Market Research and Analysis

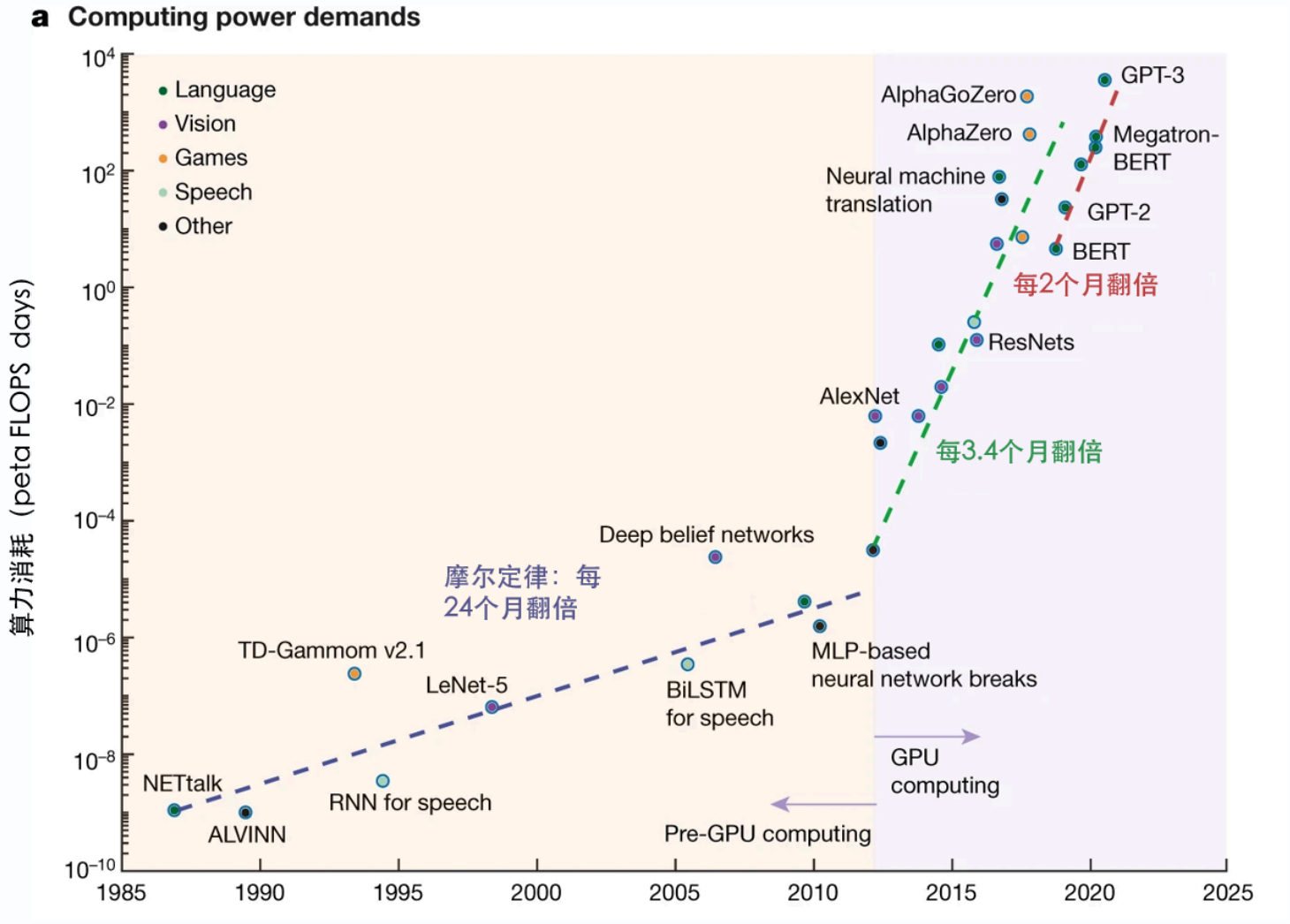
Lecture 6: AI Agents for Economic Research

- Zhenyu Zhao
- Nankai Institute of International Economics
- Feishu: 2120253538
- Email: zzynankai@outlook.com
- Website: xishanyu2.github.io

01

算力驱动的未来

顺应数字技术和人工智能发展大势，《中华人民共和国国民经济和社会发展第十五个五年规划纲要》将提升数智化发展水平单独成篇，聚焦算力、算法、数据高效供给和数智技术赋能经济社会发展，为未来5年数字中国建设划定清晰路线。

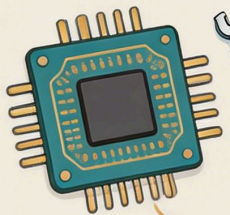

a Computing power demands
算力消耗（peta FLOPS days）
Language
Vision
Games
Speech
Other
GPT-3
AlphaGoZero
AlphaZero
Megatron-BERT
Neural machine translation
GPT-2
BERT
每2个月翻倍
ResNets
AlexNet
每3.4个月翻倍
Deep belief networks
摩尔定律：每24个月翻倍
TD-Gammom v2.1
LeNet-5
BiLSTM for speech
MLP-based neural network breaks
NETtalk
RNN for speech
ALVINN
GPU computing
Pre-GPU computing
10^{4}
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
1985
1990
1995
2000
2005
2010
2015
2020
2025

Moore's Law is the famous prognostication by Intel co-founder Gordon Moore that the number of transistors on a microchip would double every year or two. This prediction has mostly been met or exceeded since the 1970s — computing power doubles about every two years, while better and faster microchips become less expensive.

This rapid growth in computing power has fueled innovation for decades, yet in the early 21st century researchers began to sound alarm bells that Moore's Law was slowing down. With standard silicon technology, there are physical limits to how small transistors can get and how many can be squeezed onto an affordable microchip.

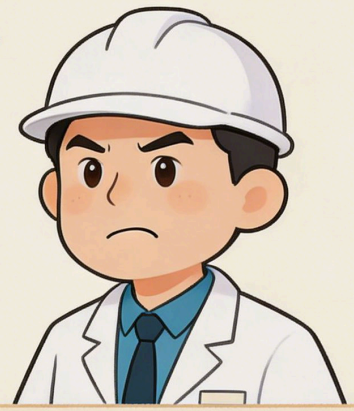
告别摩尔定律 → 华为韬定律

摩尔定律（旧）

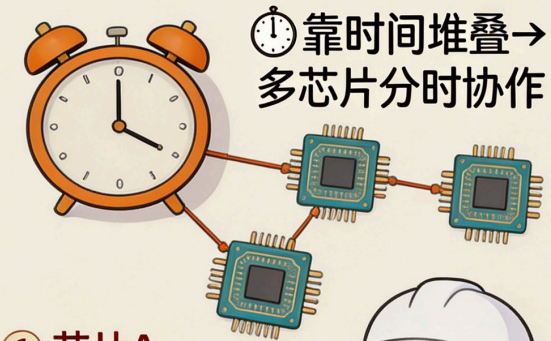


靠缩小总体管→
物理极限到顶

性能提升
越来越难

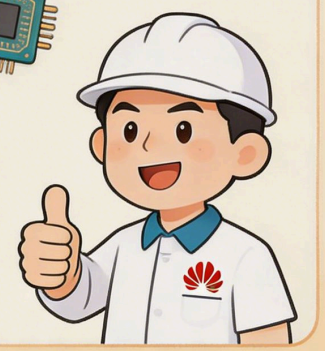


韬定律（新）



靠时间堆叠→
多芯片分时协作

① 芯片A
处理任务1
② 芯片B
处理任务2
③ 芯片C
处理任务3



时间换性能！突破物理极限

2026年5月25日，在电气电子工程师学会（IEEE）举办的国际电路系统研讨会ISCAS 2026上，华为何庭波发表题为“半导体新路径探索与实践”的主旨演讲，发表了指导半导体产业发展的新原则——韬(τ)定律。韬(τ)定律提出以“时间(τ)缩微”替代“几何缩微”作为半导体与电子系统演进的新指导原则——通过逻辑折叠等创新技术，持续压缩信号传播时延，不断提升晶体管密度，从而实现半导体与电子系统的持续演进。

「一张图看懂摩尔定律与韬定律」

「核心思想、技术路径与产业意义对比」

「摩尔定律：几何缩微（做小）」 ------ 「两条路线，解决不同增长问题」 ------ 「韬定律：时间缩微（压快）」

「维度」	「摩尔定律」	「韬定律」
1.「提出时间」	「1965年」	「2026年」
2.「提出者」	「戈登·摩尔（美国）」	「何庭波（中国）」
3.「核心思路」	「几何缩微（做小）」	「时间缩微（压快）」
4.「优化目标」	「晶体管密度」	「信号传播时间常数 τ」
5.「技术手段」	「光刻工艺进步」	「逻辑折叠 + 3D堆叠 + 先进封装」
6.「迭代节奏」	「统一节奏：2年/代」	「场景驱动：1.3x~10x/年」
7.「物理瓶颈」	「7nm以下严重收窄」	「未到理论极限」
8.「工具链」	「EDA成熟完善」	「需全新EDA工具链」
9.「成本趋势」	「先进制程成本飙升」	「成熟制程 + 立体设计」
10.「产业基础」	「依赖尖端光刻设备」	「依赖先进封装能力」
11.「中国企业影响」	「受制于设备限制」	「绕开光刻瓶颈」

「核心启示」

- 「摩尔定律偏向制程驱动」
- 「韬定律偏向架构驱动」
- 「前者依赖光刻，后者依赖封装」
- 「对中国企业具有重要战略意义」

「小白怎么理解」

「摩尔定律」

「核心是把晶体管做得更小、更密，让芯片越来越强」

VS

「韬定律」

「核心是不必一味缩小，而是通过架构、堆叠和封装把速度压快」

「一个主要追求做小，一个更强调跑快」

一张图看懂普及摩尔定律和韬定律两条定律的全面对比和小白解释

核心思想与技术路径

关键区别
- 摩尔定律关注硬件性能增长
- 韬定律关注系统智能增长
- 一个偏底层算力，一个偏上层能力

摩尔定律：靠芯片变强　　**韬定律：靠系统变聪明**

摩尔定律

小白理解：让芯片上的晶体管越来越多，算力越来越强

1 核心思想

通过提升单位芯片上的晶体管数量，持续推动计算性能提升与成本下降

2 技术路径

制程缩小 → 晶体管密度提升 → 芯片性能增强 → 单位算力成本下降

3 典型特征

- 摩件驱动
- 依赖半导体工艺进步
- 增长来自底层器件迭代

摩尔定律	比较维度	韬定律
芯片算力	提升对象	任务智能
半导体工艺	主要抓手	模型+数据+工具+系统
器件密度提升	增长来源	协同优化与流程放大
更快、更便宜的计算	结果表现	更强、更完整的任务完成能力

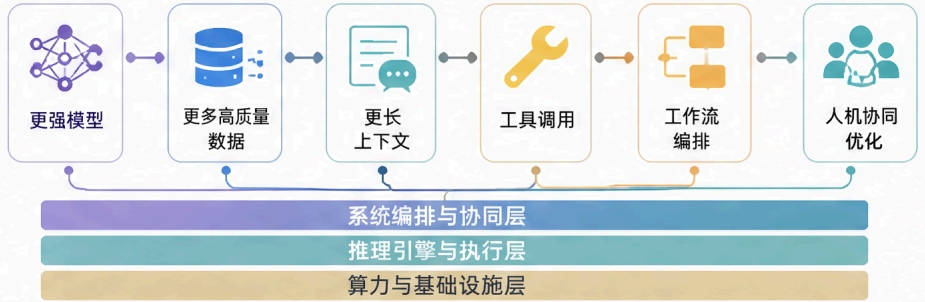
韬定律

小白理解：不只是换更强芯片，而是让模型、数据、系统和工具一起变强

1 核心思想

通过模型、数据、工具、推理、工作流与系统协同优化，
持续放大智能产出与任务完成能力

2 技术路径



3 典型特征

- 系统驱动
- 依赖软件与协同能力提升
- 增长来自整体智能系统迭代

小白版一句话

摩尔定律
核心是把机器本身做得更强

韬定律
核心是让整套智能系统更会做事

★ 前者更像升级发动机，后者更像升级整辆车和驾驶系统 ★

02

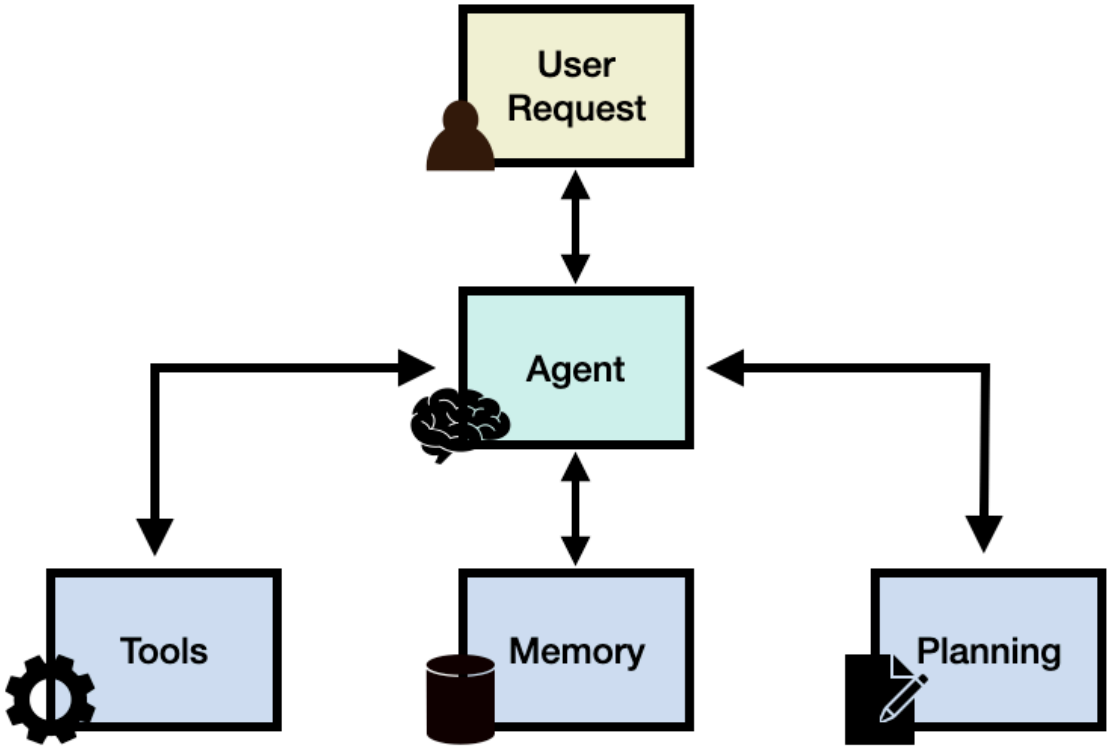
AI Agent

> AI is no longer only a writing assistant. It is becoming a research partner that extends your second brain.
> AI Won’t Replace Humans — But Humans With AI Will Replace Humans Without AI.

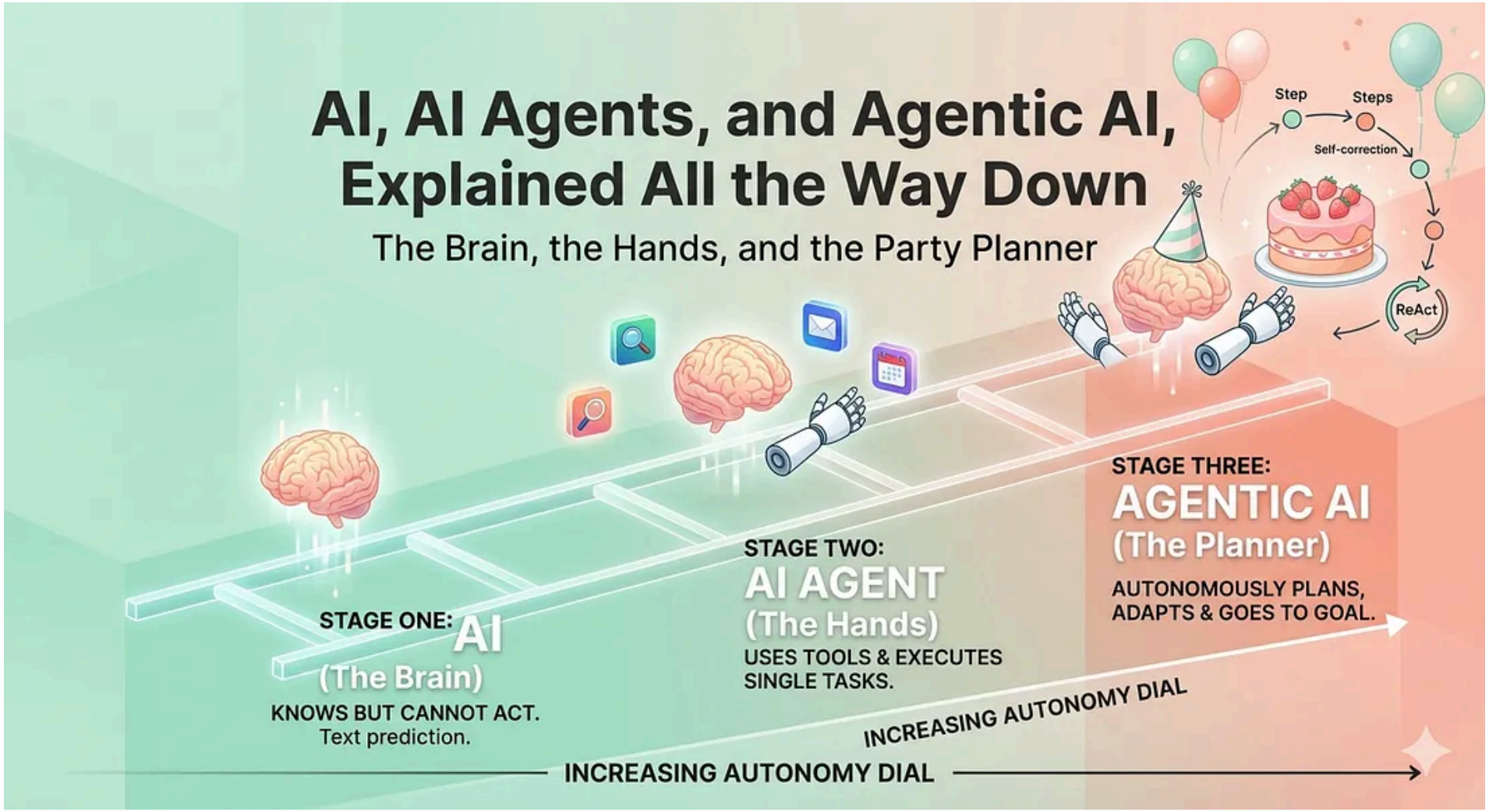
Agent 就是一个能干活的智能助手。

Agent = LLM (大脑) + Planning (规划) + Tool use (执行) + Memory (记忆)

学习 Agent 需要思维转变： 从 `对话框问答` 进化为 `目标驱动的任务执行` 。



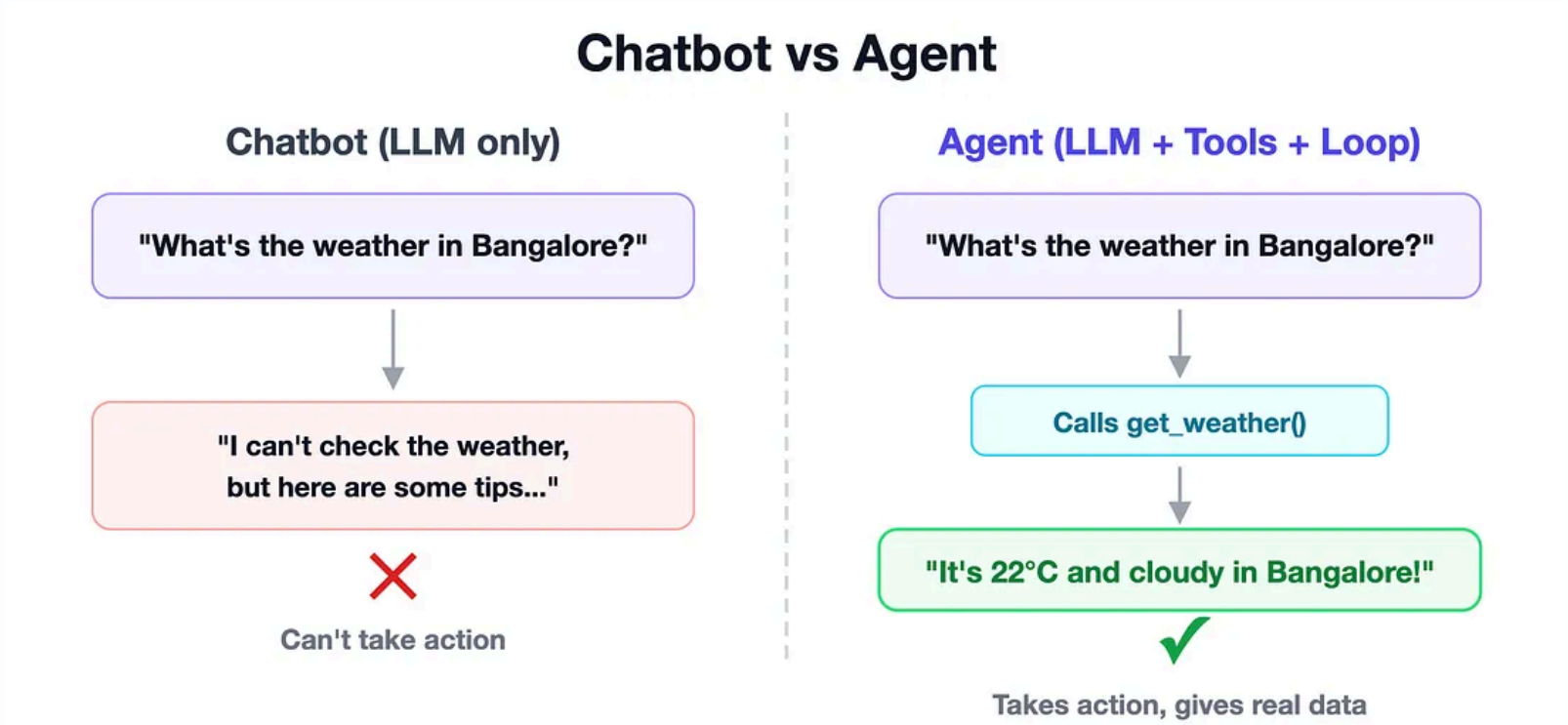
它就像是一位**拥有“数字大脑”和“手脚”的私人助理**，不仅能回答问题，还能替你接管复杂的任务并执行操作。



AI 智能体 vs. 传统 AI（聊天机器人）

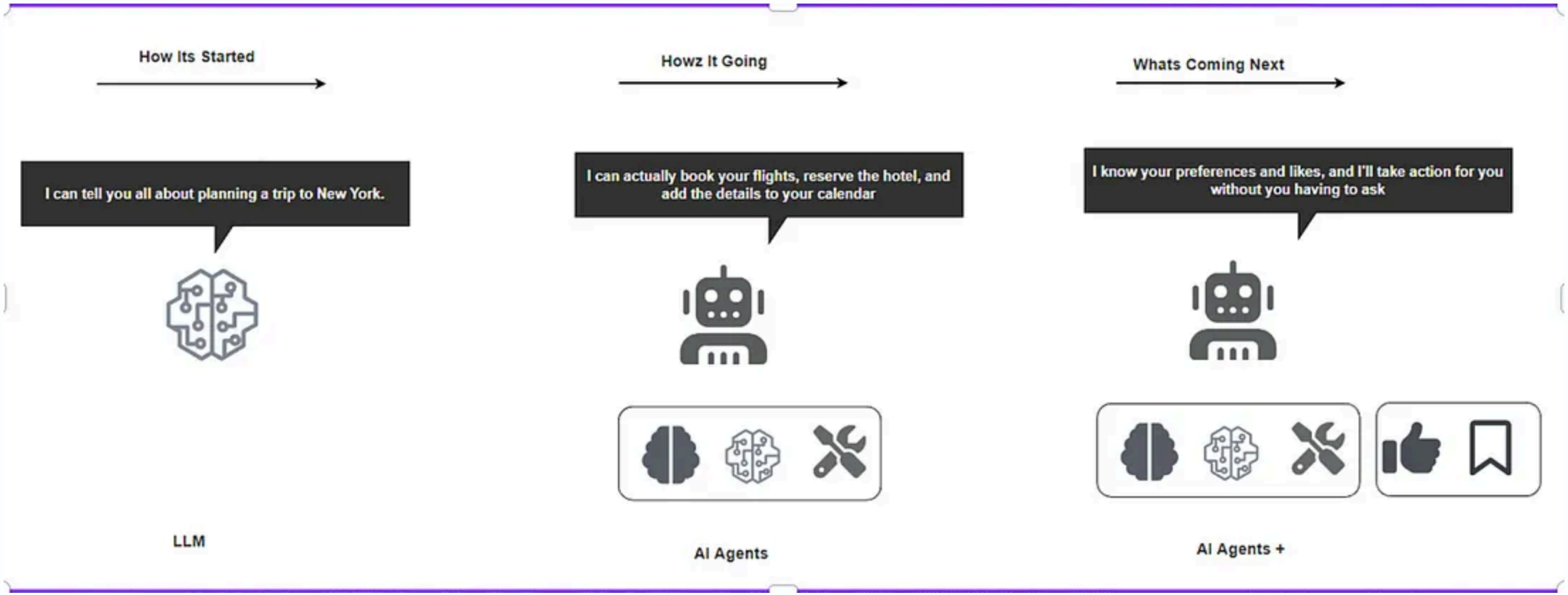
传统的聊天机器人如 ChatGPT 就像简单的客服只能**被动响应**你的输入，需要人类给出详细的每步指示；而 AI Agent 是**目标驱动**的，一旦你设定了目标，它能自主出击，独立完成多步骤操作。例如：

- 普通聊天机器人：你问“北京天气如何？”，它会回答“今天北京下雨”。
- AI Agent：你下达指令“帮我查一下下周去北京的机票和酒店，并在预算内完成预订”。Agent 会自主查询多个网站（工具）、比价（规划）、最终生成订单（行动），全程无需人类干预。



AI Agent 更像一个有自主性的员工，它能够：

- **理解任务目标**：明白你想要什么结果
- **制定计划**：思考如何达成目标
- **使用工具**：调用各种资源和API
- **自我调整**：根据反馈优化策略
- **持续执行**：直到完成任务或遇到无法解决的问题



一个完整的 AI Agent 通常包含以下四个核心模块：

1. **大脑（大语言模型LLM）**：负责理解意图、逻辑推理和生成决策。
2. **规划（Planning）**：能将复杂的大目标拆解成一步步可执行的小任务，并在遇到阻碍时自我反思与调整。
3. **记忆（Memory）**：记录过去的交互历史（短期记忆）和存储专业知识库（长期记忆）。
4. **工具使用（Tool Use）**：调用各种外部工具（如搜索引擎、数据库、计算器、API 等）来完成具体动作。

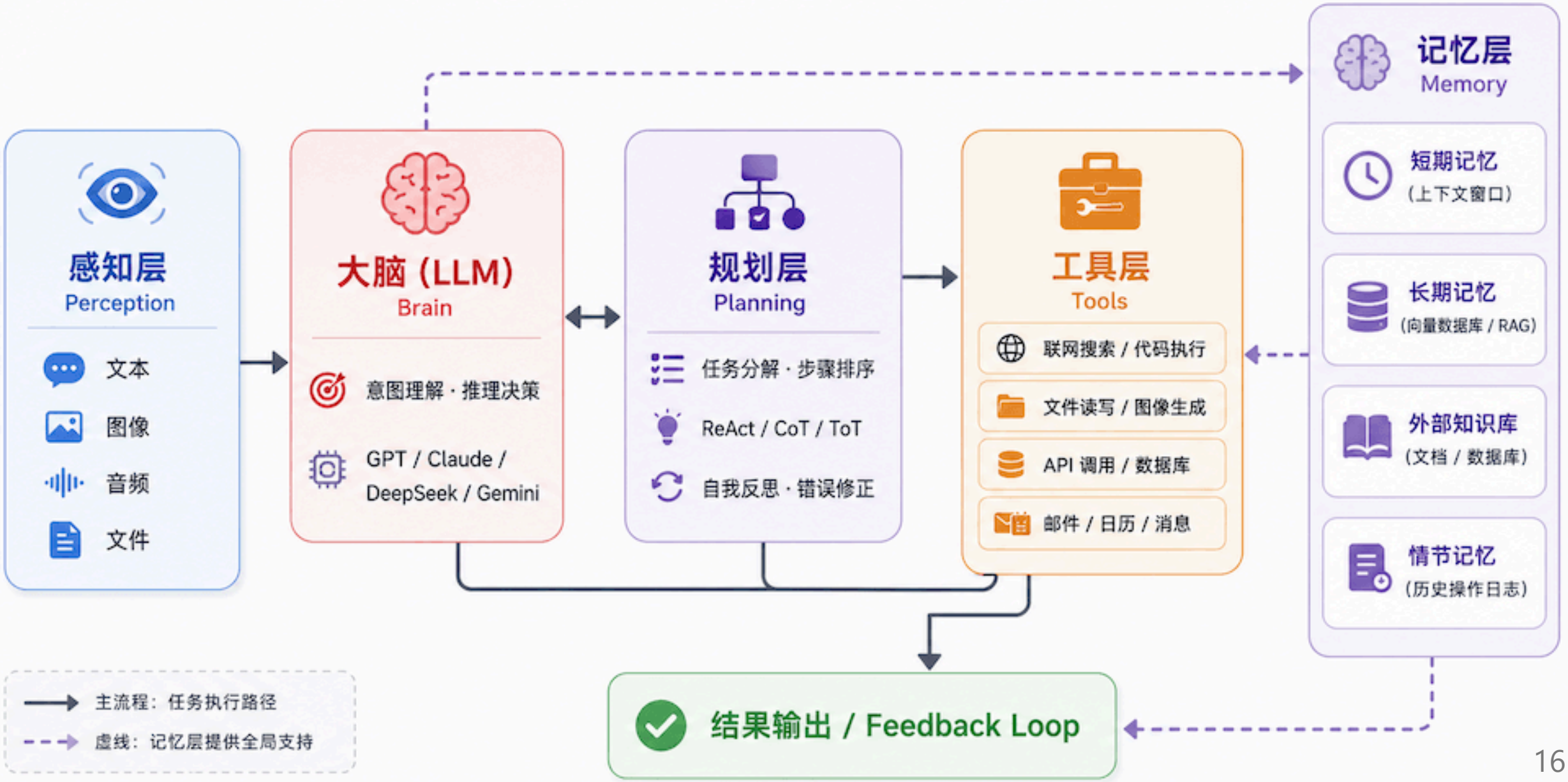
与普通大模型的差异点:

- 普通大模型：生成文本、图片等
- Agent：生成行动并执行行动，能完成实际工作

Agent 与传统 AI 模型的区别

维度	传统 AI 模型	AI Agent
交互方式	单次输入输出	多轮对话、持续交互
决策能力	基于输入直接推理	规划、反思、迭代优化
工具使用	无法主动调用外部工具	可调用搜索、计算器、API 等
记忆机制	仅限当前上下文	短期+长期记忆
目标导向	完成单一预测任务	完成复杂目标
错误处理	输出即结束	可自我纠错、重试

AI Agent 五大核心组件



0、感知层 (Perception) —— 餐厅的前台

角色：负责接待顾客，理解来自外部世界的所有输入。

Agent 在行动之前，必须先"看到"和"听到"外部信息。现代 Agent 已经不限于纯文本输入，而是具备**多模态感知**能力：

- **文本输入**：用户的自然语言指令、文档内容、代码。
- **图像 / 视频**：截图、图表，Agent 可以直接"看图"理解。
- **结构化数据**：表格、JSON、数据库查询结果。
- **环境状态**：在计算机操作类 Agent 中，当前屏幕状态、网页结构等。
- **工具返回结果**：上一步工具调用的输出，会作为新的感知输入进入下一轮循环。

感知层的输入经过整合，形成 Agent 的"当前上下文"，送入大脑进行理解和决策。

1、大脑 (Brain) —— 也就是大模型

角色：餐厅的**主厨兼经理**。

这是 Agent 最核心的部分（比如 GPT-4、Claude、DeepSeek、通义千问）。

- 它负责**听懂**你想吃什么（理解意图）。
- 它负责**指挥**其他人干活（决策）。
- 如果没有它，整个餐厅就瘫痪了。

大脑做的三件核心事

意图理解	解析用户输入，明确目标是什么	听懂顾客点了什么
推理决策	综合上下文和记忆，判断下一步该做什么	主厨决定先处理哪道菜
工具调用判断	判断是否需要调用外部工具，选择哪个工具、传入什么参数	决定用哪口锅、让谁去买食材

大脑的"智力天花板"决定了整个 Agent 的上限。同一套工具和规划框架，接入能力更强的基础模型，任务完成质量往往有质的飞跃。

2、工具 (Tools) —— 厨房里的设备

角色：厨具和帮手。

光有主厨（大脑）是不够的，还得有锅碗瓢盆才能做菜。对于 AI Agent 来说，工具就是能把决策转化为真实动作的执行单元。

工具可以按照用途分为四大类：

类别	常见工具	作用
信息获取	联网搜索、网页抓取、文档读取、数据库查询	获取 Agent 自身知识之外的实时或专业信息
计算执行	代码解释器、数学计算引擎、沙箱环境	处理需要精确计算或程序逻辑的任务
内容生成	图像生成、语音合成、文档导出	产出非文本形式的内容
系统交互	API 接口、邮件、日历、文件操作、消息发送	与外部系统、服务和真实世界进行交互

常见工具举例：**联网搜索** 信息获取像去菜市场买新鲜食材，**代码解释器** 计算执行像精密的烤箱，处理复杂计算，**画图工具** 内容生成像摆盘师，负责美观，**API 接口** 系统交互像外卖小哥，连接外部世界。

3、记忆 (Memory) —— 顾客记录本

> **角色：服务员的记性。**
>
> 你肯定不喜欢每次去餐厅都要重新报一遍：我不吃香菜！

Agent 的记忆分为以下几种类型：

- **短期记忆（In-Context Memory）**：即当前对话的上下文窗口。记住刚才你说了啥（比如你刚点了鱼，下一句说"要微辣"，它知道是指鱼）。受限于模型的上下文长度，通常在 8K 到 200K token 之间。
- **长期记忆（External Memory）**：记住你的长期偏好（比如你是素食主义者，或者你的家庭住址）。通常通过向量数据库实现持久化存储。
- **情节记忆（Episodic Memory）**：对历史任务执行过程的记录，包括"上次遇到这种情况我是怎么处理的"，帮助 Agent 从过去的经验中学习。
- **语义记忆（Semantic Memory）**：抽象的知识和事实，通常来自预训练阶段已经内化的内容，也可通过 RAG（检索增强生成）动态补充。

4、规划 (Planning) —— 烹饪流程单

> **角色：后厨的出餐 SOP**（Standard Operating Procedure，标准作业程序）。
>
> 当你点了一份佛跳墙，主厨不会乱做，而是会在脑子里生成一个清单：
>
> 1. 先备料（鲍鱼、海参...）
> 2. 再熬汤
> 3. 最后慢炖

Agent 也是一样。当你给它一个复杂任务（比如"写一份竞品分析报告"），它会自己拆解：

- 第一步：去搜集竞品 A、B、C 的资料。
- 第二步：对比它们的价格和功能。
- 第三步：把对比结果写成文章。
- 第四步：检查一遍有没有错别字。

5、Agent 运行循环 (Agent Loop)

以上各组件并非孤立存在，它们组成一个持续迭代的**感知—思考—行动—观察**闭环，这就是"Agent Loop"。Agent 不断重复这个循环，直到任务完成或达到终止条件。

感知（接收输入/环境状态）→ 思考（LLM 推理/规划分解）→ 行动（调用工具/执行操作）→ 观察（获取结果/更新记忆）→ 达到终止条件（任务完成）/继续循环（任务未完成）

这个循环让 Agent 具备了**在失败时自我纠错**的能力：如果某一步工具调用返回了错误或意外结果，"观察"阶段会将这个信息反馈给大脑，大脑在下一轮"思考"时就会调整策略。

普通的 LLM 只是 One-shot（一次性） 的响应，而 Agent 的核心在于 Iterative（迭代）。
ReAct 模式 (Reason + Act) 是目前最主流的 Agent 推理逻辑：

核心推理逻辑：从 Prompt 到 Reasoning Loop

普通的 One-shot 响应 vs Agent 的 Iterative 迭代

One-shot (普通 LLM)

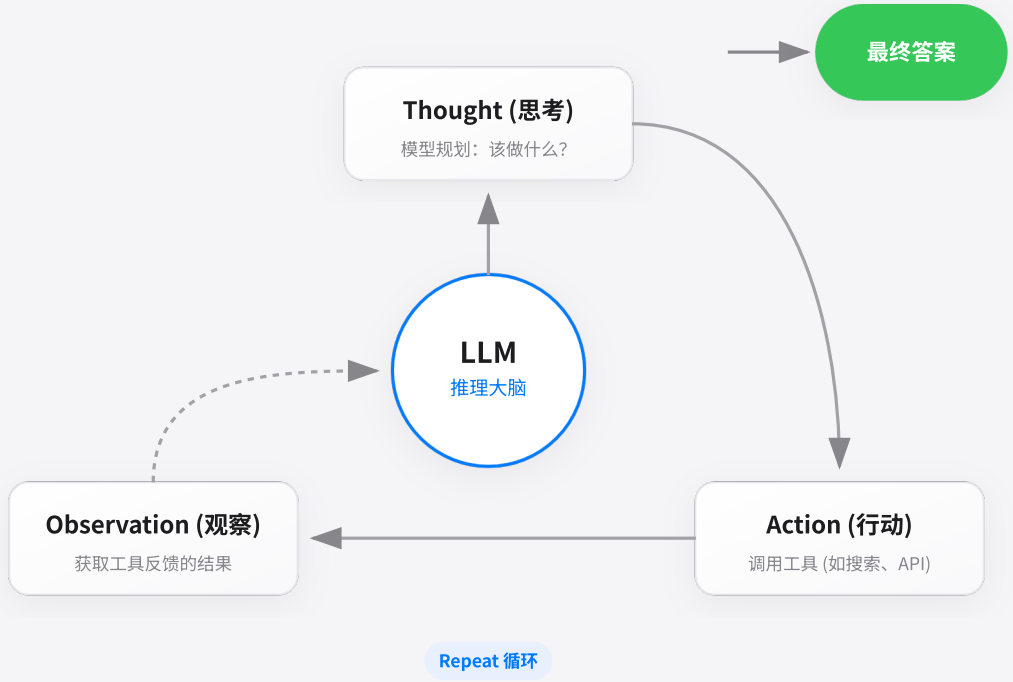
Input → [LLM] → Output

单次响应，不具备纠错能力

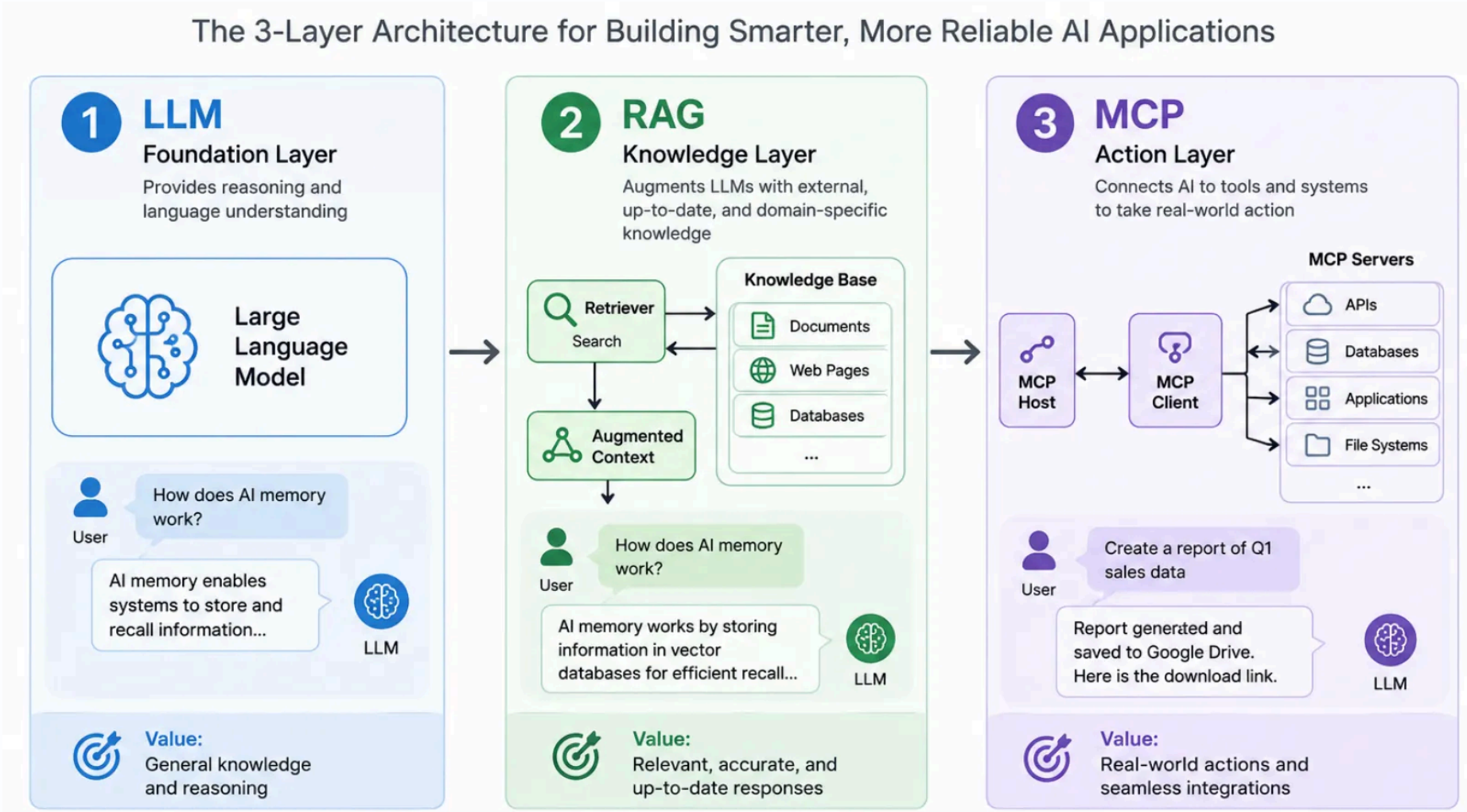
Iterative (Agent 模式)

ReAct: Reason + Act

持续思考与工具交互直到完成



从基础模型到智能体的完整AI系统架构


The 3-Layer Architecture for Building Smarter, More Reliable AI Applications
1
LLM
Foundation Layer
Provides reasoning and language understanding
Large Language Model
User
How does AI memory work?
AI memory enables systems to store and recall information...
LLM
Value:
General knowledge and reasoning
2
RAG
Knowledge Layer
Augments LLMs with external, up-to-date, and domain-specific knowledge
Retriever
Search
Augmented Context
Knowledge Base
Documents
Web Pages
Databases
...
User
How does AI memory work?
AI memory works by storing information in vector databases for efficient recall...
LLM
Value:
Relevant, accurate, and up-to-date responses
3
MCP
Action Layer
Connects AI to tools and systems to take real-world action
MCP Servers
MCP Host
MCP Client
APIs
Databases
Applications
File Systems
...
User
Create a report of Q1 sales data
Report generated and saved to Google Drive. Here is the download link.
LLM
Value:
Real-world actions and seamless integrations

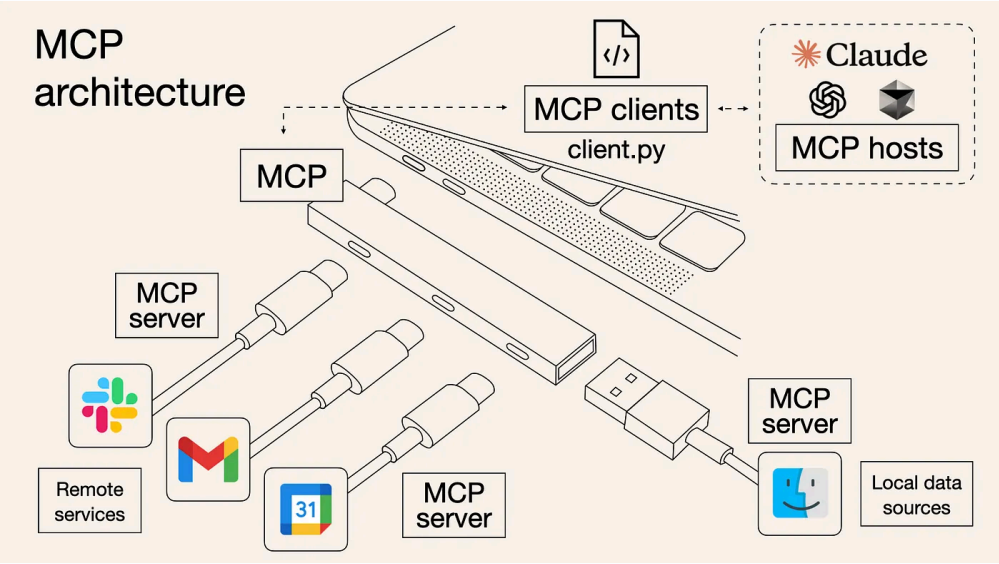
What is the Model Context Protocol (MCP)?

MCP (Model Context Protocol) is an open-source standard for connecting AI applications to external systems.

Using MCP, AI applications like Claude or ChatGPT can connect to data sources (e.g. local files, databases), tools (e.g. search engines, calculators) and workflows (e.g. specialized prompts)—enabling them to access key information and perform tasks.

Think of MCP like a USB-C port for AI applications. Just as USB-C provides a standardized way to connect electronic devices, MCP provides a standardized way to connect AI applications to external systems.

`Zotero MCP`, `Stata MCP`



What are Agent Skills?

Agent Skills are a lightweight, open format for extending AI agent capabilities with specialized knowledge and workflows.

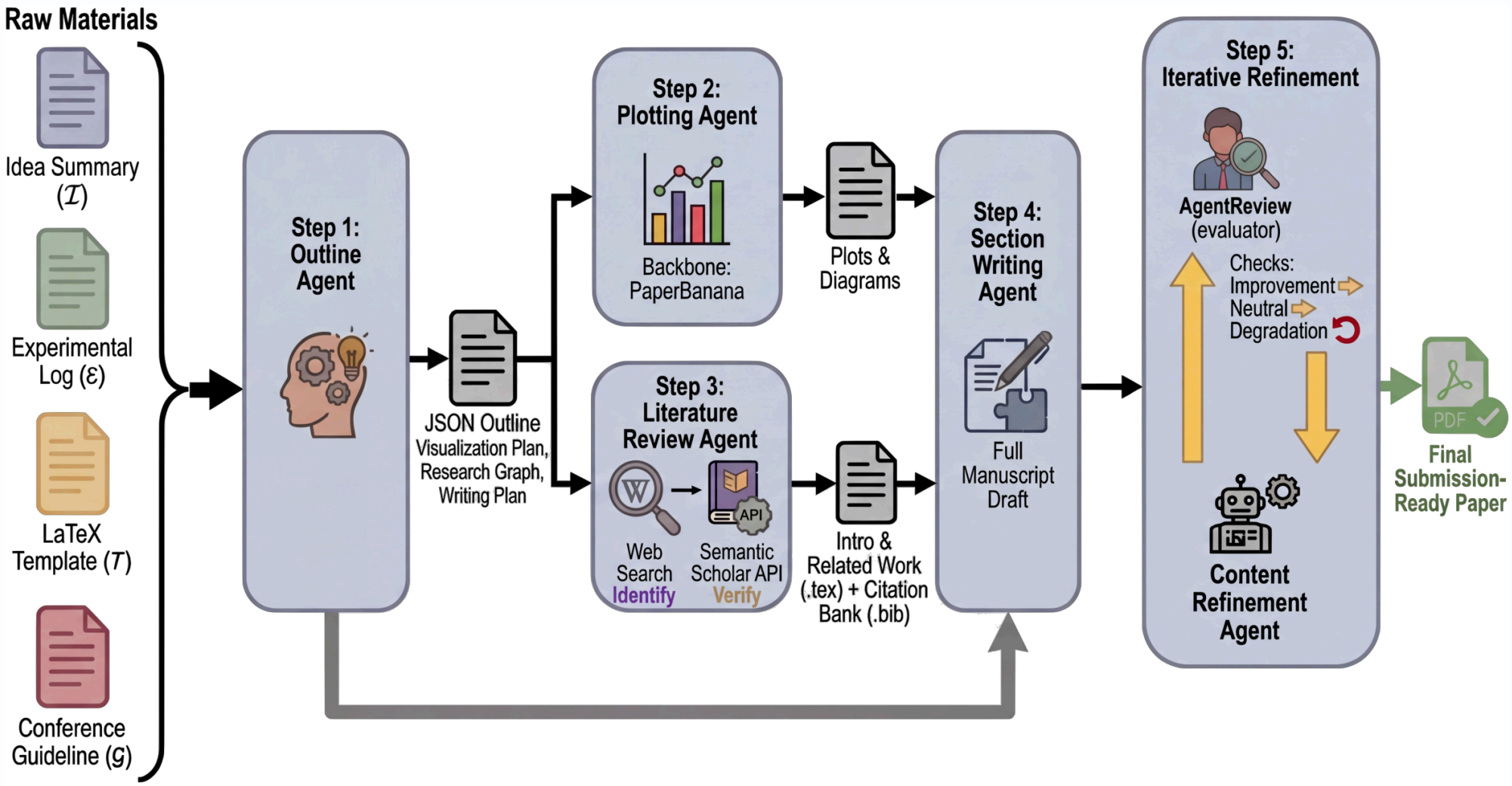
At its core, a skill is a folder containing a `SKILL.md` file. This file includes metadata (`name` and `description`, at minimum) and instructions that tell an agent how to perform a specific task. Skills can also bundle scripts, reference materials, templates, and other resources.

```
my-skill/
├── SKILL.md          # Required: metadata + instructions
├── scripts/          # Optional: executable code
├── references/       # Optional: documentation
├── assets/           # Optional: templates, resources
└── ...               # Any additional files or directories
```

Subagent and Multi-Agents

- **agent** – one autonomous actor that takes your goal and runs with it end to end
- **subagents** - a setup where a main agent acts as orchestrator, delegating work to other agents it controls. The main agent owns the flow, order, and coordination.
- **multi agents** – two or more main agents, each acting independently but able to collaborate, negotiate, or exchange results. No single agent is "the boss".

These are just different ways to get stuff done with AI. Kind of like deciding if you want to work solo, pair program, or lead a squad.


Raw Materials
Idea Summary ($\mathcal{I}$)
Experimental Log ($\mathcal{E}$)
LaTeX Template ($\mathcal{T}$)
Conference Guideline ($\mathcal{G}$)
Step 1: Outline Agent
JSON Outline
Visualization Plan, Research Graph, Writing Plan
Step 2: Plotting Agent
Backbone: PaperBanana
Plots & Diagrams
Step 3: Literature Review Agent
Web Search
Identify
Semantic Scholar API
Verify
Intro & Related Work (.tex) + Citation Bank (.bib)
Step 4: Section Writing Agent
Full Manuscript Draft
Step 5: Iterative Refinement
AgentReview (evaluator)
Checks:
Improvement
Neutral
Degradation
Content Refinement Agent
PDF
Final Submission-Ready Paper

What is Git?

Git is a free, open-source distributed **version control system** designed to track changes in source code during software development. It allows multiple developers to work on the same project simultaneously without overwriting each other's work. Rather than storing full duplicates of your project folders, Git takes "snapshots" of your files over time, building a timeline you can easily navigate or roll back if something breaks.

Git vs. GitHub (They Are Different)

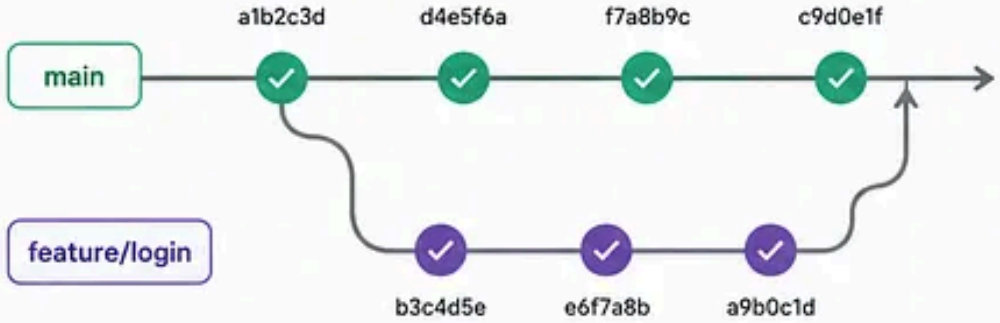
- **Git** is the local command-line software tool that manages your file history.
- **GitHub** is a cloud-based hosting platform where developers store their Git repositories to collaborate with others online. Alternative platforms include GitLab and

Git vs Agent History

CODE CHANGES vs DECISIONS & BEHAVIOR

GIT

Tracks code changes




```
</> app/
    auth.py      +
    routes.py    M
    user.py      M
    tests/       +
    README.md    M
```

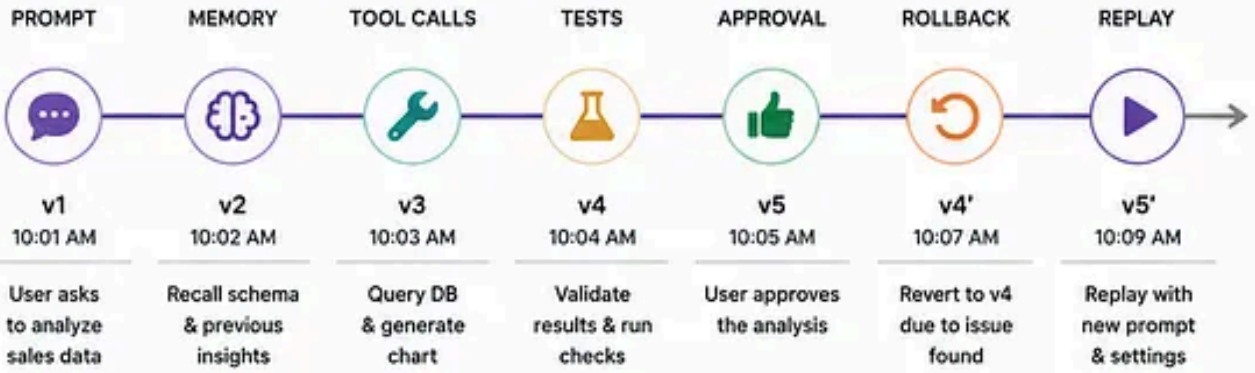
```
42  -  def login(user, pass):
43  -      if not valid(user):
44  -          return False
45  +  def login(user, pass):
46  +      if not valid(user):
47  +          raise AuthError()
```

Pull Request #42
Add login rate limiting
✓ Approved

VS

AGENT HISTORY

Tracks decisions & behavior



v3
10:03 AM

PROMPT
"Show monthly revenue by product line" for Q1"

MEMORY (DIFF)
+ Recalled sales_db schema
+ Prior analysis: Q4 trends
- Old assumption about currency

TOOL CALLS
```
run_sql(query)
generate_chart()
save_artifact()
```

TESTS
- Schema valid
- Row count > 0
- Chart rendered

OUTCOME
Success
Confidence 0.86

ROLLBACK
Revert to any checkpoint

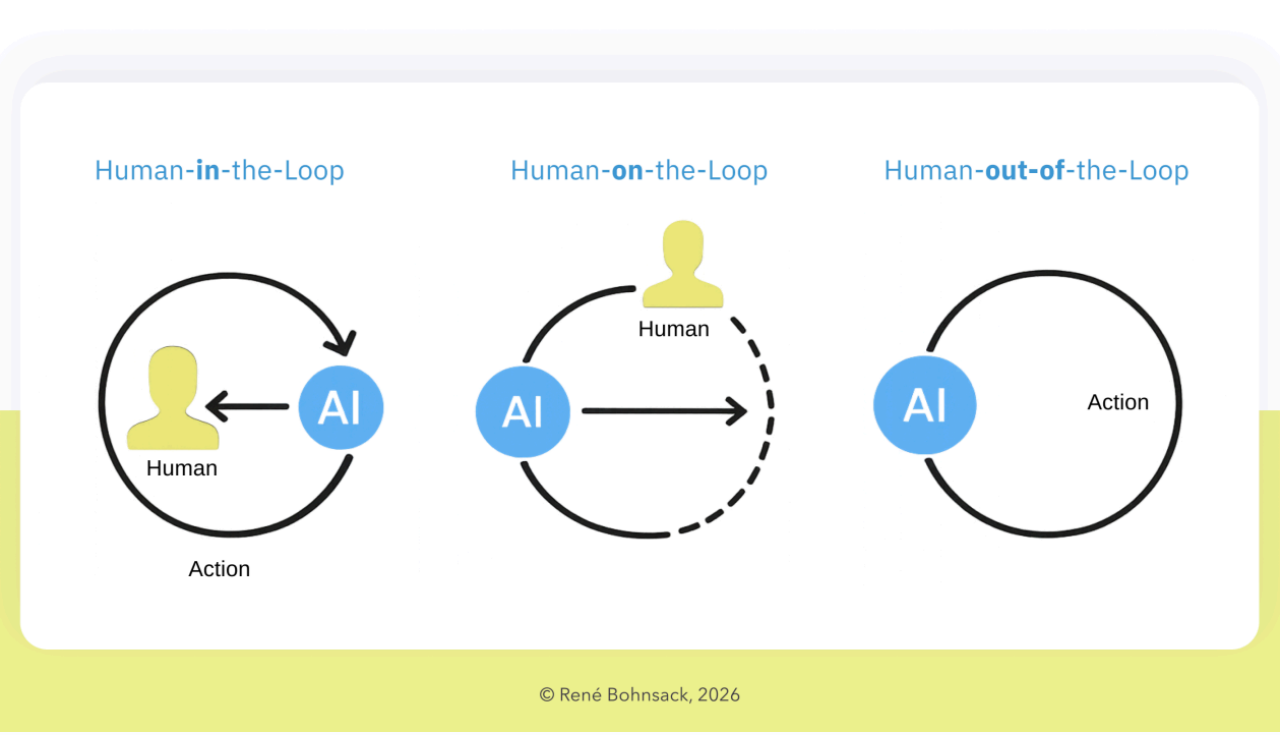
REPLAY
Re-execute from any point

Human In the Loop

Artificial intelligence (AI) and machine learning (ML) models rely on high-quality labeled data to function effectively. Annotation software plays a key role in labeling images, text, video, and audio, making AI training possible.

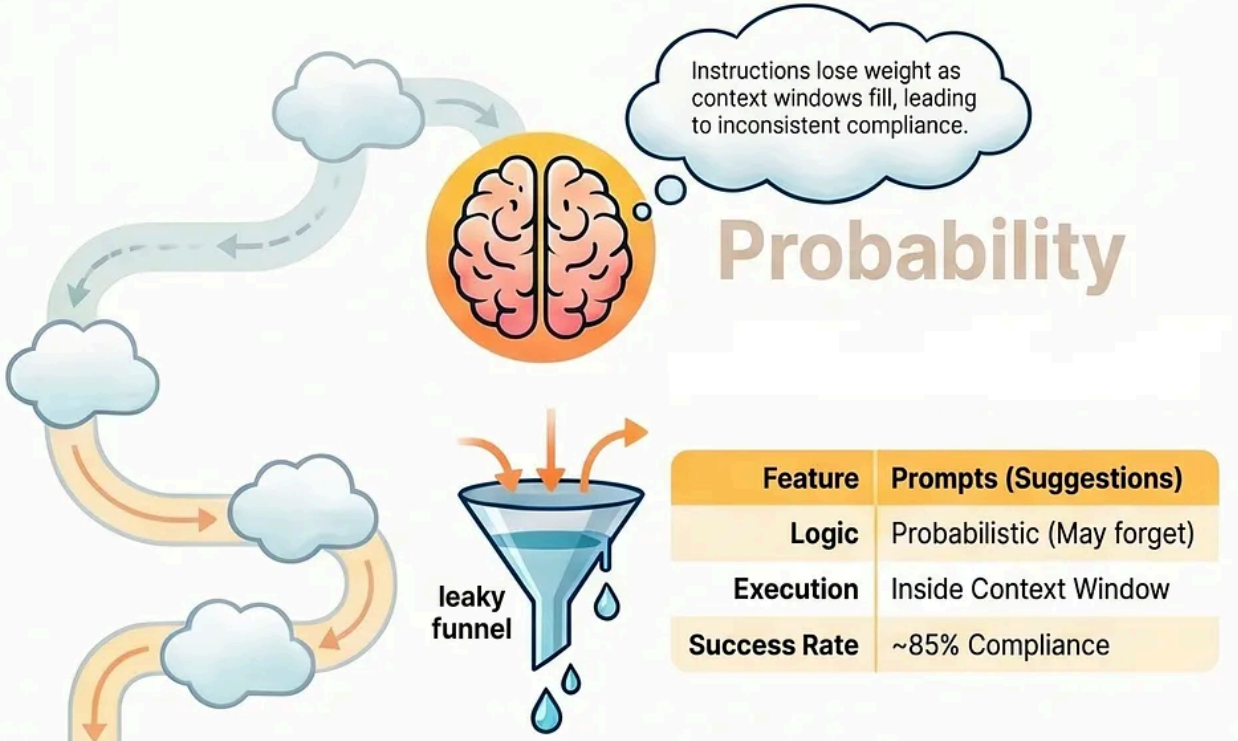
While automated softwares have significantly improved efficiency, human involvement remains essential in maintaining accuracy, handling complex cases, and reducing bias in AI models.

The approach that combines automation with human expertise is called human-in-the-loop (HITL) annotation. It ensures that AI models receive precisely labeled data, leading to better decision-making and higher accuracy.



Hooks vs. Prompts: Why Determinism Beats Probability in AI Coding

Prompts are Probabilistic Suggestions



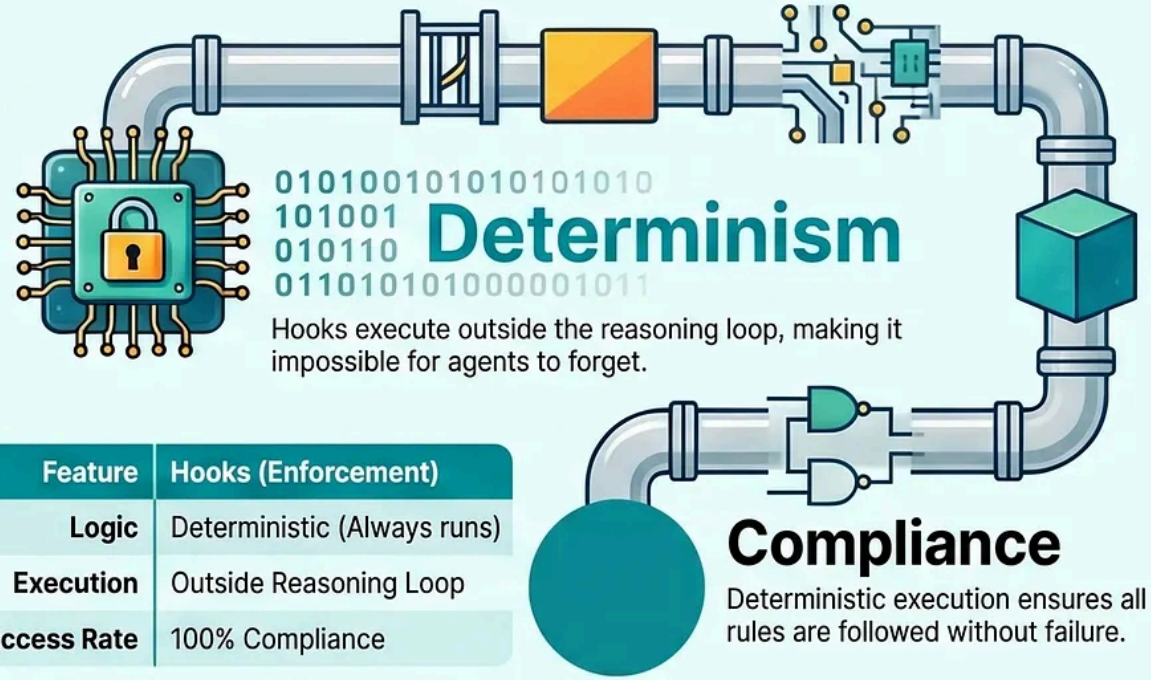
Feature	Prompts (Suggestions)
Logic	Probabilistic (May forget)
Execution	Inside Context Window
Success Rate	~85% Compliance

85%

Compliance Problem

Prompt-based instructions allow roughly 18 unchecked errors to slip through every two weeks.

Hooks are Deterministic Infrastructure

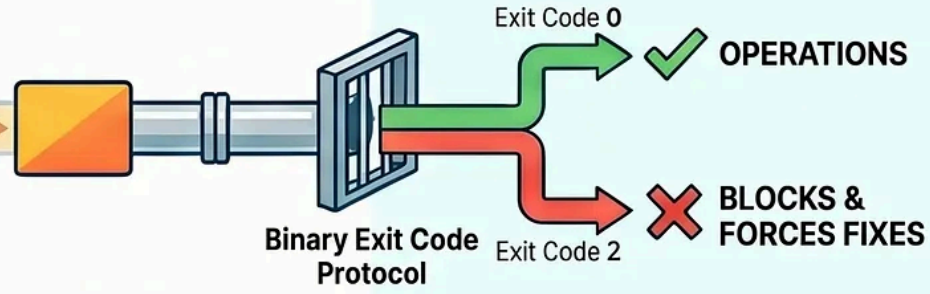


Feature	Hooks (Enforcement)
Logic	Deterministic (Always runs)
Execution	Outside Reasoning Loop
Success Rate	100% Compliance

Compliance

Deterministic execution ensures all rules are followed without failure.

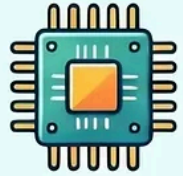
The Enforcement Mechanism



Deploy Command Hooks (Speed)

Prompt Hooks (Judgment)

Agent Hooks (Audits)



"Don't Stop" Quality Gate

Use Stop hooks to prevent completion until all automated test suites pass.

AI Agents for Economic Research

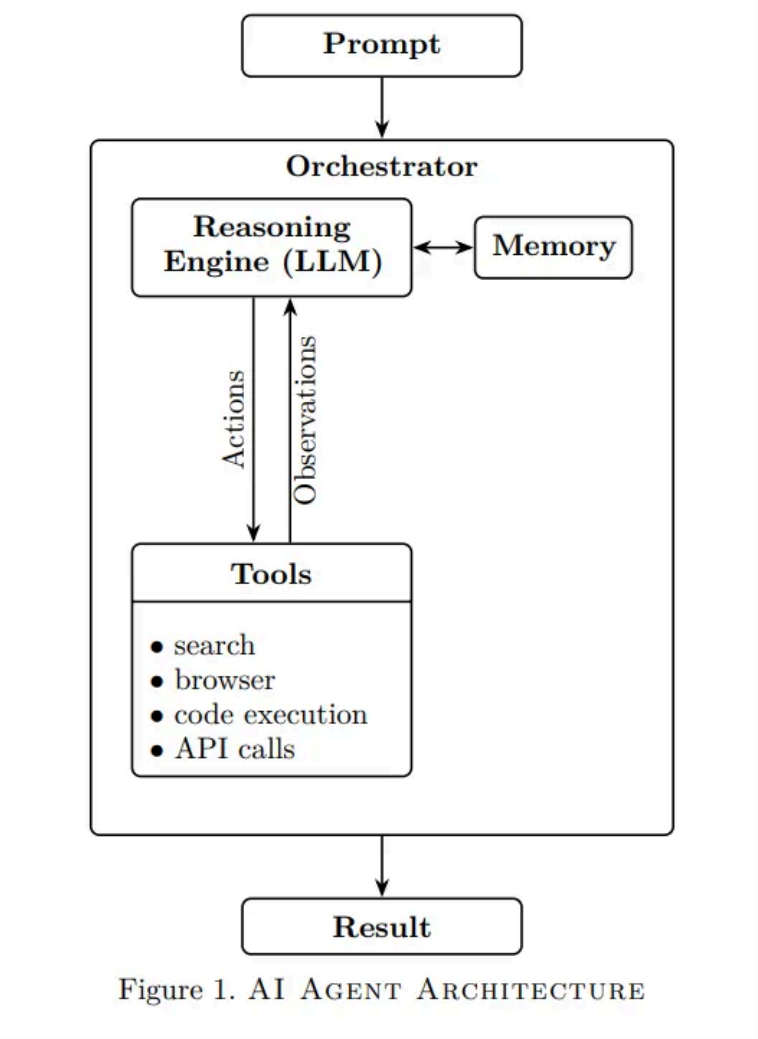


Figure 1. AI Agent Architecture

Deep Research Systems: Literature Reviews in Minutes

Deep Research agents demonstrate this capability in a powerful way. These multi-agent systems can parallel-process hundreds of sources in minutes, producing comprehensive research reports with accurate citations. When given a research question, an orchestrator agent:

1. Decomposes the question into focused subtasks
2. Spawns specialized agents to investigate different aspects in parallel
3. Synthesizes findings into coherent narratives

Deep Research systems are available in all the leading chatbots by click on “Deep Research” (see, e.g., in ChatGPT, Gemini, or Claude). While these systems compile existing knowledge rather than generating truly novel insights, they dramatically accelerate the time-intensive work of information gathering and initial synthesis. Literature reviews that once took weeks can now be completed in under an hour.

The quality of the output varies: these systems still struggle at prioritizing research at the frontier, but they are already quite capable at synthesizing well-established bodies of literature.

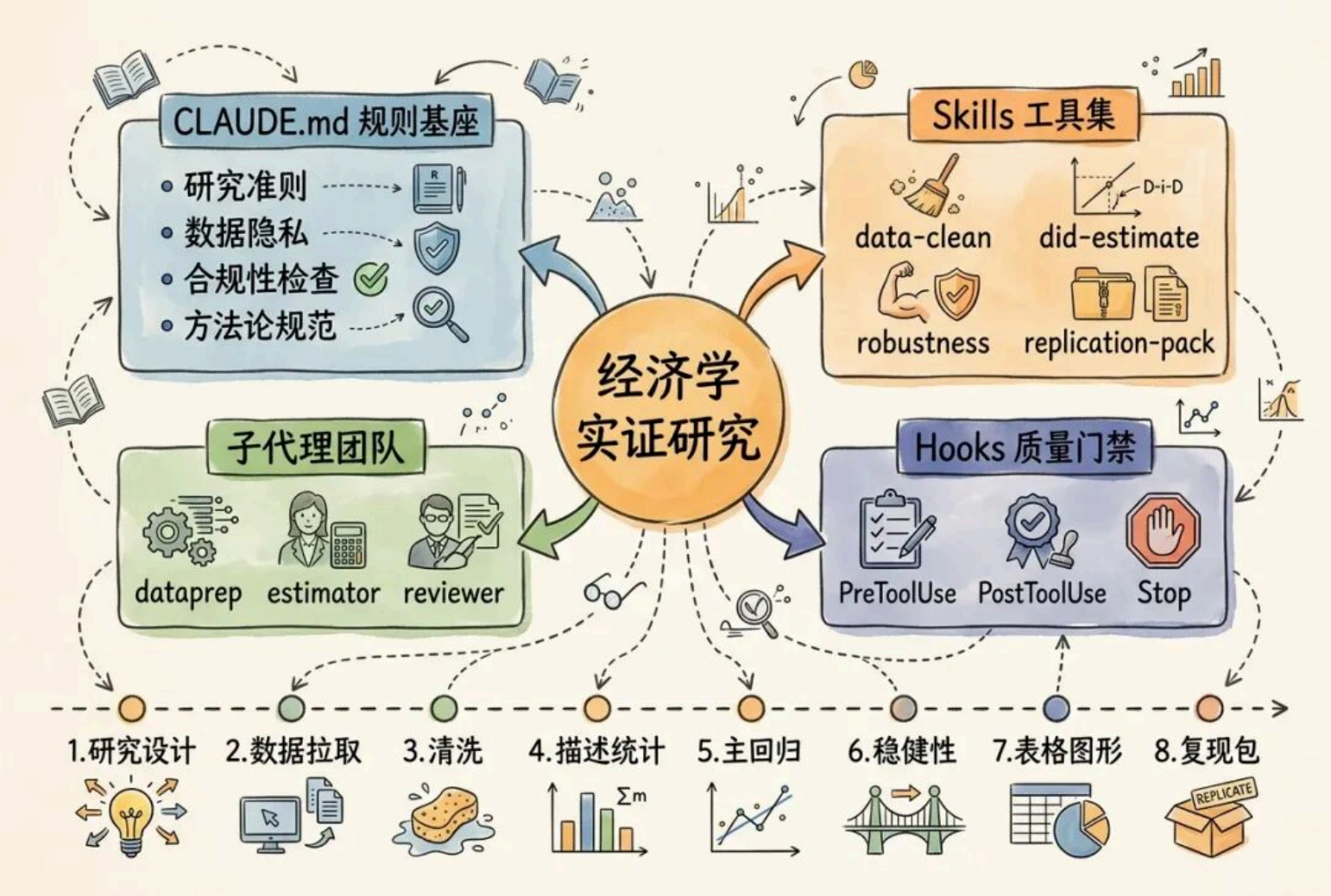
Vibe Coding and the Democratization of Technical Work

Perhaps the most transformative development for researchers is **"vibe coding"**—creating increasingly sophisticated software through natural language descriptions alone. Anthropic's Claude Code, released in February 2025, exemplifies this breakthrough. OpenAI has since released Codex, which exhibits with similar capabilities.

As is demonstrated in the paper's econometric tool example, researchers can now build complete analytical tools from simple descriptions. The system handles everything from file uploads to regression analysis to visualization—all generated in minutes without writing a single line of code.

For a discipline where computational methods are increasingly central, but programming skills remain unevenly distributed, this represents a potential leveling of the technical playing field.

The practical implications are clear: economists should engage with these tools now, not merely for productivity gains but to understand their capabilities and limitations. Building your own agents demystifies the technology and reveals both its power and boundaries. It also allows you to benefit more and more as the technology continues to advance.


CLAUDE.md 规则基座
研究准则
数据隐私
合规性检查
方法论规范
Skills 工具集
data-clean
did-estimate
D-i-D
robustness
replication-pack
经济学
实证研究
子代理团队
dataprep
estimator
reviewer
Hooks 质量门禁
PreToolUse
PostToolUse
Stop
1.研究设计
2.数据拉取
3.清洗
4.描述统计
5.主回归
6.稳健性
7.表格图形
8.复现包
REPLICATE

AI Agent配置

```
Windows PowerShell
版权所有（C） Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！https://aka.ms/PSWindows

PS C:\Users\zzyna> Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
PS C:\Users\zzyna> Get-ExecutionPolicy
RemoteSigned
PS C:\Users\zzyna> git --version
git version 2.54.0.windows.1
PS C:\Users\zzyna> git config --global user.name "xishanyu2"
PS C:\Users\zzyna> git config --global user.email "zzynankai@outlook.com"
PS C:\Users\zzyna> git config --global --list
user.name=xishanyu2
user.email=zzynankai@outlook.com
```

```
PS C:\Users\zzyna> node --version
v24.15.0
PS C:\Users\zzyna> npm install -g opencode-ai

added 3 packages in 24s
PS C:\Users\zzyna> opencode --version
1.15.0
PS C:\Users\zzyna> code --version
1.120.0
0958016b2af9f09bb4257e0df4a95e2f90590f9f
x64
```

```
(base) PS C:\Users\zzyna> pandoc --version
pandoc 3.8
Features: +server +lua
Scripting engine: Lua 5.4
User data directory: C:\Users\zzyna\AppData\Roaming\pandoc
Copyright (C) 2006-2025 John MacFarlane. Web:  https://pandoc.org
This is free software; see the source for copying conditions. There is no
warranty, not even for merchantability or fitness for a particular purpose.
```

这一步是在Anaconda Prompt里

```
(base) C:\Users\zzyna>conda init powershell
no change     D:\anaconda3\Scripts\conda.exe
no change     D:\anaconda3\Scripts\conda-env.exe
no change     D:\anaconda3\Scripts\conda-script.py
no change     D:\anaconda3\Scripts\conda-env-script.py
no change     D:\anaconda3\condabin\conda.bat
no change     D:\anaconda3\Library\bin\conda.bat
no change     D:\anaconda3\condabin\_conda_activate.bat
no change     D:\anaconda3\condabin\rename_tmp.bat
no change     D:\anaconda3\condabin\conda_auto_activate.bat
no change     D:\anaconda3\condabin\conda_hook.bat
no change     D:\anaconda3\Scripts\activate.bat
no change     D:\anaconda3\condabin\activate.bat
no change     D:\anaconda3\condabin\deactivate.bat
modified      D:\anaconda3\Scripts\activate
modified      D:\anaconda3\Scripts\deactivate
modified      D:\anaconda3\etc\profile.d\conda.sh
modified      D:\anaconda3\etc\fish\conf.d\conda.fish
no change     D:\anaconda3\shell\condabin\Conda.psm1
modified      D:\anaconda3\shell\condabin\conda-hook.ps1
no change     D:\anaconda3\Lib\site-packages\xontrib\conda.xsh
modified      D:\anaconda3\etc\profile.d\conda.csh
modified      C:\Users\zzyna\Documents\WindowsPowerShell\profile.ps1

==> For changes to take effect, close and re-open your current shell. <==


(base) C:\Users\zzyna>conda --version
conda 25.11.1

(base) C:\Users\zzyna>python --version
Python 3.13.9
```

```
(base) PS C:\Users\zzyna> git --version
git version 2.54.0.windows.1
(base) PS C:\Users\zzyna> git config --global user.name
xishanyu2
(base) PS C:\Users\zzyna> node --version
v24.15.0
(base) PS C:\Users\zzyna> npm --version
11.12.1
(base) PS C:\Users\zzyna> opencode --version
1.15.0
(base) PS C:\Users\zzyna> code --version
1.120.0
0958016b2af9f09bb4257e0df4a95e2f90590f9f
x64
(base) PS C:\Users\zzyna> conda --version
conda 25.11.1
(base) PS C:\Users\zzyna> pandoc --version
pandoc 3.8
Features: +server +lua
Scripting engine: Lua 5.4
User data directory: C:\Users\zzyna\AppData\Roaming\pandoc
Copyright (C) 2006-2025 John MacFarlane. Web:  https://pandoc.org
This is free software; see the source for copying conditions. There is no
warranty, not even for merchantability or fitness for a particular purpose.
```

```
(base) PS C:\Users\zzyna> mkdir $env:USERPROFILE\opencode-lab


    目录: C:\Users\zzyna


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         2026/5/16     21:06                opencode-lab


(base) PS C:\Users\zzyna> cd $env:USERPROFILE\opencode-lab
(base) PS C:\Users\zzyna\opencode-lab> opencode
```

03

AI赋能的未来

> 老子的《道德经》里说，万物负阴而抱阳，冲气以为和。而人类与AI，或许就会在相互的影响中，形成新的和谐体。AI，可能是我们的对手，但它，也能是我们的助手。

人工智能哲学

人工智能焦虑（AI Anxiety）

Jensen Huang says it doesn't matter what kids study in the AI era

Parents shouldn't obsess over what their kids study in the era of artificial intelligence, said Nvidia CEO Jensen Huang.

"I think that it won't matter. All the things that used to matter are still things that are going to matter in the future," Huang told Singapore's Channel NewsAsia on Monday.

Instead of chasing AI-proof subjects, **students should focus on using AI to deepen their learning and improve their craft**, Huang said.

The Nvidia chief pointed to journalism, **storytelling, the arts, and design as examples of fields that will remain valuable even as AI becomes more powerful.** He pointed out **the best interviewers are not just well prepared, but able to stay present, listen closely, and respond dynamically in the moment.**

"The ability to tell a story for an audience will remain just as important in the future as it is today," Huang said.

The Nvidia chief also referenced the Japanese concept of "wabi-sabi," or the beauty of imperfection, suggesting that uniquely human qualities could become even more prized in an AI-saturated world.

"Whatever you decide is your passion, the only one thing that you have to do is to make sure that you ask yourself: How can AI help elevate my learning, my craft, my purpose?" he said.

Huang is the latest business leader to weigh in on how AI could reshape education and work.

Earlier this month, futurist and entrepreneur Peter Diamandis told Business Insider that **kids will need qualities such as curiosity, purpose, and adaptability to succeed in the AI era.**

Meanwhile, entrepreneur and entrepreneur-turned-professor Scott Galloway said on "The Diary of a CEO" podcast that parents should focus on helping children **develop durable human skills such as storytelling, communication, and relationship-building.**

Huang echoed those themes, arguing that while AI would automate parts of many jobs, **it would also push people toward higher-level work requiring judgment and creativity.**

"A job is like a basket of tasks," he told CNA. "Many of those tasks will be automated. And my sense is that as a result of automation, **we can focus on the harder parts of our work**."

Huang also pushed back on concerns that widespread AI use could make people less intelligent or lazier thinkers.

Drawing comparisons to the rise of personal computers, the internet, and smartphones, Huang argued that previous waves of technology increased human ambition rather than diminishing it.

"Do we find ourselves busier or less busy? I think the answer is we found ourselves busier," Huang said.

AI的负外部性与监管责任

Dangers of Artificial Intelligence

- Automation-spurred job loss
- Deepfakes and social manipulation
- Privacy violations
- Algorithmic bias caused by bad data
- Socioeconomic inequality
- Weapons and military automatization
- Market volatility
- Increased criminal activity and child safety risks
- Psychological harm and overreliance

课程结语

是结束，也是开始：

知识的边界在哪里？

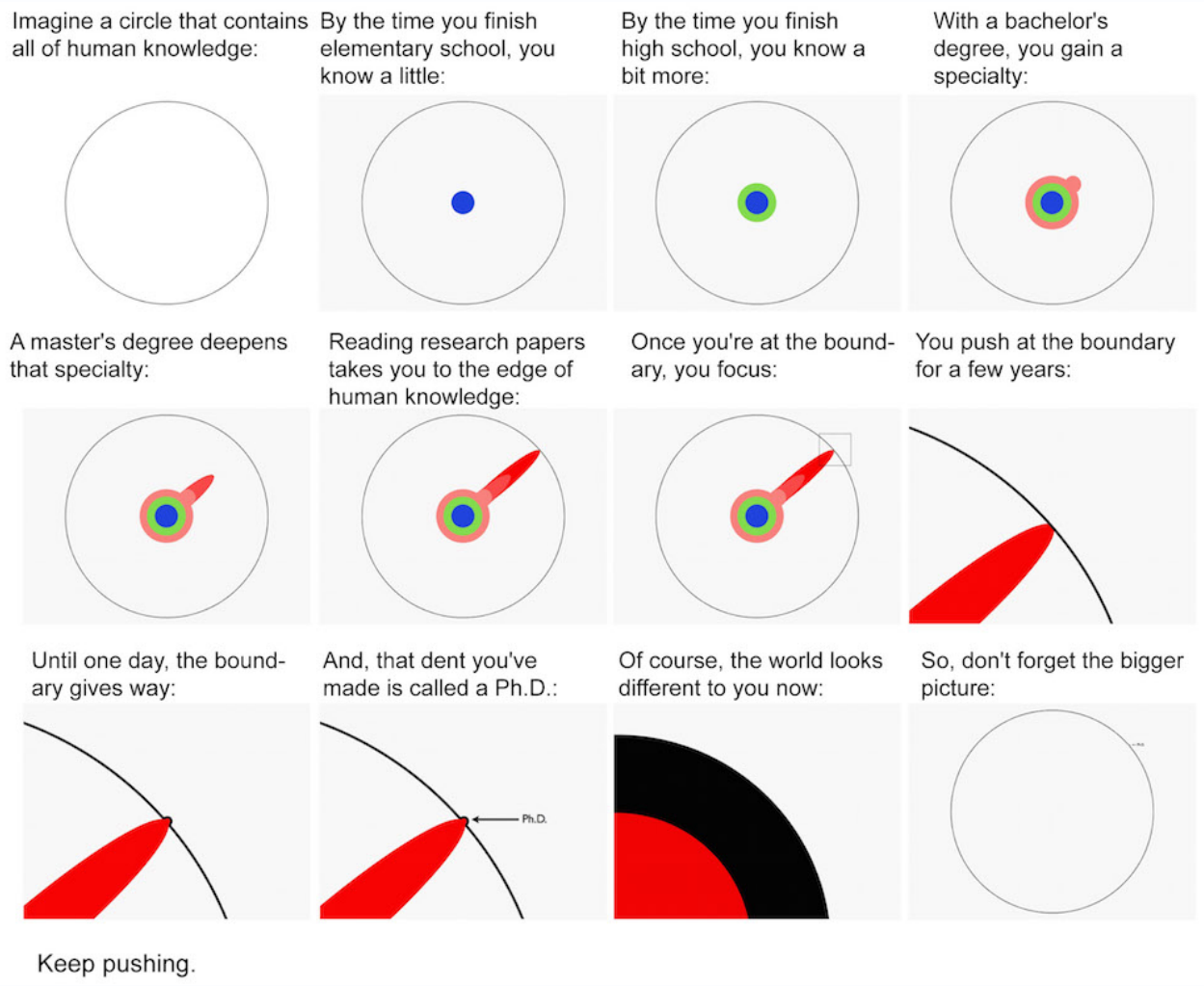
AI的能力还有多少？

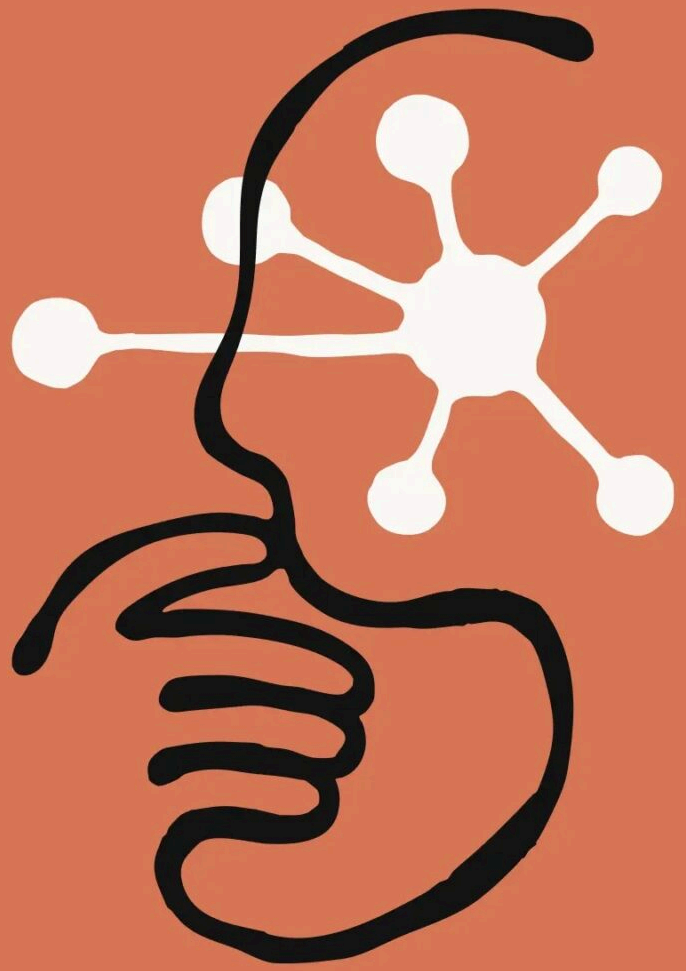
人类的上限在何处？

不要丧失对世界的好奇

不要停止对真理的追求

千里之行，始于足下...





附录1：Agent Skills如何改变知识工作者的工作方式

2026年5月15日 中山大学岭南学院 李学恒老师 讲座

AI科研范式转变

- AI技术从对话聊天（Chatbot）迭代至代理基础（Agent-based），具备执行多环节任务的能力。
- AI已实现从选题、文献综述、建模到代码生成、文章评审的一体化自动工作流，彻底颠覆科研方式。

知识工作者工作方式的重构

- 传统工作模式受限于精力，仅能完成少量主线任务，大量支线任务被搁置。
- 新工作模式分为三类：
  - 纯AI任务：AI自主完成99%以上工作，占比超过50%。
  - 人机协作任务：AI执行，人类负责验收、反馈与迭代，占比约40%。
  - 纯人工任务：涉及核心理论证明或最终决策，占比极低。
- AI带来的核心价值不仅是效率提升，更是“Except me free”，即释放人类去从事创造性想法和 previously impossible 的支线任务。

AI智能体的层级与Claude Code特性

- L0-L1：网页版对话AI（如ChatGPT网页版），仅有“嘴”，无法直接操作文件，存在黑箱操作和信息截断问题。
- L2：具备初步代码生成能力。
- L3：用智能体框架（如Claude Code），具备“手”，可直接读写本地文件、执行命令。
- L4-L5：调用外部工具，实现长时间自主运行，完成从数据清洗到LaTeX论文初稿生成的全流程。
- Claude Code vs. Open Interpreter：
  - Open Interpreter是"看不见的手”，存在安全隐患且过程不透明。
  - ClaudeCode是"看得见的手”，运行在本地VS Code终端，每一步操作透明可见，用户可实时干预权限，更安全可靠。

核心概念：Markdown为中心的工作流

- 为什么需要Markdown：大语言模型本质只能处理纯文本。PDF、Word等格式需经预处理才能被AI完整读取，否则易出现信息丢失或幻觉。
- 工作流程：
  i. 将各类文件（PDF/Word）转换为Markdown（.md）
  ii. AI基于Markdown进行内容处理、总结或生成。
  iii. 将生成的Markdown转换为目标格式（Word/PDF/LaTeX）
- Markdown通过符号（如#表示标题，|表示表格）保留结构信息，对AI友好且易于人类校对。

关键机制：Skils（技能/流程说明书）

- 定义：Skils并非外部工具，而是包含最佳实践和工作流指令的Markdown文档流程说明书)。
- 作用：指导AI如何高效、规范地执行复杂任务，避免重复试错。

关键机制：Skils（技能/流程说明书）

- Web Research Skill案例解析
  - PlanFirst：先制定调研计划，分解子主题。
  - Delegate：派遣多个子Agent并行处理不同子主题。
  - Synthesize：整合各子Agent结果，生成最终报告。
  - Best Practices：要求引用来源、避免列点式公文风格、保持逻辑连贯。
- Skill的安全性
  - 风险源于安装不明来源的Skill（可能包含恶意指令）。
  - 建议选用高星、开源社区验证过的Skil，或自行审查代码。
  - 数据安全性取决于模型提供商，本地软件仅赋予操作权限，数据仍需发送至云端模型处理。

实战演示：自动生成调研报告与课件

- 网络调研报告生成
  - 调用 `web_research` Skill，自动搜索国内外顶尖高校AI转型战略。
  - 生成符合高校公文规范的Word文档，包含完整引用和来源链接，便于人工复核。
- 课件一键生成
  - 调用 `slide_creator` Skill，基于调研文档生成PPT。
  - 流程：分析材料提取大纲 -> 生成初稿 -> 多Agent并行审阅（语言、格式、内容一致性） -> 输出最终幻灯片。
  - 优势：内容结构清晰，排版专业，大幅减少手动调整时间。

版本控制与数据安全

- Git集成：建议在VS Code中使用Git进行版本控制，AI可自动执行commit操作，实现"时光机器”般的版本回溯，防止误操作导致的数据丢失。
- 敏感数据处理：
  - AI在处理实证数据时，通常仅读取变量名或少量样本行以生成代码（Stata/R/Python），不直接接触全部原始数据。
  - 用户需评估数据泄露风险，必要时对敏感字段进行脱敏处理。

AI时代的教育者角色重塑

- Task vs.Job：
  - Task(任务)：如制作PPT、批改作业、填表等重复性工作，将被AI替目。
  - Job(职责)：如激发求知欲、批判性思维培养、团队激励、方向规划，是AI无法替代的人类核心价值。
- 教育者新定位：从知识传授者转变为学习设计师和灵感激发者，重点在于定义学习目标、凝聚人心和指导方向。

模型选择与建议

- No Pay No Gain：
  - 建议使用付费高阶模型（如Claude Pro $100/月），免费或低额度套餐无法发挥Agent潜力。
- 模型对比：
  - Claude：综合任务最可靠，代码生成能力强，但最新模型中文表现稍弱。
  - OpenAI（GPT-4o/o1）：数学推理较强，但近期更新节奏放缓。
  - 国内模型（DeepSeek/Kimi）：性价比高，Kimi在中文表达和长文档处理上表现优异，适合中文场景。
- 建议：初学者应“**吃点好的**”
  - 直接使月级模型以建立对AI能力的正确认知，再根据需求组合使用不同模型。

附录2：AI、宏观经济与 Agentic Research

2026年5月28日 中国人民大学经济学院 陈朴老师 讲座

- **AI 对劳动力市场的影响1**
  - **任务类型与生产方式**
    - 两类任务：工作任务可分为两类，一类是 AI agent 擅长的任务，另一类是需要人类沟通、判断和承担责任的任务。
    - 两种生产方式：存在 bundle 和 unbundle 两种生产方式。bundle 方式中人类整合两类任务；unbundle 方式中 AI 做第一类任务，人类只负责第二类任务，这种方式下人类工作边界变窄。

Task 1: 可验证/可编码认知	Task 2: 情境/人类判断
起草、分类、总结、计算、编码	沟通、责任、解释、现场判断
AI/agent 很擅长	人类仍然关键
容易标准化和度量	难以完全写入 contract

- Weak Bundle 和 Strong Bundle：工作可分为 Weak Bundle 和 Strong Bundle。Weak Bundle 任务联系不紧密，拆分成本低，工作边界易变窄，人类劳动份额易下降；Strong Bundle 任务联系紧密，拆分成本高，AI 多为辅助工具，人类工作边界稳定，劳动份额有韧性。

	Weak bundle	Strong bundle
拆分成本	低	高
任务关系	相对独立	高度互补
AI 的作用	替代部分任务	岗位内部增强
工作边界	变窄	保持完整
劳动收入份额	更容易下降	更有韧性

- **AI 对劳动力市场的影响2**
  - **AI 能力提升的影响**
    - 对不同职业的冲击：随着 AI 能力上升，协调成本低的 Weak Bundle 工作更容易受到冲击，能力低的工人会转向剩余任务；协调成本高的工作可能保持相对稳定。
    - 裁员机制：AI 完全接管任务一，人类将全部时间投入任务二，会使市场产品供应大幅上升。若需求弹性低，产品价格下降，生产率低的工人可能被挤出市场，导致产出提高但就业下降。
  - **需求弹性的影响**
    - 杰文森悖论：工业革命时蒸汽机效率提升，理论上煤消耗应减少，但实际需求上升，因为成本下降使需求迅速增加。
    - 不同行业的影响：创意类、研发或设计等行业，AI 出现可能使服务费用下降，需求增加，行业扩张；税务、基本会计等固定需求行业，受 AI 冲击较大，因为需求变化相对较小。

- **AI 对劳动力市场的影响3**
  - **应对建议**
    - 提高工作协调成本：建议将技能、责任和沟通等捆绑在一起，提高工作协调成本，减少 AI 冲击。
    - 利用 AI 进入高门槛任务：利用 AI 降低进入门槛，开展交叉性研究，与他人合作。
    - 提出问题和承担责任：不要只做工具人，要学会提出问题、协调资源并承担责任。

- **AI agent 对研究的帮助1**
  - **AI agent 的重要性**
    - 降低协调成本：AI agent 不仅能提高单个任务效率，更重要的是能降低任务的拆分和协调成本，改变工作边界，使人类负责监督、签字和承担责任等工作。
    - 适用于研究任务：研究工作可拆分成不同任务，AI agent 适合完成文献检索摘要、数据清理、运行脚本、生成结果等任务，但涉及专业判断和安全问题的任务需人类主导。

研究任务	是否适合 agent?
文献检索和摘要	适合，但要验证引用和 claims
数据清理和 profiling	很适合，规则清楚、可验证
回归和图表生成	适合，但要审查识别和标准误
识别策略判断	strong bundle，研究者必须主导
论文解释和政策含义	strong bundle，需要责任和领域判断

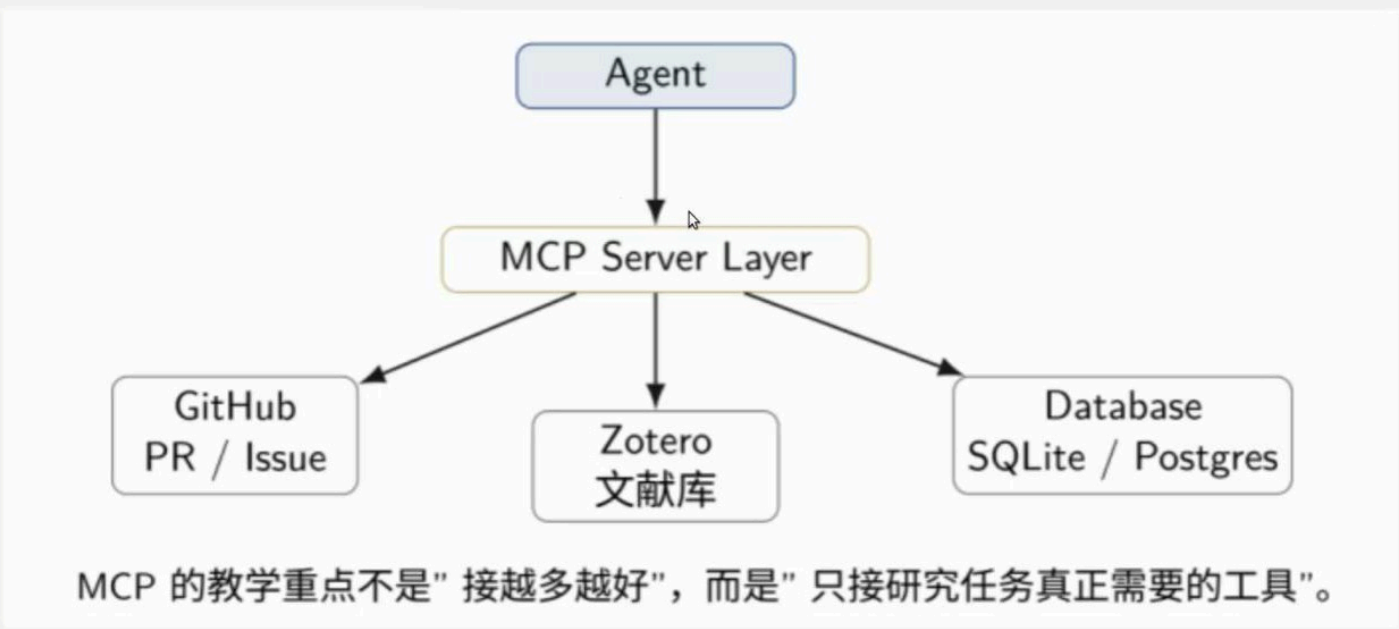
- **AI agent 对研究的帮助2**
  - **研究任务的拆分**
    - Weak Bundle 和 Strong Bundle 任务：研究中，运行脚本等联系不太紧密的任务属于 Weak Bundle；识别策略判断、文章解释和政策应用等联系紧密的任务属于 Strong Bundle，更依赖人类主导。
    - 任务分配：将研究流程拆分成不同模块，让 AI agent 执行重复性任务，人类将注意力集中在研究判断上。
  - **AI agent 的使用**
    - 适合交给 agent 的任务：能够经常重复、有明确输入输出、可验证的任务适合交给 agent，如文献读取、数据检查、代码改进、图表生成、论文写作和审稿回复等。
    - 注意事项：涉及专业判断和安全问题的任务需人类自己判断，注意数据安全，避免将机密信息交给 agent。构建工作流程，让 agent 执行任务，人类进行审查

- **主流 AI agent**

工具	适合场景
Claude Code	研究工作流、长上下文、skills/rules/agents 生
GitHub Copilot CLI	GitHub 生态、代码任务、终端协作
Codex CLI	OpenAI 用户、代码生成和实现任务

- **使用 AI agent 的基础**
  - 知识储备：了解运行命令行脚本的基本知识，掌握 GitHub 命令和 API 的使用。
  - 配置文件：使用 Markdown 格式的编写文件，让 agent 了解任务边界、数据、命令、输出等信息。
  - 研究准备：弄清楚研究问题、数据结构、可用代码，以及实证工作和稳健性判断等内容。

- **AI agent 的相关组件和应用1**
  - **MCP**
    - 作用：连接工具和外部资料的接口，可让 agent 调取外部资料和工具，如爬虫、文献库查询等。
    - 权限控制：使用 MCP 时要注意权限控制，只开放当前项目所需目录，避免暴露电脑信息，不将 API key 等私人信息放入 prompt。

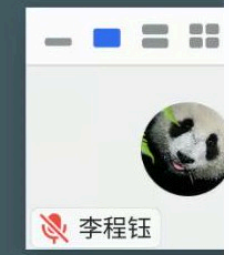


- AI agent 的相关组件和应用2
  - Skills
    - 定义和优势：可复用的工作流，包含说明文档和脚本，只有在需要时加载，可节省 token 数量。
    - 示例和应用：例如做实证分析的 skill，可在提到相关关键词时调用，包含回归命令、标准差计算等规则。常见的 skill 包括生成回归结果、审查论文、制作幻灯片等。

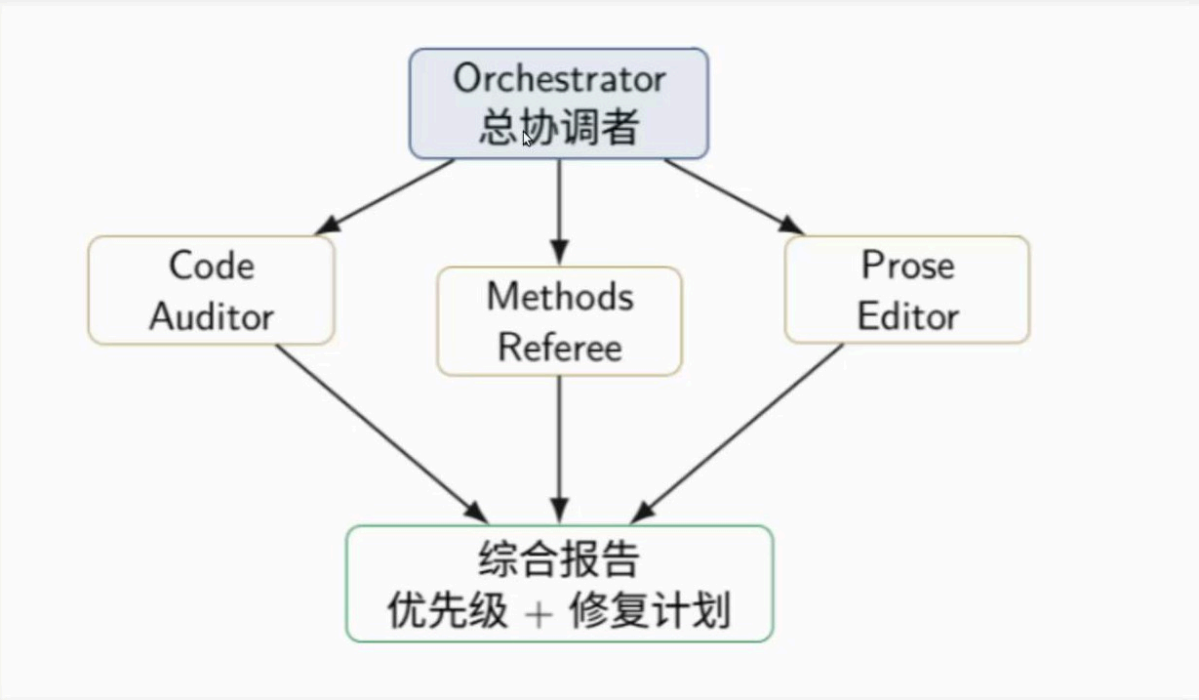
```
---
name: r-empirical-economics
description: >
  Use this skill whenever writing, reviewing, or debugging
  R scripts for panel data, event studies, DiD, IV/2SLS,
  clustered standard errors, or regression tables.
---

# R Empirical Economics Standards

1. Prefer fixest::feols() for regressions.
2. Cluster standard errors unless the design says otherwise.
3. Never modify Data/Raw/.
4. Every output table needs a companion notes file.
```



- **AI agent 的相关组件和应用3**
  - **Subagents**、Orchestration(编排)
    - 作用和优势：将任务拆分开，用独立上下文处理任务，避免上下文快速增加，减少干扰。可根据任务独立性和复杂度，为不同 subagents 设置不同模型，控制 token 消耗。
    - 应用场景：适合处理相对独立的任务，如代码审查、识别策略判断、语言表述检查等。



- **研究系统的构建和运行**
  - **研究流程**
    - 提出问题：明确研究问题、数据信息、输出要求等，编写好的 propose 应包含角色、任务、输入、输出和限制条件。
    - 构建 workflow：构建 agent 的 workflow，将任务拆分成不同步骤，如计划、执行、验证、审查和修复。
    - 避免记忆问题：为避免 agent 遗忘之前的任务内容，可让 agent 将工作内容保存在文件中，下次从该文件继续工作。

Phase	做什么
Plan	MUST / SHOULD / MAY；明确边界和验证标准
Implement	agent 在 guardrails 内执行
Verify	运行脚本、检查输出、核对数值
Review	多个 subagents 并行审查
Fix	修 critical / major，再回到 verify

附录3：AI智能体在经济金融研究中的应用

2026年6月2日 澳门城市大学金融学院 许文立老师 讲座

1. AI智能体的核心价值与认知澄清

- AI智能体并非简单的聊天机器人，它能自主调用本地软件（如Stata, Python），生成和执行代码，处理数据分析任务，减少了人为操作和报错。
- AI生成的分析结果基于用户提供的客观数据，因此不具备“幻觉”问题，但其输出仍需研究者进行审核。
- 当前AI无法“一键生成”发表级别的论文，但能极大提升研究者的效率，尤其是在重复性工作上。

2. 主流AI编程工具对比

- 全球主流工具为Cloud Code和OpenAI Codex，功能强大，推荐使用。
- Kimi Code是国内工具，若无法使用主流工具，也可体验其基础功能。

3. 典型应用场景与案例分享

- **数据处理与收集**: AI可协助研究者进行文本数据整理、大规模数据爬取（需遵守robots协议）及数据清洗。
- **文献整理**: AI能高效地从大量文献中提取、总结并整理信息，生成结构化的表格，节省了大量时间。
- **实证分析**: AI擅长执行回归分析、绘制图表（如事件研究图），其生成的图表质量很高。但研究者仍需明确指令（Prompt），并检查AI生成的代码和结果。
- **DSEG建模**: 介绍多智能体协作工作流，该工作流能协助分解建模任务，如校准参数、求解稳态、调试代码及进行经济学解释。

4. 使用原则与未来展望

- AI是强大的助手，而非“黑箱”。研究者需具备专业知识，明确指令，并对AI的输出进行审核和验证。
- 推荐使用成熟的学者制定的工作流（Workflow），将其固化为Markdown文档，以规范化研究过程。
- AI的普及将提升整体研究效率和质量，但要求研究者自身具备更强的严谨性和批判性思维。

附录4：扣子全链路一站式ai工作台 释放你的创意和生产力

> 2026年6月4日 火山引擎高级解决方案架构师 施泽晶 讲座

- 扣子 3.0 全新升级，欢迎咨询下单 - 南开飞书云文档
- 扣子3.0离谱更新：把Codex、Claude Code拉进一个项目工作?

- **扣子平台介绍**
  - **平台背景与应用范围**
    - 团队背景：施泽晶来自字节跳动的火山引擎团队，该团队将字节跳动原生能力对外输出。
    - 高校应用情况：扣子已在南开大学及全国多个高校广泛使用，如南开大学的软件学院、生科院、文学院、商学院等都大量使用，且各学院还自建了很多功能。
  - **平台升级与能力特点**
    - 全面升级：扣子从去年的智能体 agent 全面升级为多人多智能体，具备开箱即用的特点，可进行编程和日常任务处理。
    - 集成多种能力：集成了Claude Code、Codex 等能力，是一个 AI 全面工作站，能伴随用户使用，越用越了解用户需求；具备经过公司严格审查的行业专家技能，对科研论文收集分析等有帮助。
    - 跨多端工作：支持跨多端工作，老师可通过移动端操作电脑，整理资料并定时发送到手机，还可通过移动端完成工作。

- **扣子平台使用方法1**
  - **技能商店与使用**
    - 技能商店功能：技能商店包含自媒体、金融、法律、互联网、科研教育等多种技能，可一键添加到技能包使用。
    - 技能使用示例：以科研技能为例，可进行论文检索、分析等工作；还包括学术审评者技能，能模拟 5 个专业领域审稿人进行论文评审。
    - 技能添加与使用要点：添加技能时可选择加到智能体中，使用时若不清楚技能功能可询问，同时要注意模型选择，文字处理建议用 DeepSeek 等，代码处理用 GLM 或豆包等，有 API key 可添加自定义模型。

- **扣子平台使用方法2**
  - **实际操作案例**
    - 科研分析技能：`科研分布 Skill`，该技能可按步骤进行研究分析，将结果存到文档目录，供后续加工和启发思考。
    - 参会申请技能：针对参会申请，创建相应技能，通过一系列操作实现自动填表，在重要决策时会提示用户，确保信息安全。
    - 资料整理技能：让扣子帮自己整理微信聊天记录和资料，授权后可定期完成整理任务，用户可通过手机随时调用资料。

- **扣子平台功能拓展**
  - **视频创作功能**
    - 视频创作流程：利用字节 Seedance2.0 模型进行对话式视频创作，可生成脚本和大纲，根据用户反馈生成内容和分镜，用户可上传参考素材。
    - 费用消耗：视频创作费 token，每秒视频约花 1000 积分，相当于1元，修改视频建议逐段修订。
  - **编程功能**
    - 编程技能特点：编程技能与Claude Code 类似，可开发网页应用等，建议选择 GLM 等模型，完成后可发布、预览。
    - 教学应用：可利用编程功能制作教学教案和材料，方便教学。
  - **多智能体协作**
    - 智能体组合应用：用户可构建大型项目组，组合多个智能体，如查找数据、自媒体运营等，让智能体互相沟通协作完成工作。
    - 案例分享：利用多智能体进行 AI for science 研究的案例，多个智能体协同完成调研、写作、PPT 制作等工作。

- **使用建议与注意事项**
  - **使用建议**
    - 主动提问：在扣子对话框中主动提问，选好技能，让系统告知操作步骤，逐步上手使用。
    - 多次使用：多次使用扣子，系统会根据用户历史记录更了解用户需求，提供更精准的服务。
  - **注意事项**
    - 视频制作：视频制作费算力，修改时逐段修订，避免整体重新生成。
    - 技能收费：部分上架技能收费，公司认证和自建的免费，用户可上传自己的技能。
    - 数据安全：企业版不捕捉信息，私域库可通过 VPN 通道调用，输入输出 token 不留存，用户数据存于安全沙箱。