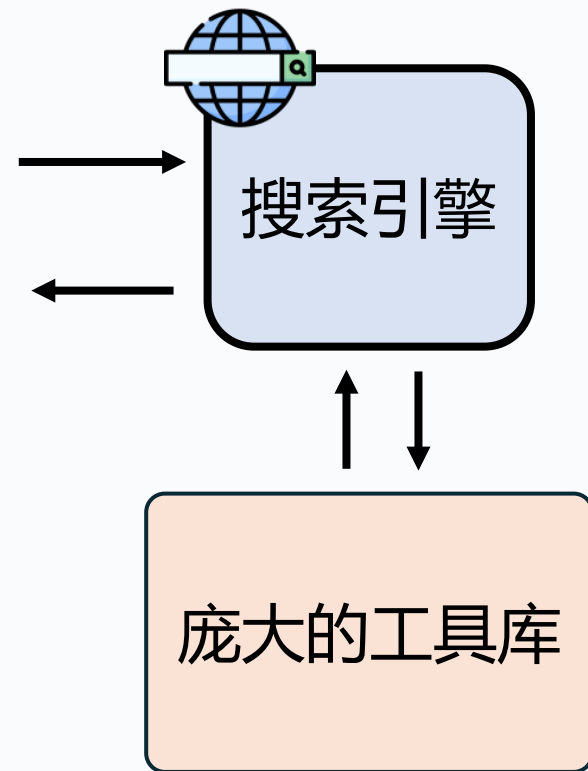


The GitHub tools require more than 4,600 tokens.

# 选择需要的工具

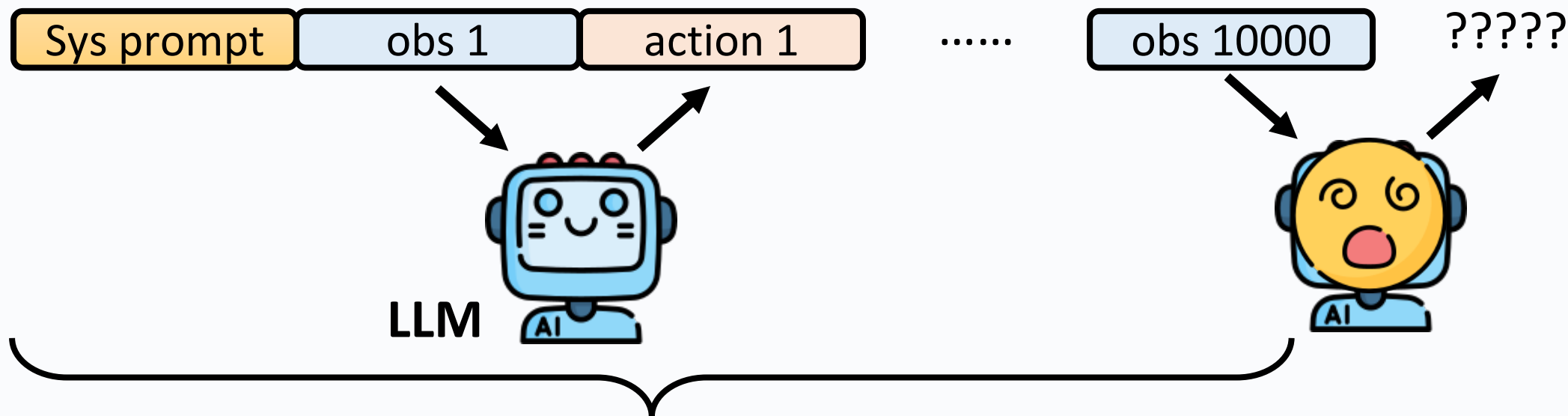


能否再厉害一点?  
让LLM自己说需要什么工具!



# 选择需要的记忆

挑选记忆



不断回忆一生经历过的所有事情...

# 选择需要的记忆

[https://reverie.herokuapp.com/arXiv\\_Demo](https://reverie.herokuapp.com/arXiv_Demo)



Park, Joon Sung, et al. "Generative agents: Interactive simulacra of human behavior." *Proceedings of the 36th annual acm symposium on user interface software and technology*. 2023.

# 选择需要的记忆

## 记忆流 (memory stream)

- 记录了agent的所有经历
- 最基本的元素是**Observation**
  - 自己做过的事
  - 环境中对象的状态
  - 感知到的其他agent的行为

太琐碎了!

## Memory Stream

```
2023-02-13 22:48:20: desk is idle
2023-02-13 22:48:20: bed is idle
2023-02-13 22:48:10: closet is idle
2023-02-13 22:48:10: refrigerator is idle
2023-02-13 22:48:10: Isabella Rodriguez is stretching
2023-02-13 22:33:30: shelf is idle
2023-02-13 22:33:30: desk is neat and organized
2023-02-13 22:33:10: Isabella Rodriguez is writing in her journal
2023-02-13 22:18:10: desk is idle
2023-02-13 22:18:10: Isabella Rodriguez is taking a break
2023-02-13 21:49:00: bed is idle
2023-02-13 21:48:50: Isabella Rodriguez is cleaning up the
kitchen
2023-02-13 21:48:50: refrigerator is idle
2023-02-13 21:48:50: bed is being used
2023-02-13 21:48:10: shelf is idle
2023-02-13 21:48:10: Isabella Rodriguez is watching a movie
2023-02-13 21:19:10: shelf is organized and tidy
2023-02-13 21:18:10: desk is idle
2023-02-13 21:18:10: Isabella Rodriguez is reading a book
2023-02-13 21:03:40: bed is idle
2023-02-13 21:03:30: refrigerator is idle
2023-02-13 21:03:30: desk is in use with a laptop and some papers
on it
```

...

# 选择需要的记忆

## 即时性

- 最近发生过的事情权重更高;

## 重要性

- 区分平凡的事件和重要的事件;
- 直接用LLM打分。 →

yaml

```
1 | On the scale of 1 to 10, where 1 is purely mundane  
  | (e.g., brushing teeth, making bed) and 10 is extremely  
  | poignant (e.g., a break up, college acceptance), rate  
  | the likely poignancy of the following piece of memory.  
2 |  
3 | Memory: buying groceries at The Willows Market and  
  | Pharmacy Rating: <fill in>
```

## 相关性

- 与当前情况相关的记忆对象分配更高的分数，使用向量检索引擎即可。

# 选择需要的记忆

## OpenClaw的记忆选择机制

### ## Memory Recall

Before answering anything about prior work, decisions, dates, people, preferences, or todos: run `memory_search` on `MEMORY.md + memory/*.md`; then use `memory_get` to pull only the needed lines.

为什么读取memory还需要一个工具?

`memory_get`: reads a specific memory Markdown file (workspace-relative), optionally from a starting line and for N lines.

# Context Engineering

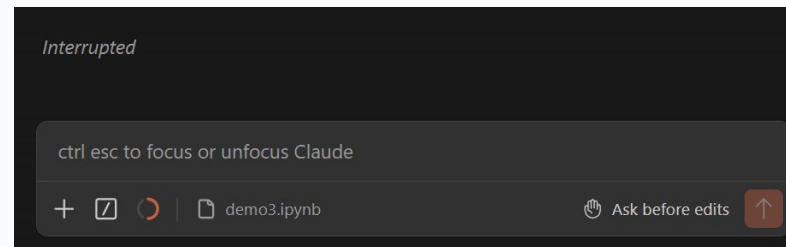
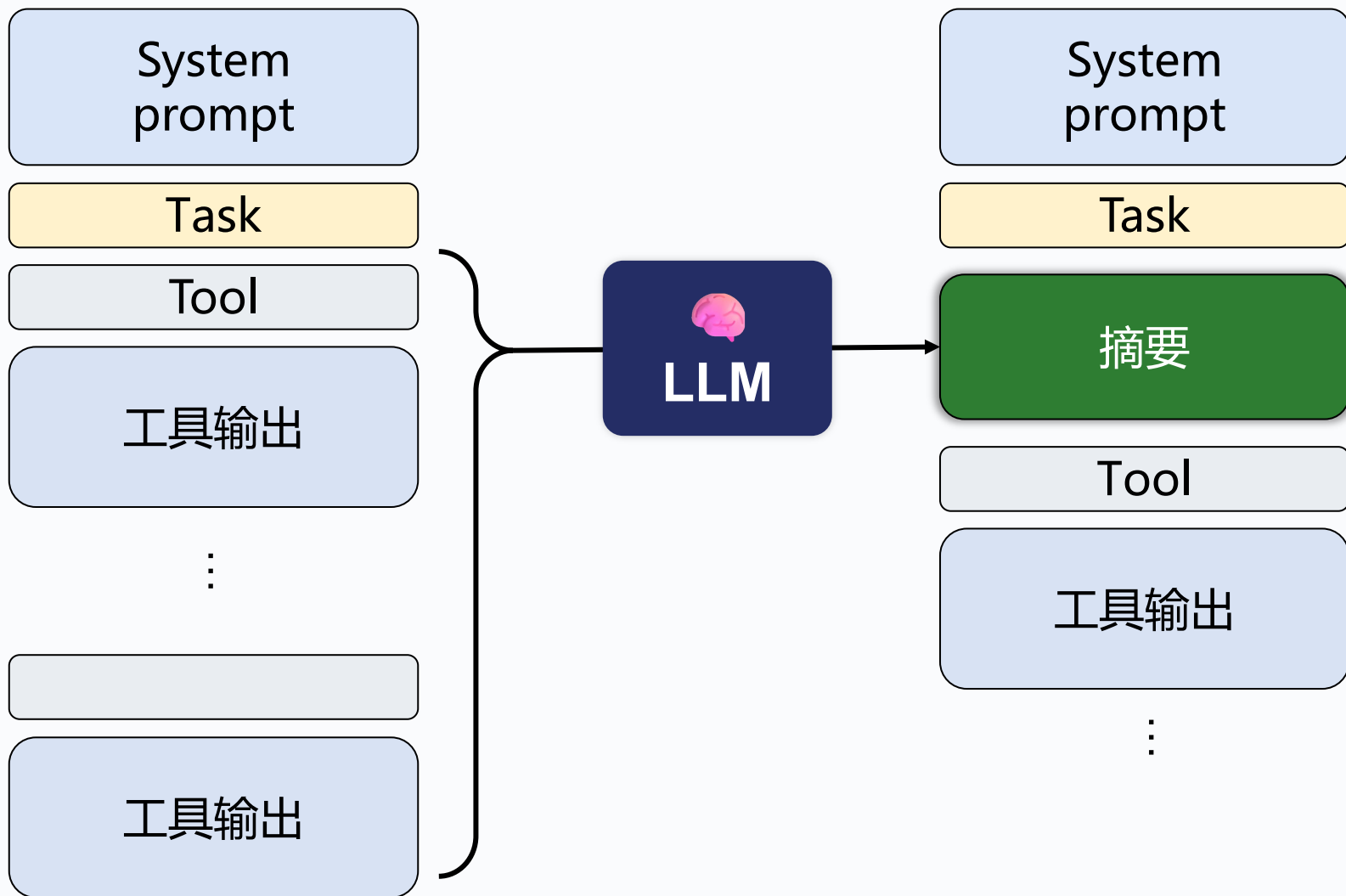
---

选择

压缩

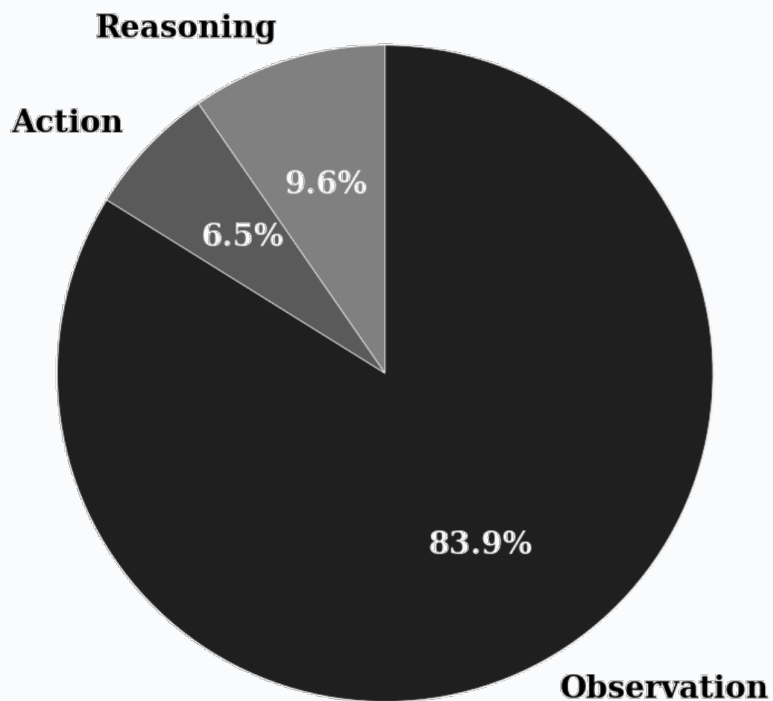
多智能体

# 压缩

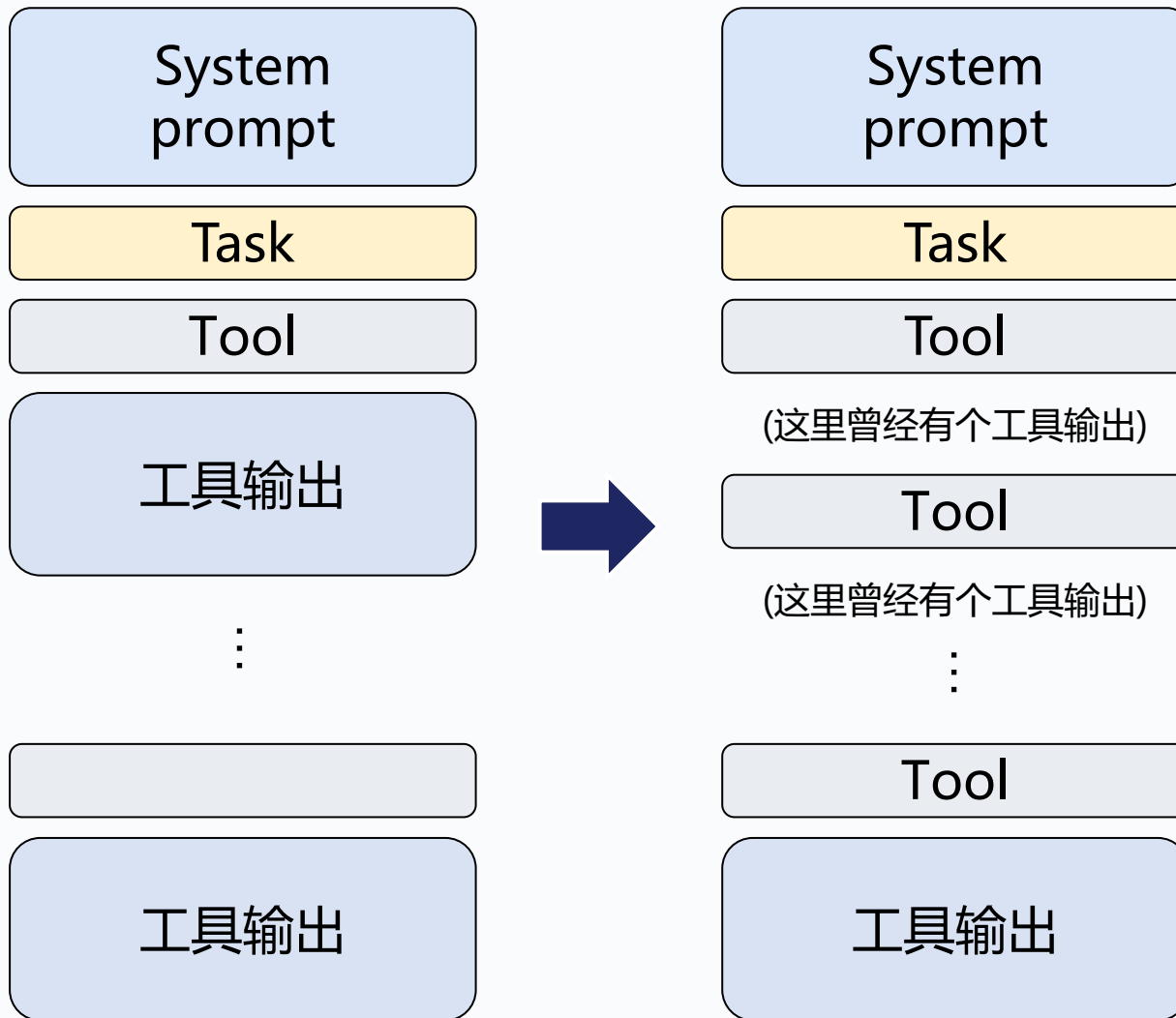


# 压缩：Observation Mask

Token Type Distribution - Raw Agent

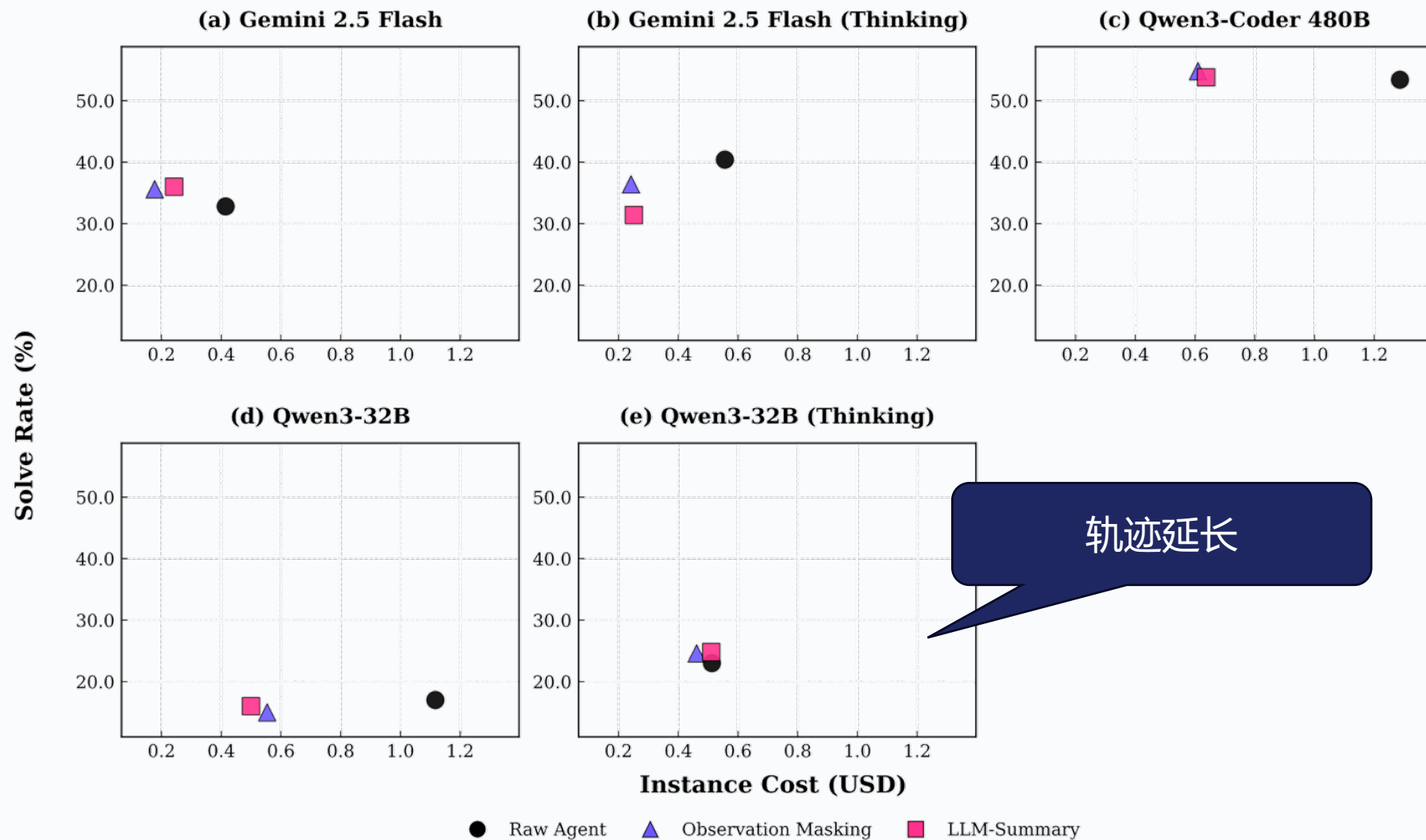


<https://arxiv.org/abs/2508.21433>

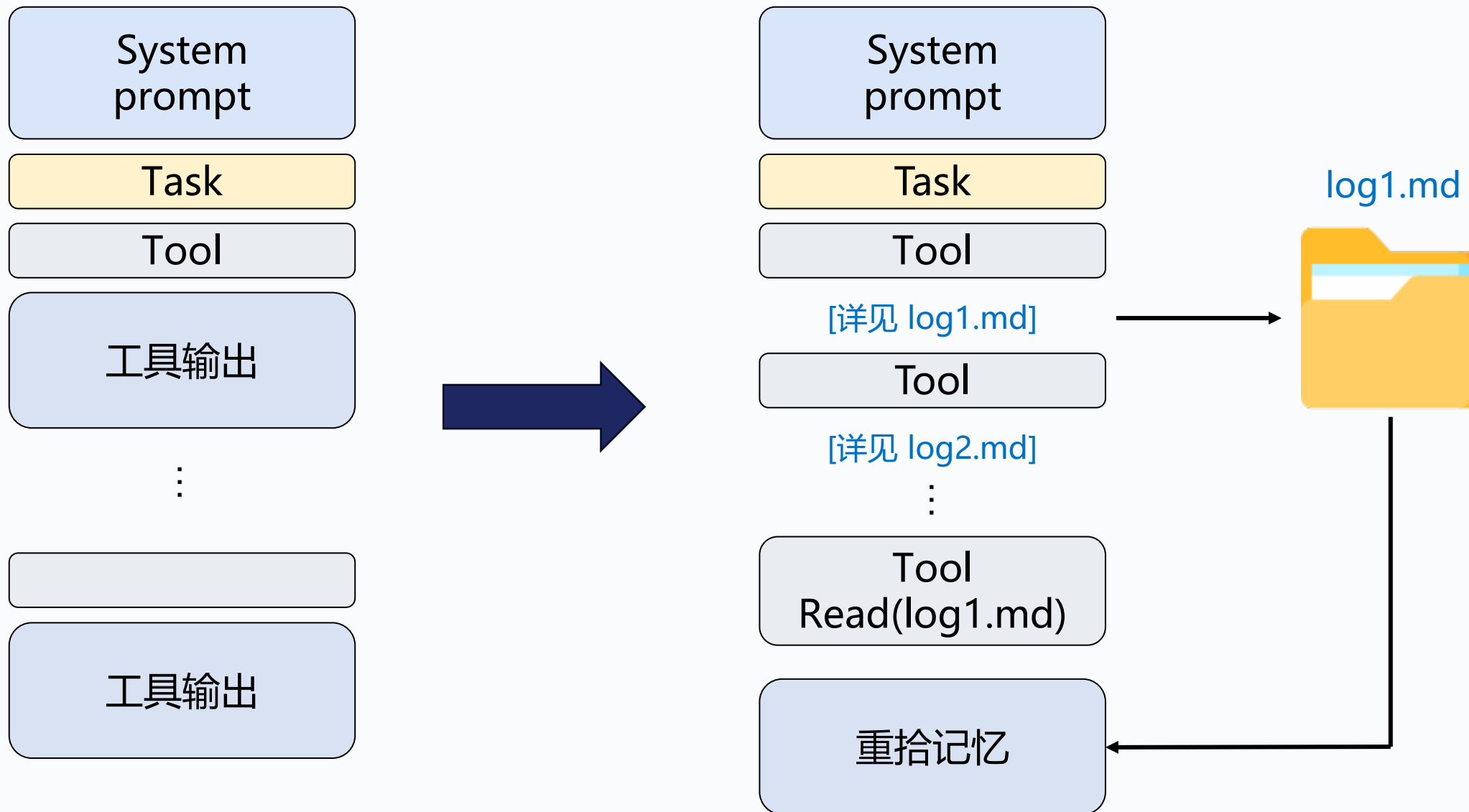


# 压缩: Observation Mask

SWE-bench



# 压缩



# Context Engineering

$I_1 \leftarrow$  initial input

$C_1 \leftarrow$  empty

For  $t = 1$  to  $\infty$

$$O_t = LLM(I_t, C_t)$$

$$C_{t+1} \leftarrow F(C_t, I_t, O_t)$$

$I_1 \leftarrow$  initial input

$C_1 \leftarrow$  empty

For  $t = 1$  to  $\infty$

$$O_t = LLM(I_t, P_t)$$

$$C_{t+1} \leftarrow F(C_t, I_t, O_t)$$

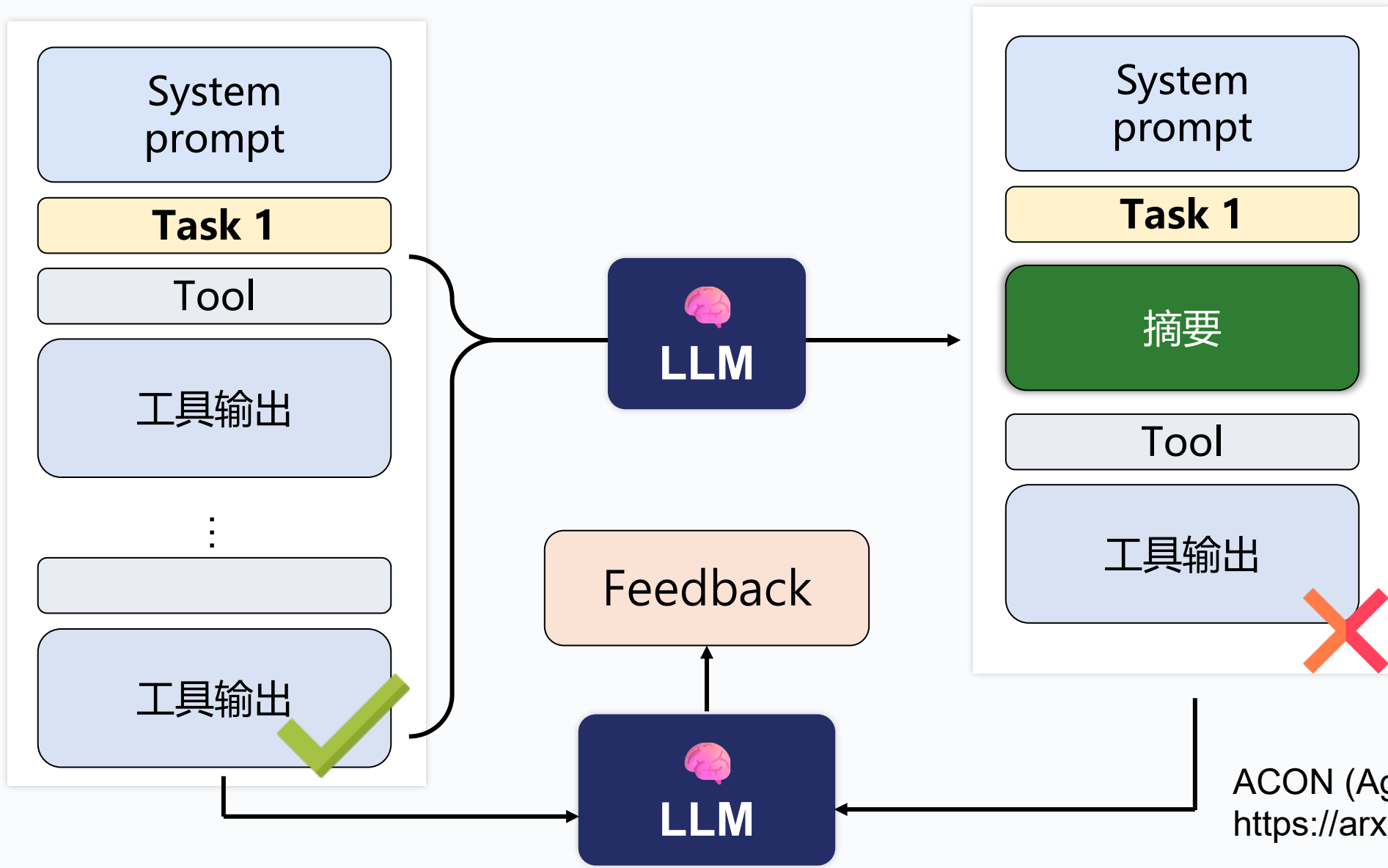
$\{P_{t+1}, M_{t+1}\}$

$\{P_t, M_t\}$

$$C = \{P, M\}$$

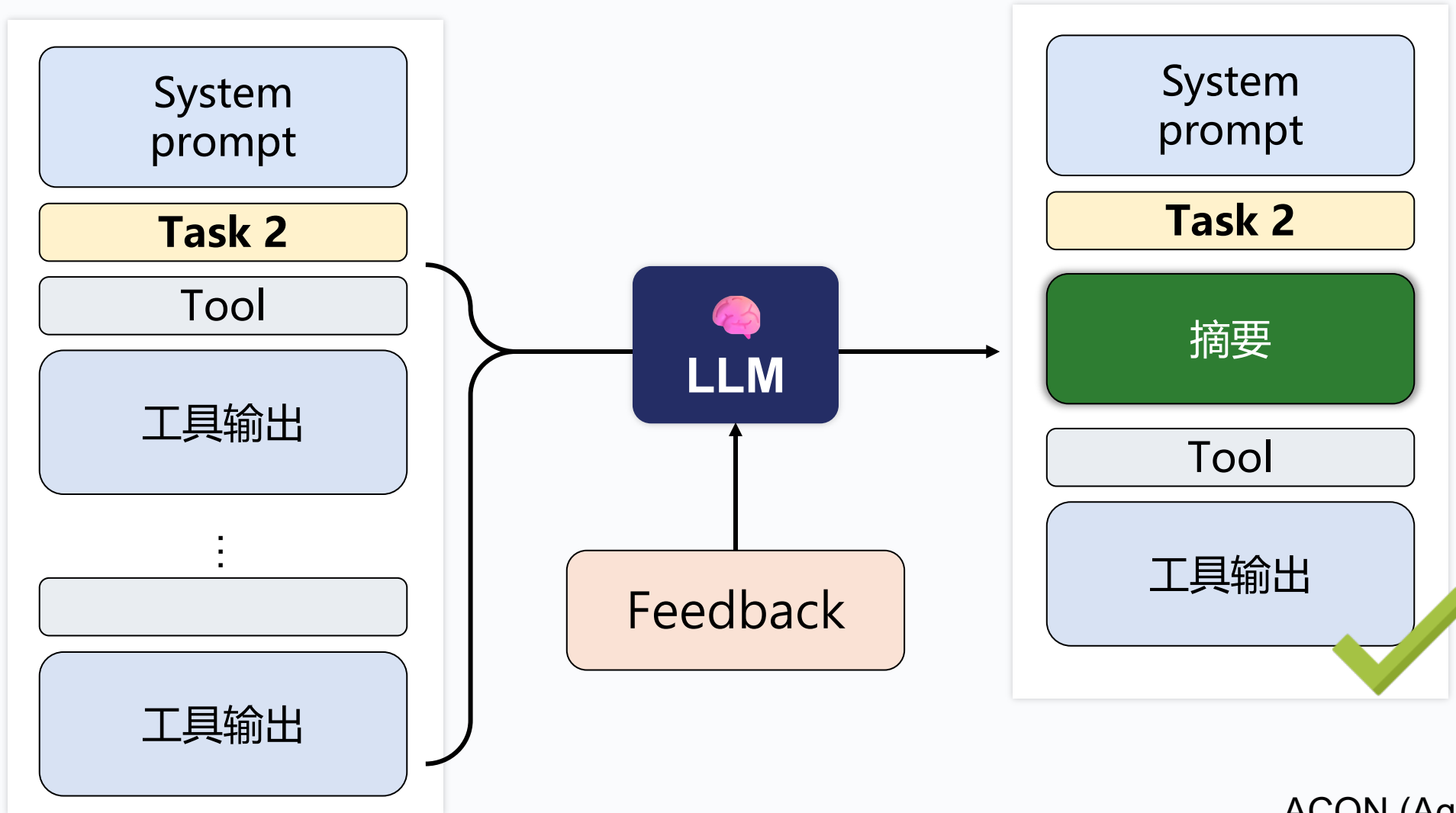
$C_t$ 中可以放入到  
LLM的部分

# 压缩带来的“崩溃”



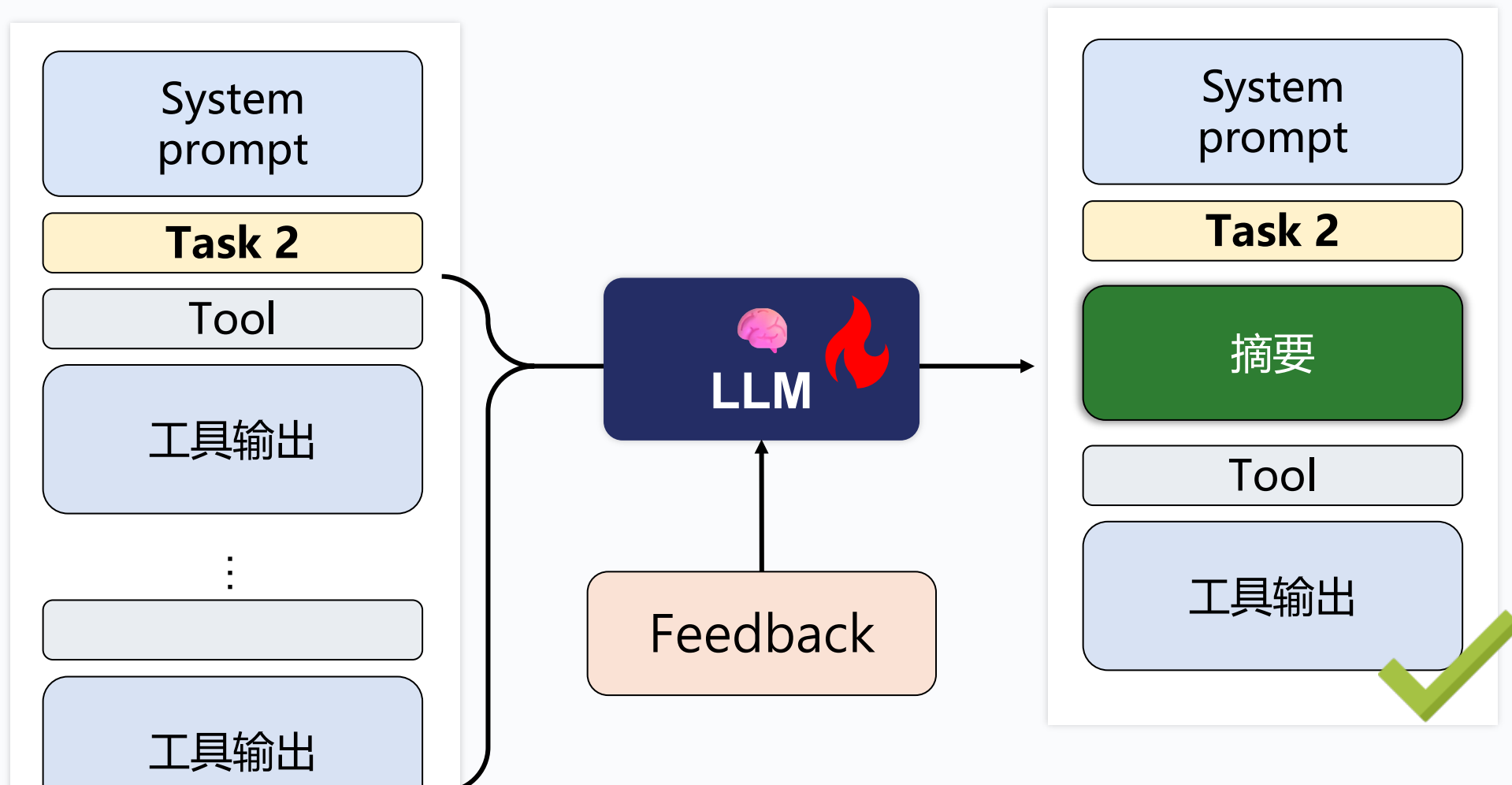
ACON (Agent Context Optimization)  
<https://arxiv.org/abs/2510.00615>

# 压缩带来的“崩溃”



# 专用压缩模型

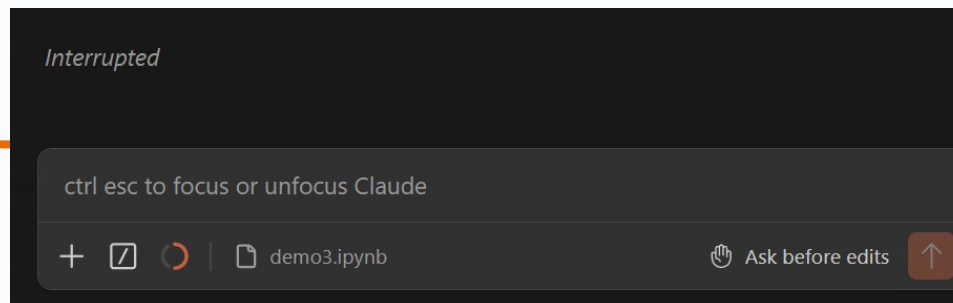
Claude Code 使用了一个专门的压缩模型 J7 来处理上下文压缩。



# 何时压缩?

语言模型不喜欢压缩

需要专门训练LLM  
才能获得主动压缩的能力



A failed example (django\_\_django-13012)

**[System]** ... The user will request for reflection on previous steps by saying "#reflection". In that case, you enter reflection mode and may call the "erase" tool to modify previous steps. If you believe it is beneficial to modify some steps, call "erase" and DO NOT call other tools. Otherwise, DO NOT continue the task and DO NOT call any tool.

**[User]** Problem statement: ...

**[Assistant]** ... **[Tool]** ... **[Assistant]** ... **[Tool]** ... **[Assistant]** ... **[Tool]** ...  
(existing steps in the trajectory)

**[User]** #reflection. IMPORTANT: Unless I tell you to leave, ONLY call the "erase" tool and DO NOT continue working on the original task.

**[Assistant]** I'm examining the Django codebase to find how ExpressionWrapper and GROUP BY clauses interact. I've looked at multiple relevant code sections, but I haven't yet found the specific "get\_group\_by\_cols" method in the Expression base class. Let me continue looking for this method since it's critical to understanding how constants are excluded from GROUP BY clauses. Let me continue by examining more of the Expression class code to find the relevant "get\_group\_by\_cols" method: ... (calling other tools)

# Context Engineering

---

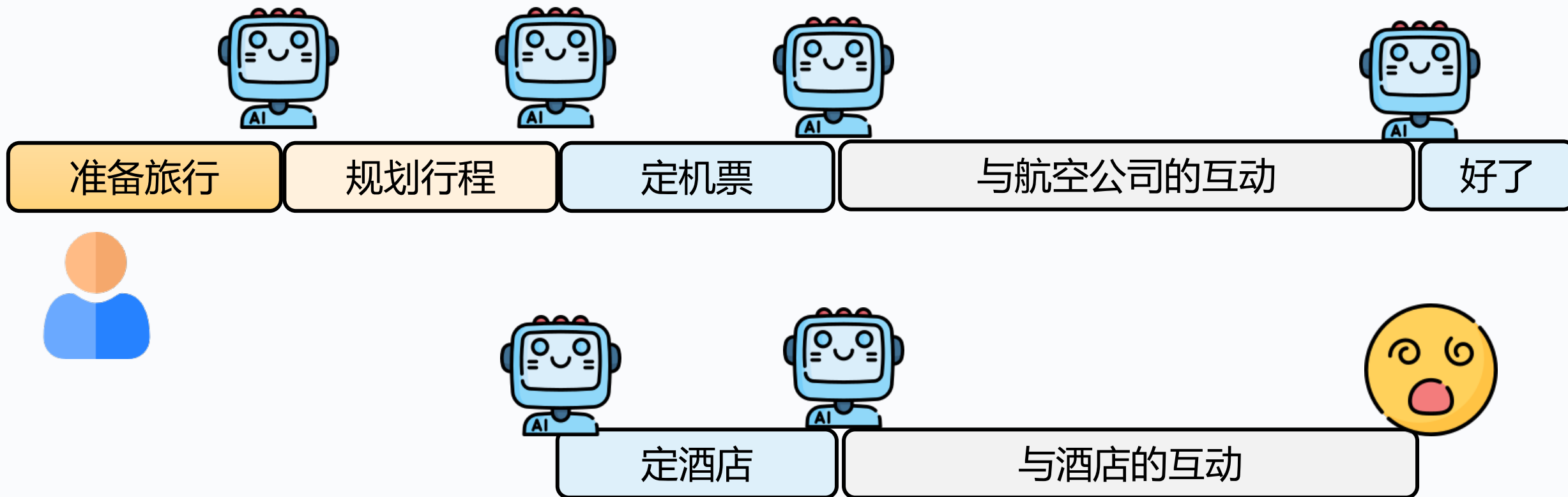
选择

压缩

多智能体

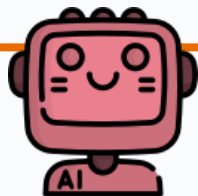
# Multi-Agent

只有一个主Agent的情况

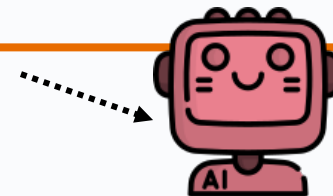


# Multi-Agent

Agent 1



Context中只有定机票

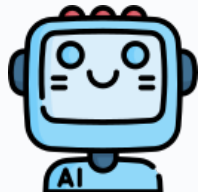
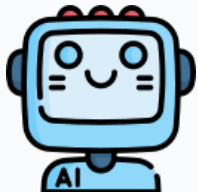


去订机票

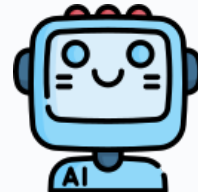
与航空公司的互动

Return: 好了

Leader



Context 中只有  
一个主流程



准备旅行

规划行程

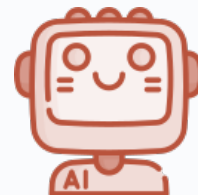
Agent1(去定机票)

好了

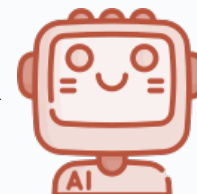
Agent2(去订酒店)

好了

Agent 2



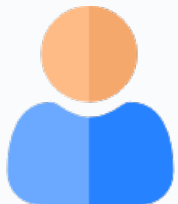
Context中只有订酒店



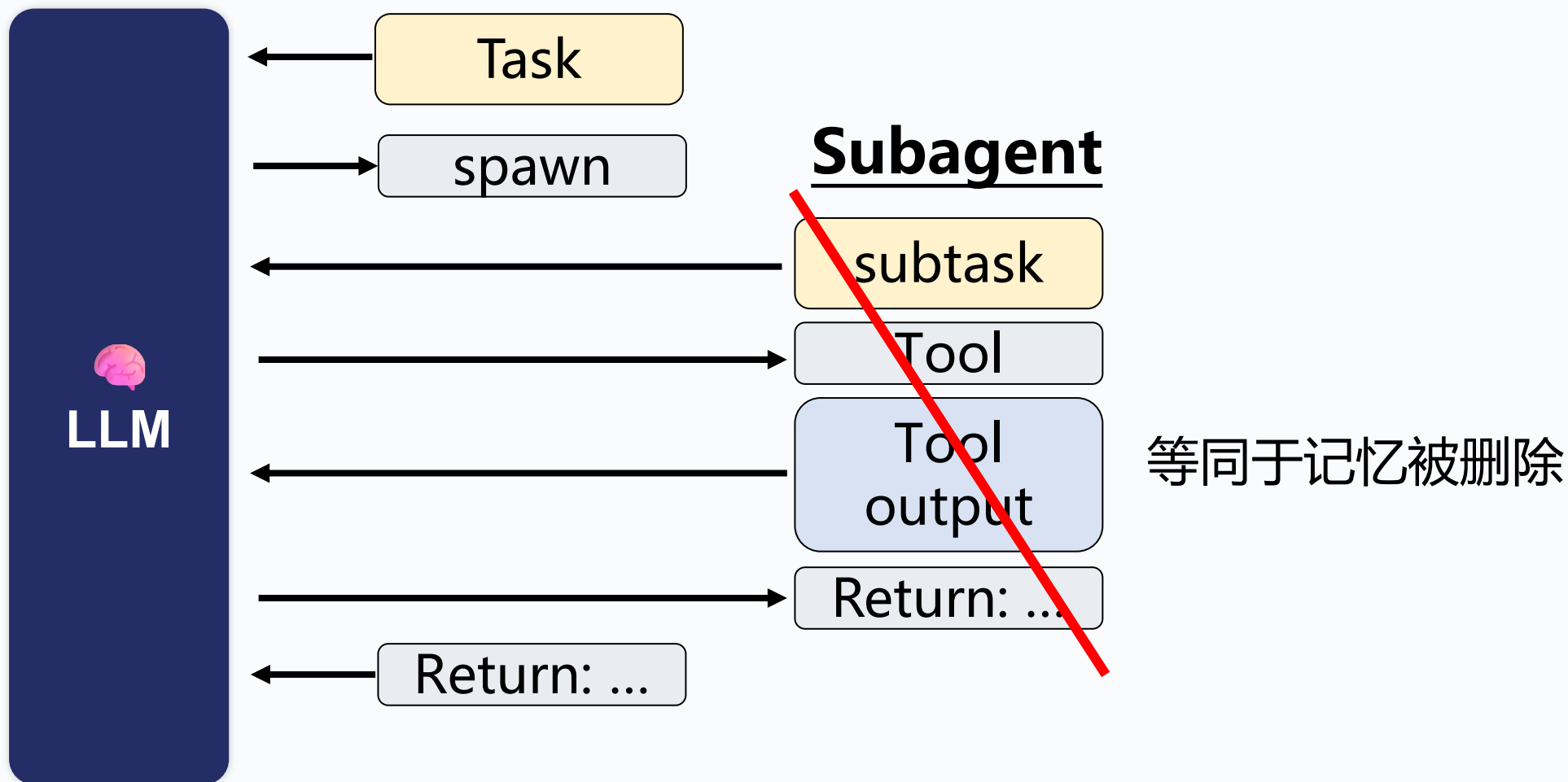
去订酒店

与酒店的互动

Return: 好了

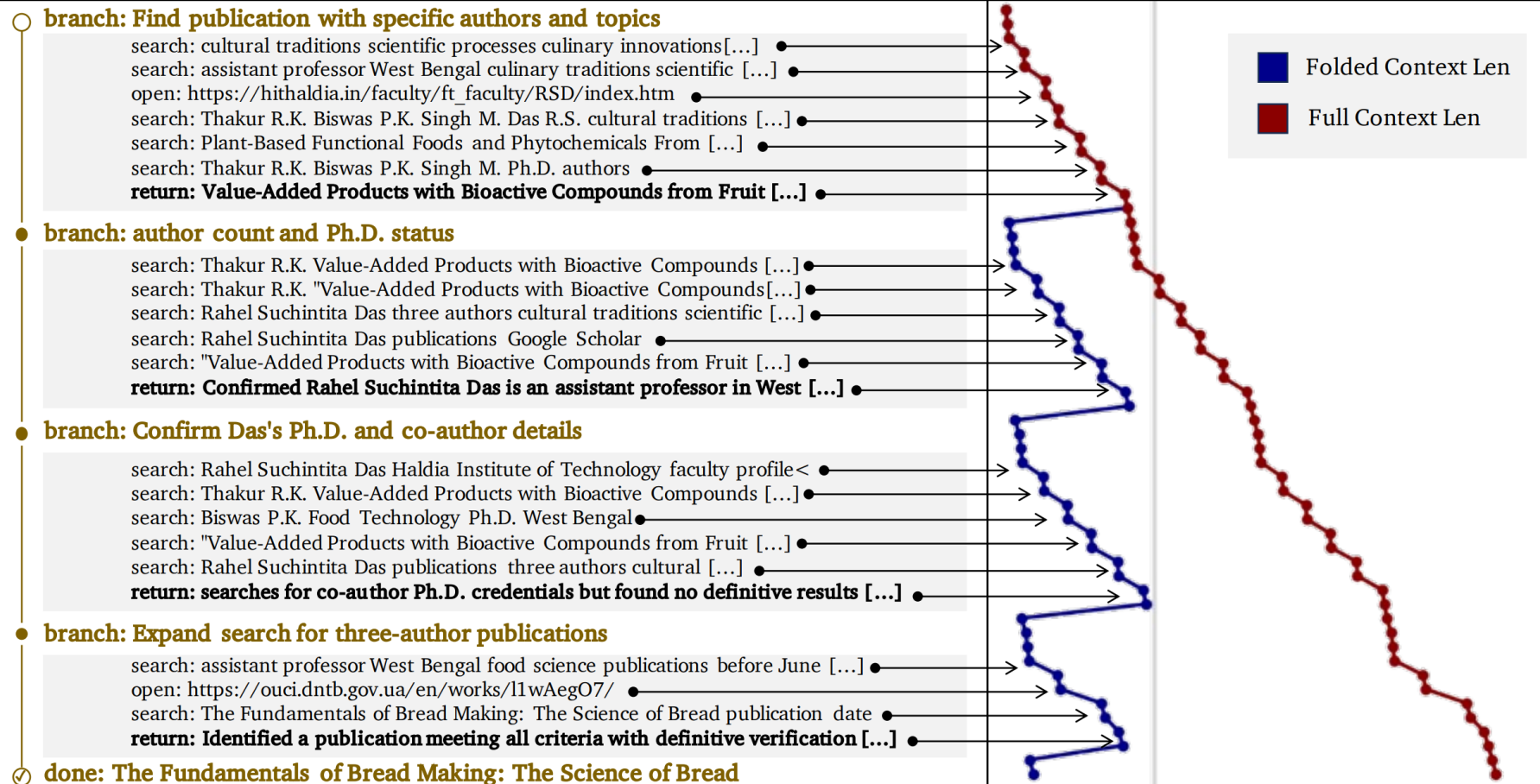


# Subagent 可以看作主动压缩



# Subagent 可以看作主动压缩

Question: Identify the title of a research publication published before June 2023, that mentions Cultural traditions, scientific processes, and culinary innovations. It is co-authored by three individuals: one of them was an assistant professor in West Bengal and another one holds a Ph.D.



(a) Tool Call History

6,524 32,768 107,008

(b) Context Length

# 上下文管理能否也交给Agent?

$I_1 \leftarrow$  initial input

$C_1 \leftarrow$  empty

For  $t = 1$  to  $\infty$

$O_t = LLM(I_t, C_t)$

$C_{t+1} \leftarrow F(C_t, I_t, O_t)$

$I_1 \leftarrow$  initial input

$C_1 \leftarrow$  empty

For  $t = 1$  to  $\infty$

$O_t = LLM(I_t, P_t)$

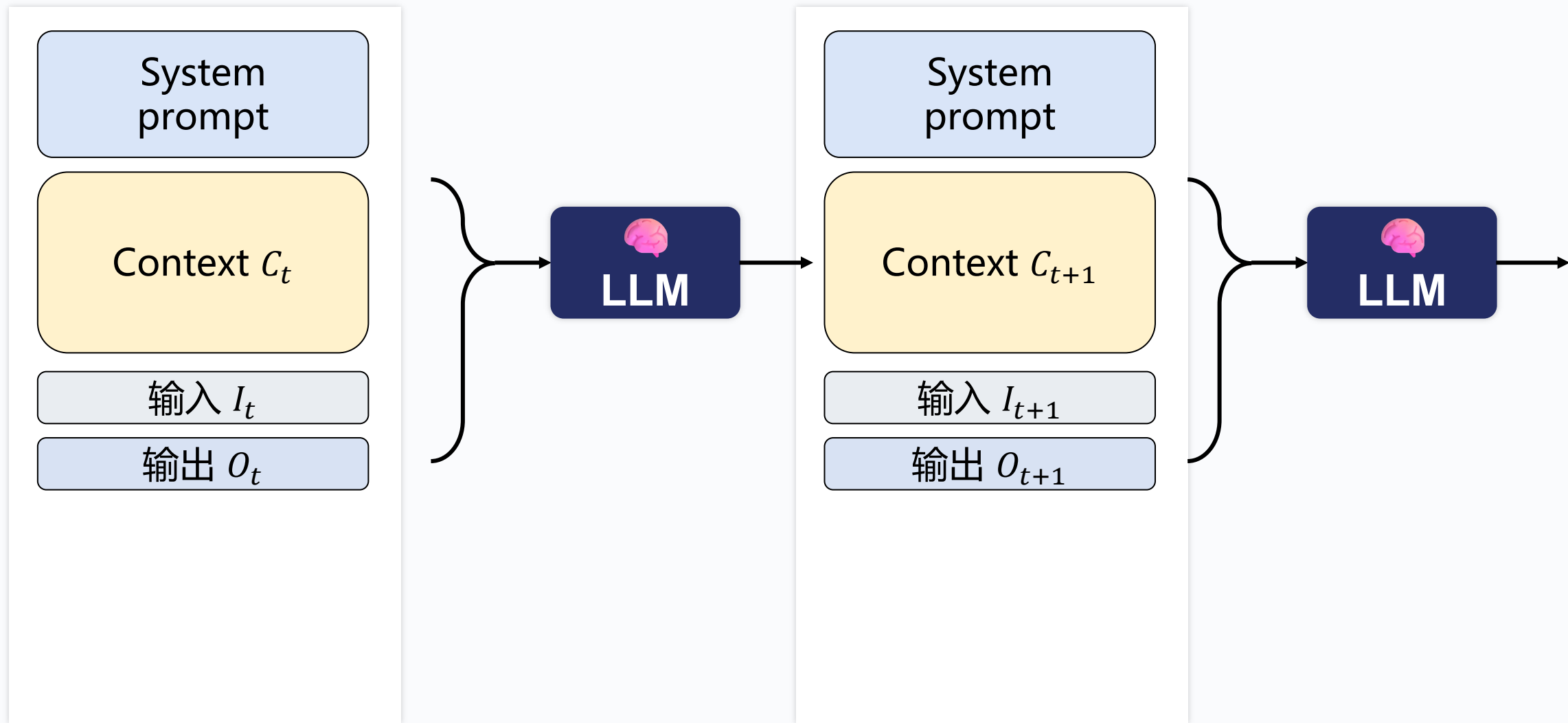
$C_{t+1} \leftarrow F(C_t, I_t, O_t)$

↑  
一切交给LLM

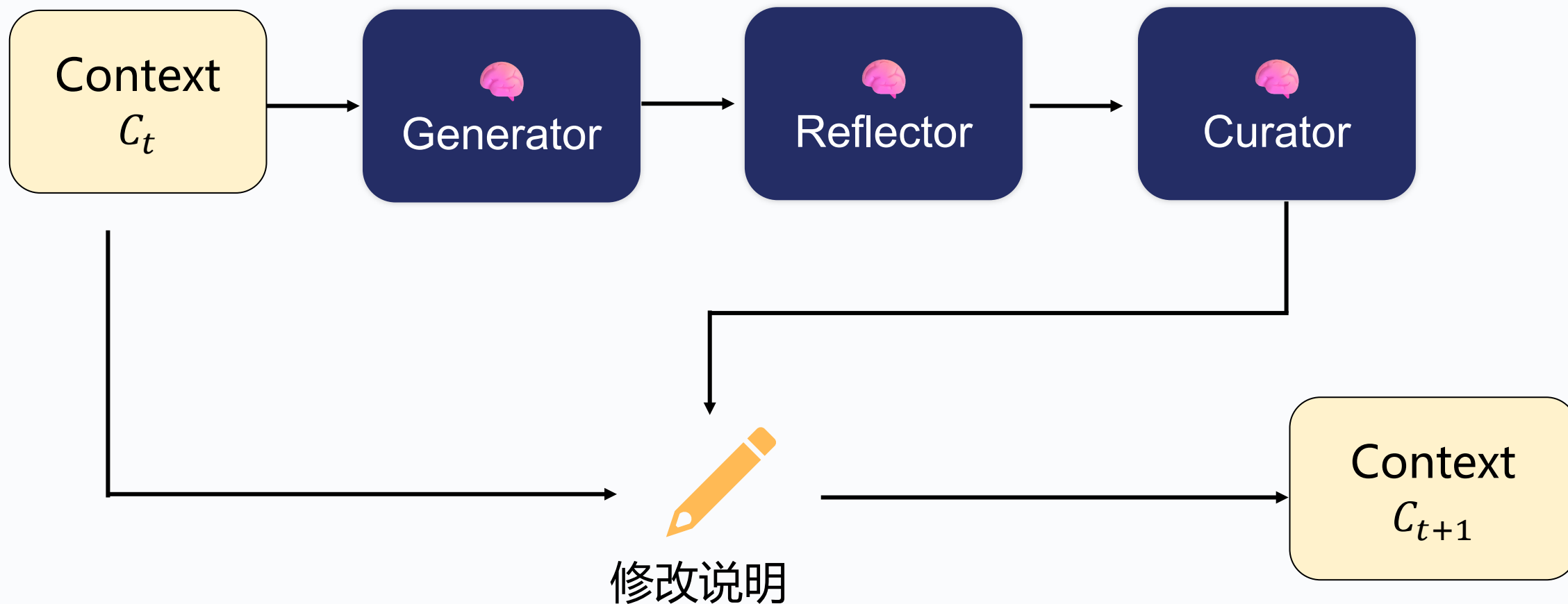
$C = \{P, M\}$

↙  $C_t$ 中可以放入到  
LLM的部分

# 自动上下文管理 (Agentic Context Engineering)



# 自动上下文管理 (Agentic Context Engineering)



# 自动上下文管理 (Agentic Context Engineering)

