Agent的自主规划与工具开发

>> 今天的学习目标

Agent的自主规划与工具开发

- AI Agent对比
- 智能体自主规划设计

反应式 (Reactive)

深思熟虑 (Deliberative)

混合式 (Hybrid)

- CASE：私募基金运作指引问答助手（反应式）
- CASE：智能投研助手（深思熟虑）
- CASE：投顾AI助手（混合式）

AI Agent

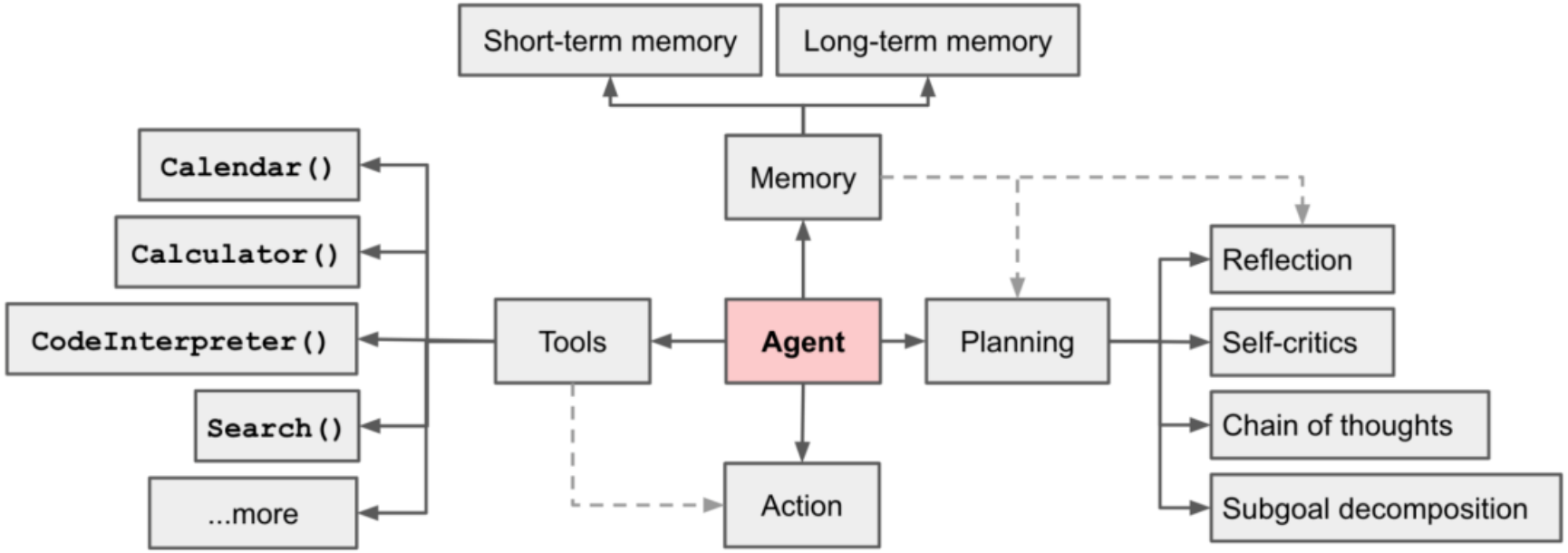
AI Agent：

- 通过LLM，能够自主理解、规划、执行复杂任务的系统。给它一个目标，AI Agents就能完成剩下的全部工作

规划（Planning）：将任务分解为较小的、可管理的子目标

记忆（Memory）：短期记忆，进行上下文学习；长期记忆，一般通过外部载体储存和快速检索来实现。

工具使用（Tool use）：调用外部API，获取额外信息



Generative Agents

从自动化到自主化：

- Generative Agents: Interactive Simulacra of Human Behavior，2023

https://arxiv.org/pdf/2304.03442v1.pdf

- Google和Stanford学者构造了一个《西部世界》小镇，打造模拟人类行为的Agent
- Generative Agents（生成式智能体），利用生成模型来模拟可信人类行为的Agent，并证明它们能生成可信的个人和突发的群体行为

Generative Agents: Interactive Simulacra of Human Behavior

Joon Sung Park
Stanford University
Stanford, USA
joonspk@stanford.edu

Joseph C. O'Brien
Stanford University
Stanford, USA
jobrien3@stanford.edu

Carrie J. Cai
Google Research
Mountain View, CA, USA
cjcai@google.com

Meredith Ringel Morris
Google Research
Seattle, WA, USA
merrie@google.com

Percy Liang
Stanford University
Stanford, USA
pliang@cs.stanford.edu

Michael S. Bernstein
Stanford University
Stanford, USA
msb@cs.stanford.edu

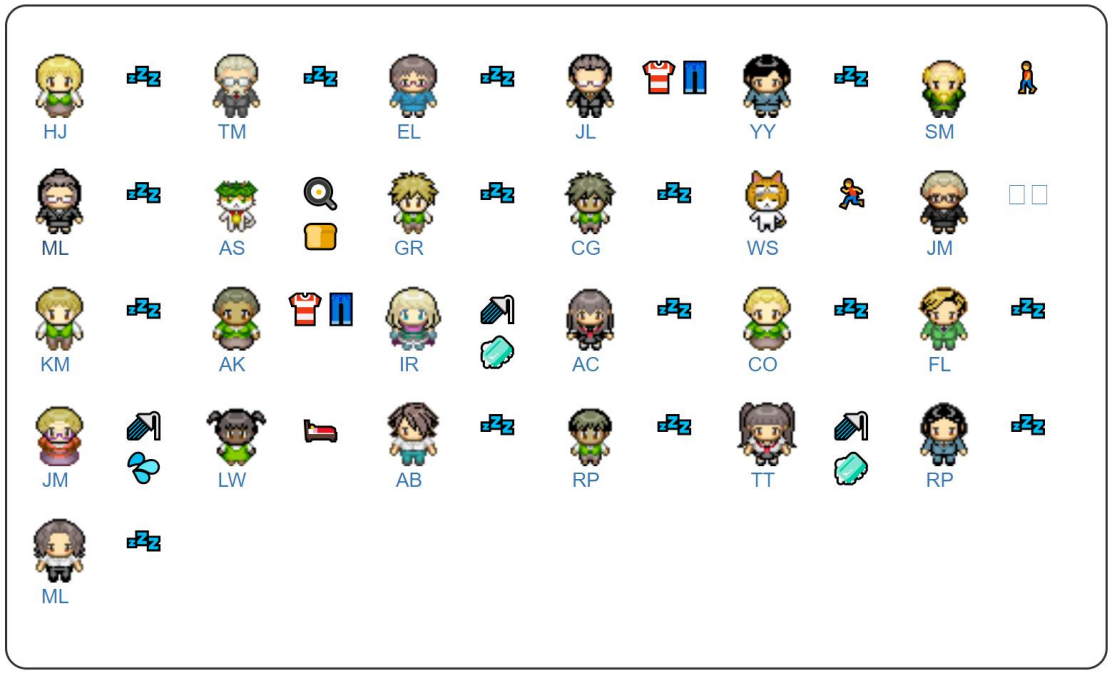
Generative Agents

This is a pre-computed replay of a simulation that accompanies the paper entitled "Generative Agents: Interactive Simulacra of Human Behavior." It is for demonstration purposes only.

Current Time:

Monday, February 13, 2023 at 6:13:10 AM

▶ Play ⏸ Pause



http://reverie.herokuapp.com/arXiv_Demo/#

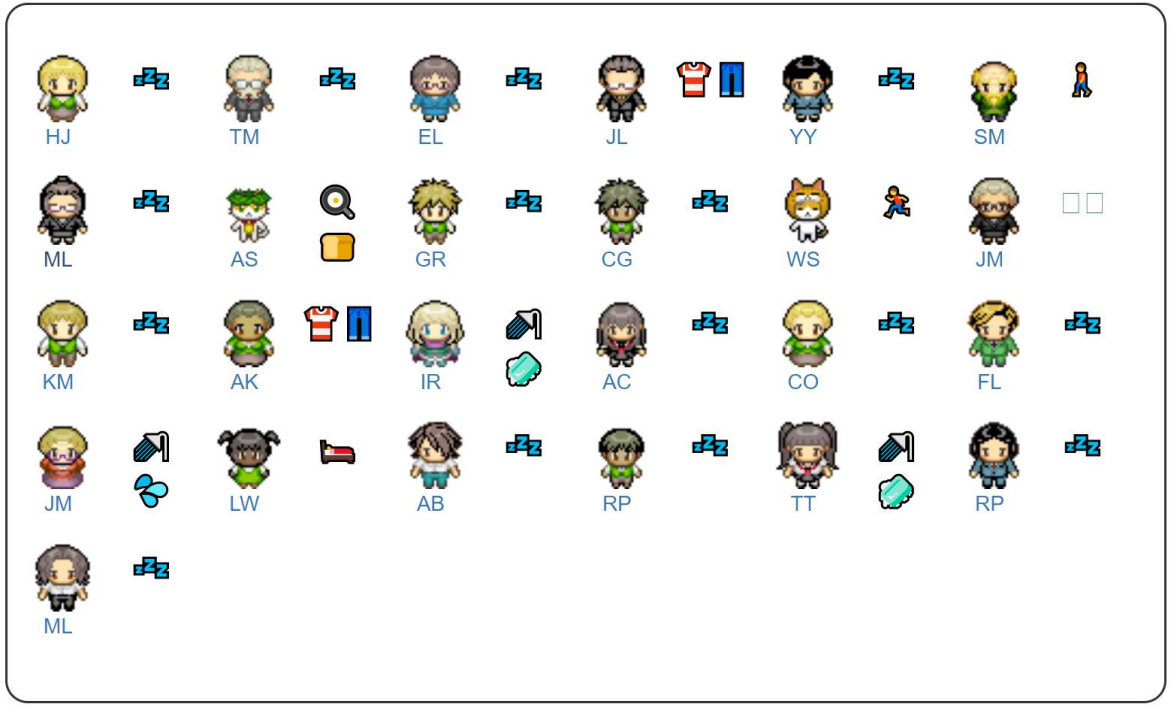
25个Agent居住在 Smallville小镇，每个Agent都可以：

- 与别人和环境交流；
- 记住并回忆它们所做的和观察到的事情；
- 反思这些观察结果；
- 制定每天的计划

Generative Agents

用自然语言设定每个Agent的角色，包括它们的职业，与其他Agent的关系

比如：John Lin 是一名药店店主，他乐于助人。他一直在寻找使客户更容易获得药物的方法。John Lin 的妻子是大学教授 Mei Lin ，它们和学习音乐理论的儿子 Eddy Lin 住在一起；John Lin 非常爱它的家人；John Lin 认识隔壁的老夫妇 Sam Moore 和 Jennifer Moore 好几年了……



在沙盒的每个Step，Agent都输出一个自然语句，描述它们当前的动作，比如：Isabella Rodriguez 正在写日记、Isabella Rodriguez 正在查看邮件等。

然后这些自然语言将转化为影响沙盒的具体动作，动作以一组表情符号显示在沙盒界面上，这些表情符号提供了动作的抽象表征。

Agent与环境交互

Smallville 小镇有许多公共场景，包括咖啡馆、酒吧、公园、学校、宿舍、房屋和商店。

每个公共场景包括自己的功能和设施，比如房子中有厨房、厨房中有炉子。在Agent的生活空间中有床、桌子、衣柜、架子、浴室、厨房等。

Agent可以在 Smallville小镇随意走动，导航前往一座建筑，或者去接近另一个Agent。Agent的移动由 Generative Agents 的架构和沙盒引擎控制（当模型指示Agent移动到某个位置时，它会计算Agent到达目的地的步行路径，然后开始移动）

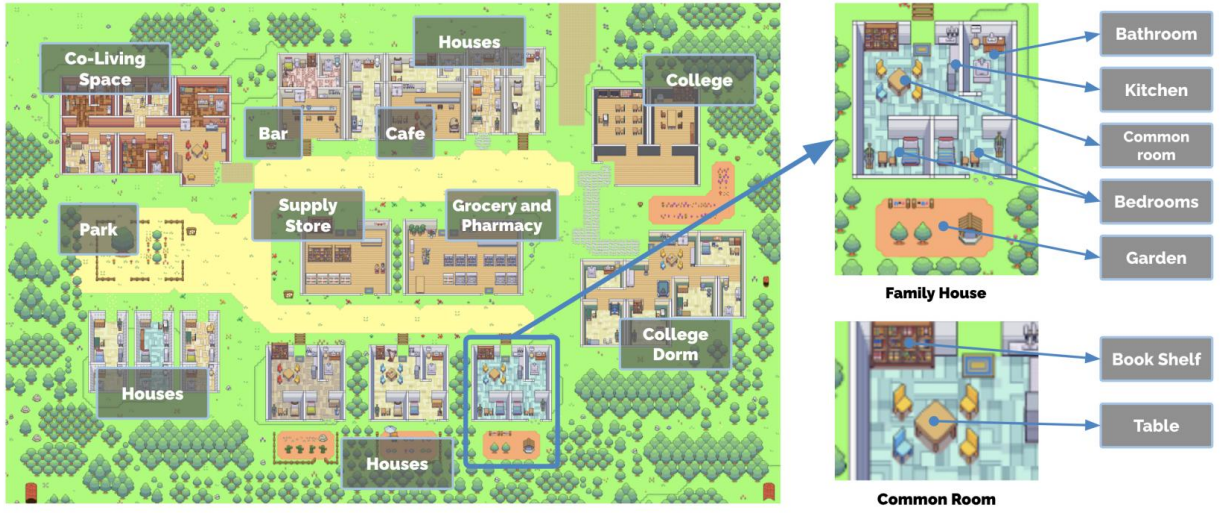


Figure 2: The Smallville sandbox world, with areas labeled. The root node describes the entire world, children describe areas (e.g., houses, cafe, stores), and leaf nodes describe objects (e.g., table, bookshelf). Agent remember a subgraph reflecting the parts of the world they have seen, in the state that they saw them.

在公共环境中，Agent会影响其他人的状态，比如：某Agent睡觉时床是被占用的，当Agent用完早餐冰箱可能是空的

Agent之间的信息传播

Generative Agents具有社交属性，信息可以在Agent之间进行传播

比如Sam 和 Tom 在杂货店中相遇，Sam 告诉了 Tom 他在竞选镇长；

在 Sam 离开后，Tom 和 John 讨论了 Sam 赢得选举的机会；

逐渐地，Sam 的竞选成为镇上的热门话题，有人支持他，有人犹豫不决。

Sam: Hey Tom, how's it going?
Tom: Good, thanks. What's up?
Sam: Well, I wanted to talk to you about something. I'm actually running for mayor in the upcoming local election.
Tom: Really? That's great news! Why are you running?
Sam: I've been involved in local politics for years now,

John: I heard that Sam Moore is running for mayor in the local election. Do you think he has a good chance of winning?
Tom: I do think he has a good chance. He's been working hard in the community and I think he will get a lot of support. What do you think?
John: I think it's great that he's running. I'm curious to see who else is running and how the election will turn out.

Agent的关系记忆

随着时间的推移，小镇上的Agent形成了新的关系，并记住了它们与其他Agent的互动。

比如Sam 一开始不认识 Latoya。

在约翰逊公园散步时，Sam 碰到了 Latoya，互相做了自我介绍，Latoya 提到自己正在进行一个摄影项目：

> Latoya：我正在这里给一个项目拍摄照片。

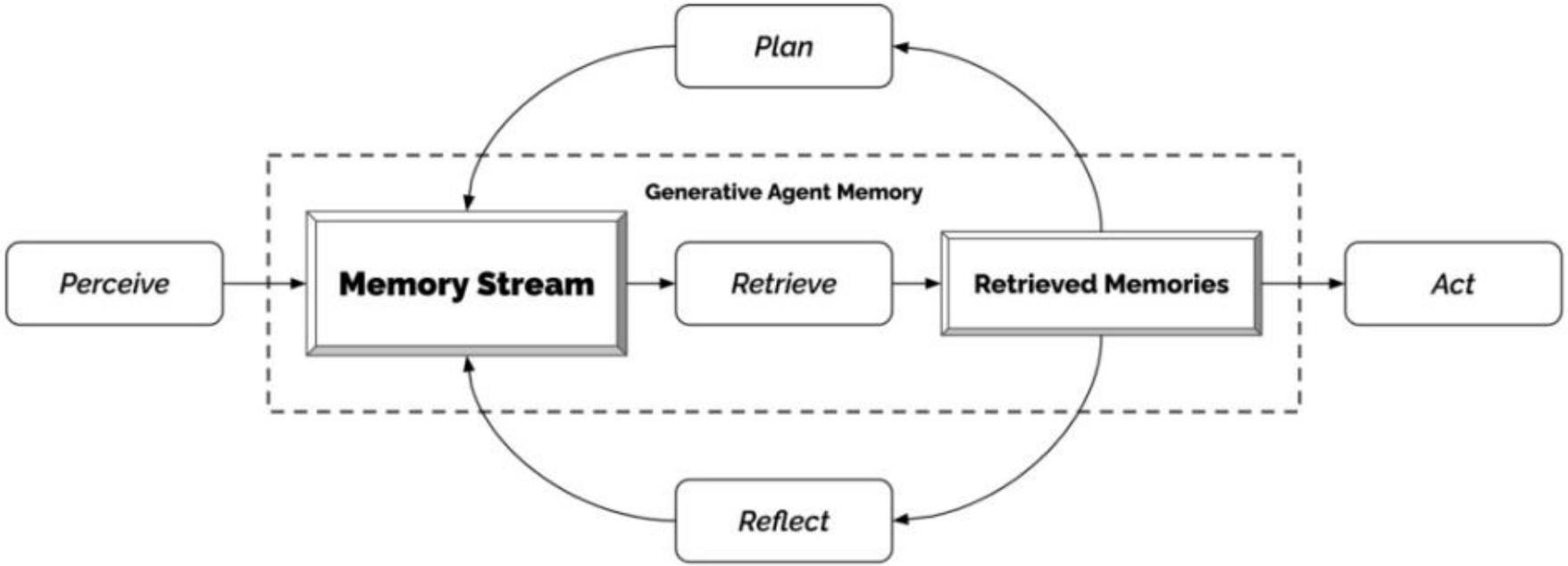
在后来他们又相遇了，Sam 与 Latoya 的对话表明了他们还记得这件事

> Sam：Latoya，你的项目进展如何?
> Latoya：进展得很好!

记忆流：

- 是Generative Agents 的核心模块，全面记录Agent经验的数据库。
- Agent会从记忆流中检索相关记录，用来规划Agent的动作，并对环境做出适当反应。每次行为都会被记录，从而生成更高级别的行为指令。
- 所有的内容都会被记录，并以自然语言进行推理，这样Agent能够使用大语言模型（GPT3.5或GPT4.0）进行推理。

Generative Agents核心模块



1）长期记忆，Agent要做多轮决策，所以要考虑更长的上下文，这些信息会超过 GPT 模型的 token 上限。

2）外部工具，使用外部工具可以让GPT调用更多的服务作为输入/输出，增强GPT的能力，这好比人会使用这些外部工具，那么GPT也可以使用

3）短期记忆，基于Attention机制将长期记忆中最相关的部分喂给GPT，生成结果

AI Agent对比

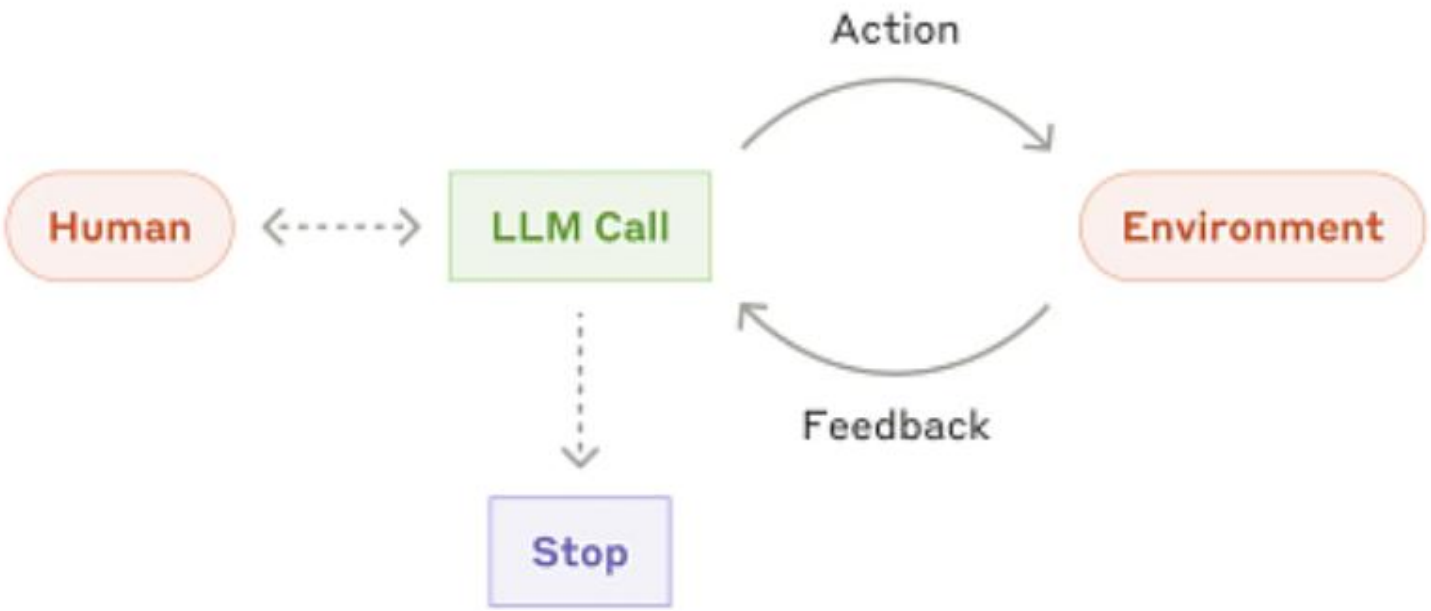
工具	核心定位	架构特点	适用场景
LangChain	开源LLM应用开发框架	基于链（Chain）的线性或分支工作流，支持Agent模式	快速构建RAG、对话系统、工具调用等线性任务
LangGraph	LangChain的扩展，专注于复杂工作流	基于图（Graph）的循环和条件逻辑，支持多Agent协作	需要循环、动态分支或状态管理的复杂任务（如自适应RAG、多Agent系统）
Qwen-Agent	通义千问的AI Agent框架	基于阿里云大模型，支持多模态交互与工具调用	开源，集成多种工具，MCP调用
Coze	字节跳动的无代码AI Bot平台	可视化拖拽界面，内置知识库、多模态插件	快速部署社交平台机器人、轻量级工作流
Dify	开源LLM应用开发平台	API优先，支持Prompt工程与灵活编排	开发者定制化LLM应用，需深度集成或私有化部署

AI Agent工具定位与对比

Agent：自主智能体的力量

Agent 通常被实现为LLM通过工具调用（基于环境反馈）在循环中执行动作的系统。

正如Anthropic指出的，Agent可以处理复杂的任务，但其实现很简单。它们通常是LLM根据环境反馈使用工具的循环。因此，清晰全面的设计工具集以及文档对于Agent的成功至关重要。



Agent：自主智能体的力量

Thinking：什么时候使用Agent?

Agent适用于那些开放性问题，这些问题很难或无法预测所需的步骤数量，并且无法硬编码固定路径。LLM可能会运行多个回合，因此你需要对其决策能力有一定的信任。

智能体的自主性使其非常适合在受信任的环境中扩展任务。然而，自主性也意味着更高的成本和可能出现的错误累积。

=> 建议在沙盒环境中进行广泛的测试，并设置适当的防护栏。

- 编程Agent可以解决涉及对多个文件进行编辑的任务。
- Computer Use 实现中，Claude通过计算机完成任务。

打造Best Practice

AI智能体和工作流是互补的，可以集成在一起以实现最佳效果，尤其是在复杂的现实世界应用中。

- 增强自动化

AI智能体可以自主处理特定任务，而工作流则将这些任务协调成一个连贯、高效的过程。

- 可扩展性

在结构化工作流中结合多个AI智能体，可以使组织高效扩展运营，减少人工工作量，提高生产力。

- 弹性与适应性

虽然单个智能体可以应对局部变化，但工作流可以动态调整整体流程，用来与战略目标保持一致。

在智能制造系统中：

- AI智能体可以监控设备性能、预测维护需求并优化生产计划。
- 工作流则负责原材料采购、生产排序、质量保证和物流，确保从原材料到产品交付的无缝过渡。

打造Best Practice

选择适合你的系统才是成功的关键：

- 在AI领域，成功并不是关于构建复杂的系统，而是构建最适合需求的系统。
- 从简单的提示开始，只有在简单方案解决步了时，再添加多步智能体系统。

在实现智能体时，有三个核心原则：

1. 保持智能体设计的简洁性：避免不必要的复杂性，专注于核心功能。
2. 优先考虑透明性：明确展示智能体的规划步骤，让用户清楚了解其决策过程。
3. 打造Function/MCP：打造工具，以及说明文档和测试，确保Agent与外部环境的交互。

框架可以帮助你快速上手，但不要害怕在进入生产阶段时减少抽象层，直接使用基础组件。遵循这些原则，你可以创建出不仅强大而且可靠、可维护且值得用户信赖的智能体系统。

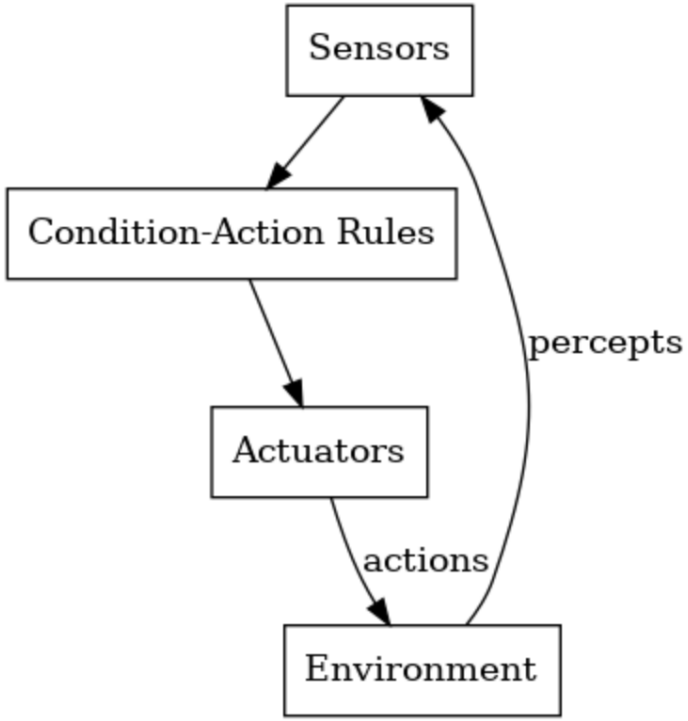
智能体分类（反应式 Reactive）

Thinking：都有哪些典型的AI Agent?

1、反应式 (Reactive)

反应式架构：快速决策的“直觉型”智能体

反应式架构是AI智能体设计中最简单直接的模式。在这种架构中，一个大型语言模型（LLM）首先分析当前情况，确定下一步要采取的行动。然后，在环境中执行该行动，产生观察结果作为反馈。LLM处理这些观察结果，重新评估下一步行动，选择另一个行动，并继续这个循环，直到任务完成。



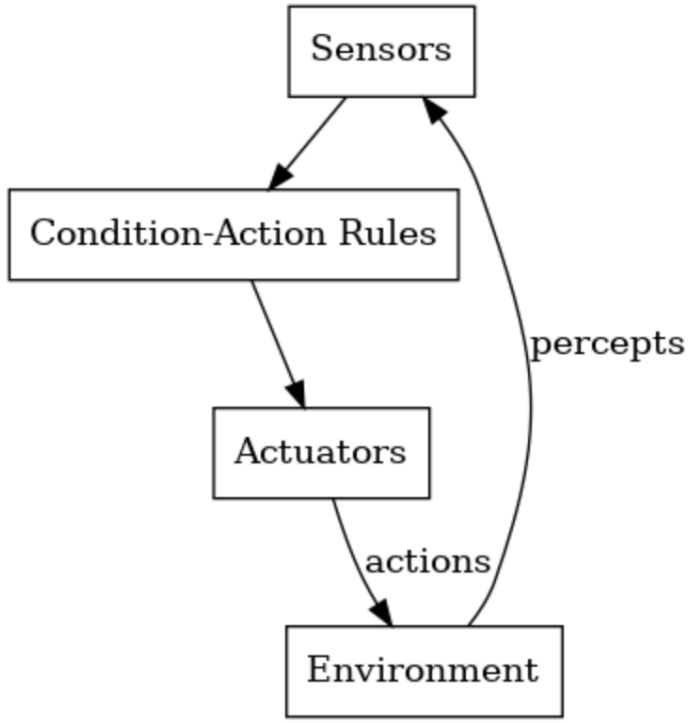
智能体分类（反应式 Reactive）

反应式架构（Reactive Architecture）

特点：基于当前环境即时决策，无长期规划，依赖预设规则快速响应。

工作原理：

- 感知：获取环境输入（如传感器数据）。
- 决策：LLM或规则系统立即生成响应动作。
- 执行：执行动作并观察结果，循环往复直至任务完成。



优势：

- 速度快：无复杂推理，适合毫秒级响应的场景（如机器人避障、高频交易）
- 简单可靠：行为由明确规则驱动，易于设计和验证。

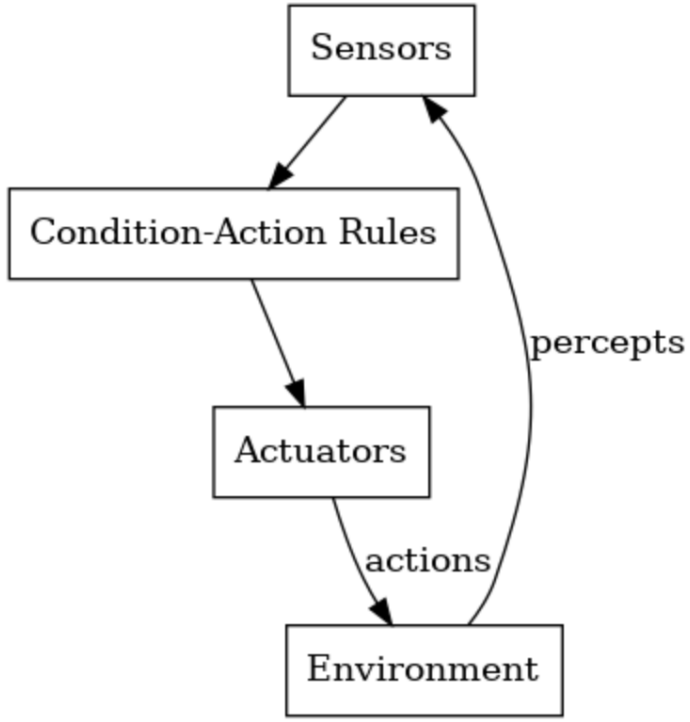
局限：

- 缺乏适应性：无法处理未预见的场景或需多步规划的任务。
- 短视性：仅优化当前动作，可能陷入局部循环（如机器人绕圈）。

智能体分类（反应式 Reactive）

典型应用：

- 机器人：扫地机器人避障、无人机紧急悬停。
- 游戏NPC：敌人对玩家攻击的即时反应。
- 工业控制：传感器超限时触发警报或停机。



适用场景：任务规则明确、响应需实时，且无需长期策略的简单环境。

“条件反射”——像膝跳反应一样快速直接，但无法应对复杂变化。

CASE：私募基金运作指引问答助手

TO DO：私募基金运作指引问答助手

采用反应式智能体架构设计，区别于传统的工作流模式，该智能体能够主动思考、自主决策并选择合适的工具来回答用户问题，具备真正的智能体特性。

1. 自主决策：智能体能够根据问题类型自主选择使用哪种工具回答问题
2. 透明思考过程：展示清晰的思考过程，使决策链路可追踪
3. 知识边界感知：明确区分知识库内容和模型知识
4. 多工具协作：集成多种工具，包括关键词搜索、类别查询和直接问答
5. 异常处理：妥善处理边界情况和超出知识范围的问题

基于LangChain的Agent框架实现，使用通义千问(Qwen-Turbo)模型作为底层大语言模型，通过工具选择、思考链路和自主决策能力，提供高质量的私募基金规则咨询服务。

CASE：私募基金运作指引问答助手

Step1. 数据准备

系统基于私募基金运作规则构建了精简的知识库(FUND_RULES_DB)，每条规则包含：

- ID：规则唯一标识
- 类别：规则所属类别(如"设立与募集"、"监管规定"等)
- 问题：规则对应的常见问题
- 答案：规则的详细内容


```
{
   "id": "rule001",
   "category": "设立与募集",
   "question": "私募基金的合格投资者标准是什么？",
   "answer": "合格投资者是指具备相应风险识别能力和风险承担能力，投资于单只私募基金的金额不低于100万元且符合下列条件之一的单位和个人：\n1. 净资产不低于1000万元的单位\n2. 金融资产不低于300万元或者最近三年个人年均收入不低于50万元的个人"
},
……
```

CASE：私募基金运作指引问答助手

Step2. 工具设计

系统实现了三种核心工具：

1. 关键词搜索工具：通过关键词匹配检索相关规则，search_rules_by_keywords

2. 类别查询工具：根据规则类别查询相关规则，search_rules_by_category

3. 直接回答工具：基于问题与规则的匹配度直接回答用户问题，answer_question

CASE：私募基金运作指引问答助手

Step3. Agent架构搭建

基于LangChain的Agent框架，构建了完整的反应式智能体系统：

1. 自定义提示模板(CustomPromptTemplate)：定义Agent的思考和决策格式

2. 自定义输出解析器(CustomOutputParser)：解析LLM的输出，确定下一步行动

3. Agent执行器(AgentExecutor)：协调智能体与工具的交互

CASE：私募基金运作指引问答助手

Step4. 知识边界处理

专门设计了处理超出知识库范围问题的机制：

1. 问题主题识别：识别用户问题涉及的具体主题

2. 明确边界区分：在回答中明确区分知识库内容和模型知识

3. 提供有价值建议：引导用户寻求更权威的信息来源

CASE：私募基金运作指引问答助手

私募基金的合格投资者标准是什么?

```
> Entering new AgentExecutor chain...
---
Question: 私募基金的合格投资者标准是什么?
Thought: 我需要查询私募基金规则中关于合格投资者的标准。
Action: 关键词搜索
Action Input: 合格投资者 标准

Observation:类别: 设立与募集
问题: 私募基金的合格投资者标准是什么?
答案: 合格投资者是指具备相应风险识别能力和风险承担能力，投资于单只私募基金的金额不低于100万元且符合下列条件之一的单位和个人:
1. 净资产不低于1000万元的单位
2. 金融资产不低于300万元或者最近三年个人年均收入不低于50万元的个人
我现在知道答案了。
Final Answer: 私募基金的合格投资者是指具备相应风险识别能力和风险承担能力，投资于单只私募基金的金额不低于100万元且符合下列条件之一的单位和个人:
1. 净资产不低于1000万元的单位;
2. 金融资产不低于300万元或者最近三年个人年均收入不低于50万元的个人。

> Finished chain.
回答: 私募基金的合格投资者是指具备相应风险识别能力和风险承担能力，投资于单只私募基金的金额不低于100万元且符合下列条件之一的单位和个人:
1. 净资产不低于1000万元的单位;
2. 金融资产不低于300万元或者最近三年个人年均收入不低于50万元的个人。
```

CASE：私募基金运作指引问答助手

如果我是一个净资产800万元的机构，我能投资私募基金吗?

```
> Entering new AgentExecutor chain...
---
Question: 如果我是一个净资产800万元的机构，我能投资私募基金吗?
Thought: 我需要了解关于机构投资者资格的相关规则。
Action: 类别查询
Action Input: 机构投资者资格

Observation:未找到类别为 '机构投资者资格' 的规则。
对不起，在我的知识库中没有关于机构投资者资格的详细信息。一般来说，私募基金对投资者的资质有一定的要求，包括资金实力、投资经验等。净资产达到800万元的机构可能会符合某些私募基金的投资门槛，但具体还需要参考相关的法律法规和私募基金的具体规定。

建议您查阅最新的私募基金相关法律法规或咨询专业的金融顾问以获取准确的信息。

Final Answer: 根据现有信息无法确定，建议查阅最新法律法规或咨询专业金融顾问。

> Finished chain.
回答：根据现有信息无法确定，建议查阅最新法律法规或咨询专业金融顾问。
```

这里Agent自主选择了 工具2：根据规则类别查询search_rules_by_category，没有找到知识

CASE：私募基金运作指引问答助手

如果我是一个净资产800万元的机构，我能投资私募基金吗?

```
> Entering new AgentExecutor chain...
---
Thought: 我需要确认机构投资者的资格要求，以判断净资产800万元是否满足条件。
Action: 关键词搜索
Action Input: 机构投资者 私募基金 净资产 资格要求

Observation:类别: 设立与募集
问题: 私募基金的合格投资者标准是什么?
答案: 合格投资者是指具备相应风险识别能力和风险承担能力，投资于单只私募基金的金额不低于100万元且符合下列条件之一的单位和个人:
1. 净资产不低于1000万元的单位
2. 金融资产不低于300万元或者最近三年个人年均收入不低于50万元的个人

类别: 设立与募集
问题: 私募基金的最低募集规模要求是多少?
答案: 私募证券投资基金的最低募集规模不得低于人民币1000万元。对于私募股权基金、创业投资基金等其他类型的私募基金，监管规定更加灵活，通常需符合基金合同的约定。
我现在知道答案了。

Final Answer: 根据私募基金的合格投资者标准，机构投资者需要净资产不低于1000万元。因此，作为净资产为800万元的机构，您目前不符合投资私募基金的资格要求。请注意，除了净资产要求外，投资于单只私募基金的金额还不得低于100万元。建议您咨询专业法律顾问或等待净资产达到要求后再进行投资。

> Finished chain.
回答：根据私募基金的合格投资者标准，机构投资者需要净资产不低于1000万元。因此，作为净资产为800万元的机构，您目前不符合投资私募基金的资格要求。请注意，除了净资产要求外，投资于单只私募基金的金额还不得低于100万元。建议您咨询专业法律顾问或等待净资产达到要求后再进行投资。
```

这里Agent自主选择了 工具1：通过关键词搜索相关规则 search_rules_by_keywords，找到了知识

CASE：私募基金运作指引问答助手

私募基金可以投资哪些资产类别?

```
> Entering new AgentExecutor chain...
Thought: 用户询问私募基金可以投资的资产类别，我可以通过类别查询来获取相关信息。
Action: 类别查询
Action Input: 私募基金可投资资产类别

Observation:未找到类别为 '私募基金可投资资产类别' 的规则。
Direct response detected: 对不起，在我的知识库中没有关于私募基金可投资资产类别的详细信息。根据我的经验，私募基金通常可以投资于股票、债券、房地产、大宗商品、对冲基金和其他另类资产等。但具体的投资范围可能会受到当地法律法规和基金合同的限制。建议您查阅相关法规或基金合同以获取准确的信息。

> Finished chain.
回答：对不起，在我的知识库中没有关于私募基金可投资资产类别的详细信息。根据我的经验，私募基金通常可以投资于股票、债券、房地产、大宗商品、对冲基金和其他另类资产等。但具体的投资范围可能会受到当地法律法规和基金合同的限制。建议您查阅相关法规或基金合同以获取准确的信息。
```

打卡：反应式 Reactive

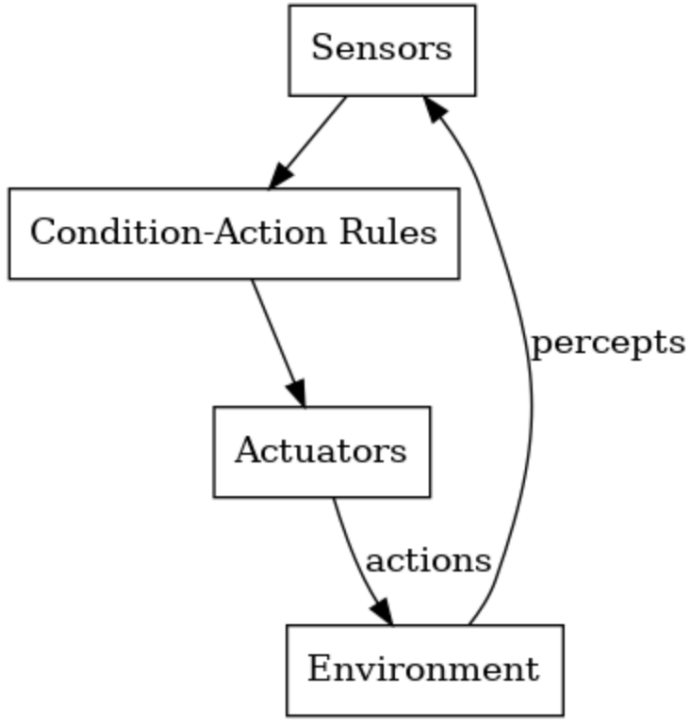
结合自己的业务场景，搭建Reactive Agent

Step1，数据准备

Step2，工具设计

Step3，Agent搭建（提示词模版，输出解析器，Agent执行）

Step4，知识边界处理



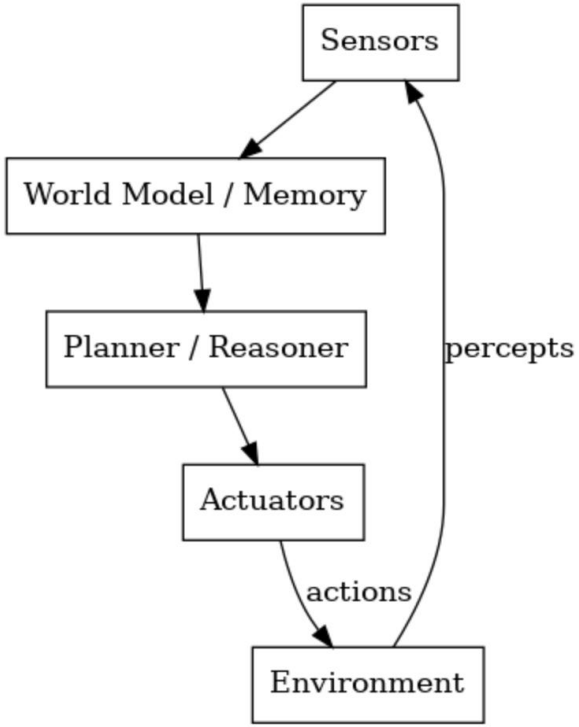
智能体分类（深思熟虑 Deliberative）

深思熟虑智能体（Deliberative Agent）

特点：基于内部模型进行规划，通过推理选择最优行动方案，具有长期目标导向性。

核心流程：

- 感知：获取环境信息
- 建模：更新内部世界状态表示
- 推理：生成候选计划并模拟结果
- 决策：选择最优方案执行



优势：

✓ 能处理多步复杂任务

✓ 优化长期目标而非即时反馈

✓ 适应动态变化环境

智能体分类（深思熟虑 Deliberative）

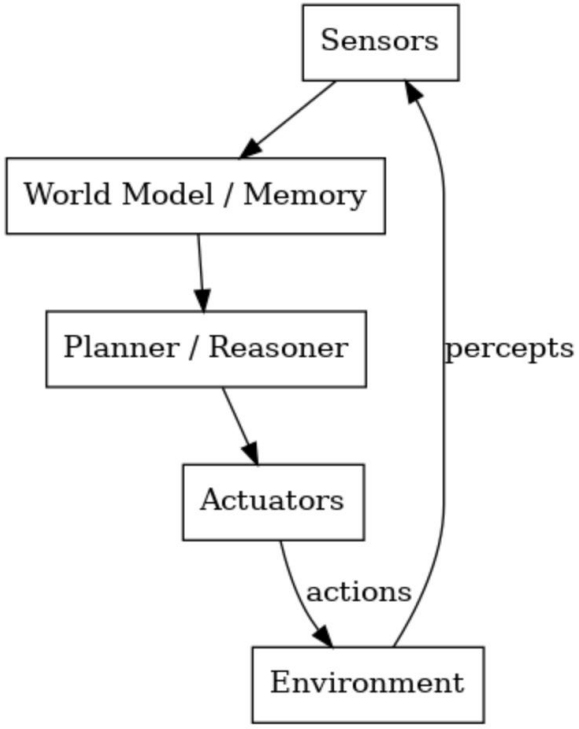
典型示例：

路径规划智能体：

- 生成多条候选路线
- 评估安全性/耗时等指标
- 选择最优路径执行

适用场景：

需战略规划的任务（如物流调度、投资决策等）



像下棋高手——每步棋都经过推演，而非凭直觉反应

CASE：智能投研助手

TO DO：智能投研助手

为投资研究场景设计，基于LangGraph实现的深思熟虑型智能体。

该智能体能够整合市场数据，进行多步骤分析和推理，生成投资观点和研究报告。

与反应式智能体不同，深思熟虑型智能体通过内部建模、推理和规划，制定最优行动方案，特别适合需要多步骤思考和长期目标优化的任务。

1. 内部建模：构建市场模型和行业理解
2. 多方案生成：基于市场模型创建多个候选投资策略
3. 方案评估：对比分析各方案优劣，考虑多维度因素
4. 长期规划：优化整体投资回报，而非单点决策
5. 推理透明：能够解释决策过程和选择理由

CASE：智能投研助手（步骤）

具体实现步骤：

Step1，感知阶段

收集和整理市场数据与信息，包括：

- 市场概况和最新动态
- 关键经济和市场指标
- 近期重要新闻
- 行业趋势分析

Step2. 建模阶段

基于收集的数据构建内部世界模型，分析：

- 当前市场状态评估
- 经济周期判断
- 主要风险因素
- 潜在机会领域
- 市场情绪分析

CASE：智能投研助手（步骤）

Step3. 推理阶段

生成多个候选投资分析方案，每个方案包含：

- 投资假设
- 分析方法
- 预期结果
- 置信度
- 方案优缺点

Step4. 决策阶段

评估候选方案并选择最优投资观点，形成：

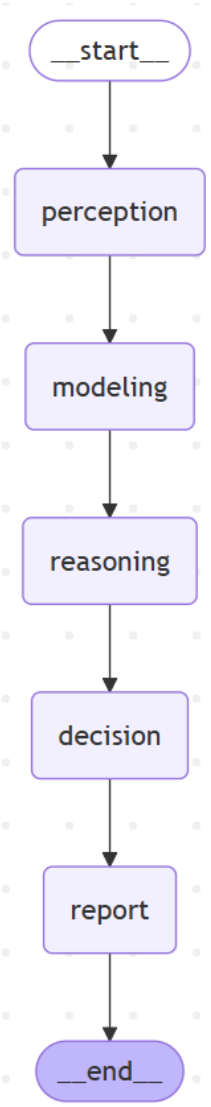
- 投资论点
- 支持证据
- 风险评估
- 投资建议
- 时间框架

CASE：智能投研助手（步骤）

Step5. 报告阶段

生成完整的投资研究报告，包含：

- 报告标题和摘要
- 市场和行业背景
- 核心投资观点
- 详细分析论证
- 风险因素
- 投资建议
- 时间框架和预期回报



CASE：智能投研助手（状态定义）

```
class ResearchAgentState(TypedDict):
    # 输入
    research_topic: str  # 研究主题
    industry_focus: str  # 行业焦点
    time_horizon: str  # 时间范围(短期/中期/长期)

    # 处理状态
    perception_data: Optional[Dict[str, Any]]  # 感知阶段收集的数据
    world_model: Optional[Dict[str, Any]]  # 内部世界模型
    reasoning_plans: Optional[List[Dict[str, Any]]]  # 候选分析方案
    selected_plan: Optional[Dict[str, Any]]  # 选中的最优方案

    # 输出
    final_report: Optional[str]  # 最终研究报告

    # 控制流
    current_phase: Literal["perception", "modeling", "reasoning", "decision", "report"]
    error: Optional[str]  # 错误信息
```

使用`ResearchAgentState`类型定义智能体状态

CASE：智能投研助手（输出模型）

```
class PerceptionOutput(BaseModel):
    """感知阶段输出的市场数据和信息"""
    market_overview: str
    key_indicators: Dict[str, str]
    recent_news: List[str]
    industry_trends: Dict[str, str]

class ModelingOutput(BaseModel):
    """建模阶段输出的内部世界模型"""
    market_state: str
    economic_cycle: str
    risk_factors: List[str]
    opportunity_areas: List[str]
    market_sentiment: str
```

使用Pydantic模型严格定义各阶段输出，
确保结构化数据

```
class ReasoningPlan(BaseModel):
    """推理阶段生成的候选分析方案"""
    plan_id: str
    hypothesis: str
    analysis_approach: str
    expected_outcome: str
    confidence_level: float
    pros: List[str]
    cons: List[str]

class DecisionOutput(BaseModel):
    """决策阶段选择的最优投资观点"""
    selected_plan_id: str
    investment_thesis: str
    supporting_evidence: List[str]
    risk_assessment: str
    recommendation: str
    timeframe: str
```

CASE：智能投研助手（工作流实现）

```
def create_research_agent_workflow() -> StateGraph:
    # 创建状态图
    workflow = StateGraph(ResearchAgentState)

    # 添加节点
    workflow.add_node("perception", perception)
    workflow.add_node("modeling", modeling)
    workflow.add_node("reasoning", reasoning)
    workflow.add_node("decision", decision)
    workflow.add_node("report", report_generation)

    # 设置入口点
    workflow.set_entry_point("perception")

    # 设置边和转换条件
    workflow.add_edge("perception", "modeling")
    workflow.add_edge("modeling", "reasoning")
    workflow.add_edge("reasoning", "decision")
    workflow.add_edge("decision", "report")
    workflow.add_edge("report", END)

    # 编译工作流
    return workflow.compile()
```

核心工作流通过LangGraph的`StateGraph`实现，将各阶段连接成一个有向图

CASE：智能投研助手（各阶段处理逻辑）

每个阶段都遵循类似的处理模式：

1. 检查前置条件（上一阶段数据）
2. 准备LLM提示
3. 调用LLM进行推理
4. 解析结果并更新状态
5. 错误处理与恢复机制

例如，推理阶段实现：

```
def reasoning(state: ResearchAgentState) -> ResearchAgentState:
    try:
        # 检查前置条件
        if not state.get("world_model"):
            return {
                **state,
                "error": "推理阶段缺少世界模型",
                "current_phase": "modeling"  # 回到建模阶段
            }

        # 准备提示
        prompt =
ChatPromptTemplate.from_template(REASONING_PROMPT)

        # 构建输入
        input_data = {
            "research_topic": state["research_topic"],
            "industry_focus": state["industry_focus"],
            "time_horizon": state["time_horizon"],
```

CASE：智能投研助手（各阶段处理逻辑）

```
 "world_model": json.dumps(state["world_model"],
ensure_ascii=False, indent=2)
        }
        # 调用LLM
        chain = prompt | llm | JsonOutputParser()
        result = chain.invoke(input_data)

        # 更新状态
        return {
            **state,
            "reasoning_plans": result,
            "current_phase": "decision"
        }
    except Exception as e:
        return {
            **state,
            "error": f"推理阶段出错: {str(e)}",
            "current_phase": "reasoning"  # 保持在当前阶段
        }
```

CASE：智能投研助手（使用方法）

```
if __name__ == "__main__":
    print("=== 深思熟虑智能体 - 智能投研助手 ===\n")

    # 用户输入
    topic = input("请输入研究主题 (例如: 新能源汽车行业投资机会): ")
    industry = input("请输入行业焦点 (例如: 电动汽车制造、电池技术): ")
    horizon = input("请输入时间范围 [短期/中期/长期]: ")

    print("\n智能投研助手开始工作...\n")

    # 运行智能体
    result = run_research_agent(topic, industry, horizon)

    # 处理结果
    if result.get("error"):
        print(f"\n发生错误: {result['error']}")
    else:
        print("\n=== 最终研究报告 ===\n")
        print(result.get("final_report", "未生成报告"))
```

请输入研究主题 (例如: 新能源汽车行业投资机会): 新能源汽车行业投资机会

请输入行业焦点 (例如: 电动汽车制造、电池技术): 电动汽车制造、电池技术

请输入时间范围 [短期/中期/长期]: 中期

CASE：智能投研助手（生成报告）

新能源汽车行业投资研究报告：中端电动汽车市场的中期增长机会

报告摘要

新能源汽车行业正处于快速发展的阶段，全球政策支持、技术进步以及消费者环保意识的增强共同推动了电动汽车（EV）和电池技术的需求增长。本报告聚焦于中端电动汽车市场及下一代电池技术的投资机会，结合市场数据、行业趋势和风险评估，提出以成本优化和规模化生产为核心的投资策略。我们建议优先投资具备较强成本管理和规模效应的企业，并关注充电基础设施建设带来的协同效应。

---

1. 市场和行业背景

1.1 行业现状

新能源汽车行业目前处于快速增长阶段，市场需求强劲，技术创新活跃。根据最新市场数据：

- **全球电动汽车销量同比增长率**：2022年达到42%，显示出强劲的市场需求。
- **锂电池成本**：降至$107/kWh（2022年），持续下降的成本推动了电动汽车价格竞争力的提升。
- **政府补贴**：平均每辆电动车补贴在$5,000-$8,000之间，加速了电动车的普及。

1.2 行业驱动因素

- **政策支持**：各国政府通过购车补贴、税收优惠和排放法规等措施支持电动车发展。
- **技术进步**：固态电池、快充技术和模块化平台的研发显著提升了电动汽车的性能和生产效率。
- **消费者需求**：环保意识增强和电动车性价比提升推动了市场需求的增长。

CASE：智能投研助手（生成报告）

1.3 近期动态

- 特斯拉宣布将在墨西哥建立新工厂，专注于生产下一代电动汽车平台。
- 宁德时代与宝马达成战略合作协议，共同开发新一代电池技术。
- 美国能源部预测，到2030年美国电动车充电基础设施投资需求将达到400亿美元。

---

2. 核心投资观点

中端电动汽车市场将成为未来3-5年销量增长的核心驱动力。随着生产效率的提升和成本的下降，中端市场将受益于规模经济效应和技术进步。同时，下一代电池技术（如固态电池和快充技术）的突破将进一步推动行业的利润增长。

---

3. 详细分析论证

3.1 中端电动汽车市场的潜力

- **市场规模**：中端市场具有相对稳定的消费者基础和庞大的潜在需求。随着电动车价格的下降，更多消费者能够负担得起中端车型。
- **成本优化**：模块化平台的应用显著提高了生产效率，降低了单位制造成本。
- **政策支持**：政府补贴和税收优惠政策进一步增强了中端电动车的价格竞争力。

3.2 下一代电池技术的发展

- **高能量密度**：当前主流锂离子电池正在向更高能量密度方向发展，而固态电池被认为是下一代技术的关键。
- **快充技术**：超级快充站的普及将显著改善用户体验，减少充电时间。
- **原材料回收**：电池回收利用技术的进步有助于缓解锂、钴等关键材料的供应压力。

CASE：智能投研助手（生成报告）

3.3 充电基础设施建设
- **全球范围扩展**：为满足快速增长的电动车保有量，全球范围内的充电网络建设正在加速。
- **新兴市场机会**：新兴市场对充电桩的需求尚未完全释放，提供了巨大的投资机会。
- **技术升级**：家庭充电桩和超级快充站的普及将进一步提升用户便利性。

---

4. 风险因素
尽管新能源汽车行业前景广阔，但仍需关注以下风险因素：
1. **原材料价格波动**：锂、钴等关键电池材料的价格波动可能对成本控制构成压力。
2. **技术迭代风险**：固态电池等新技术的研发进度可能不及预期，影响现有技术的市场竞争力。
3. **地缘政治不确定性**：供应链全球化可能导致受国际贸易摩擦或区域性政策变化的影响。
4. **市场竞争加剧**：随着越来越多的企业进入市场，竞争可能加剧，导致利润率下降。

---

5. 投资建议
5.1 投资目标
优先投资于中端电动汽车市场的领先企业，特别是那些能够通过规模经济和技术优势实现成本优化的公司。
5.2 投资逻辑
- **市场内部模型支持**：新能源汽车行业处于快速增长阶段，中端市场的扩展将受益于生产效率提升和成本下降。
- **政策和技术双重驱动**：政策支持和技术进步有助于降低中端电动汽车的成本，增强产品竞争力。
- **稳定且庞大的市场需求**：中端市场适合大规模资金投入，具备较高的回报潜力。

CASE：智能投研助手（生成报告）

5.3 监控指标
- **原材料价格**：密切关注锂、钴等关键材料的价格波动。
- **市场竞争动态**：跟踪主要竞争对手的技术进展和市场份额变化。
- **政策变化**：评估各国政府对电动车的支持力度是否发生变化。

---

6. 时间框架和预期回报
6.1 时间框架
- **中期（3-5年）**：重点关注中端电动汽车市场的增长和下一代电池技术的商业化进程。
6.2 预期回报
- **市场规模扩张**：预计全球中端电动汽车销量将以每年30%-40%的速度增长。
- **成本下降**：锂电池成本有望进一步下降至$80/kWh以下，推动电动车价格竞争力提升。
- **投资回报**：基于市场增长和技术进步，预计投资年化回报率可达15%-20%。

---

7. 结论
新能源汽车行业正处于快速发展阶段，中端电动汽车市场和下一代电池技术是未来3-5年的核心投资机会。通过选择具备成本管理和规模效应的企业，并关注充电基础设施建设的协同效应，投资者可以在行业中长期发展中获得可观回报。同时，需密切监控原材料价格波动、技术迭代风险和市场竞争动态，以及时调整投资策略。

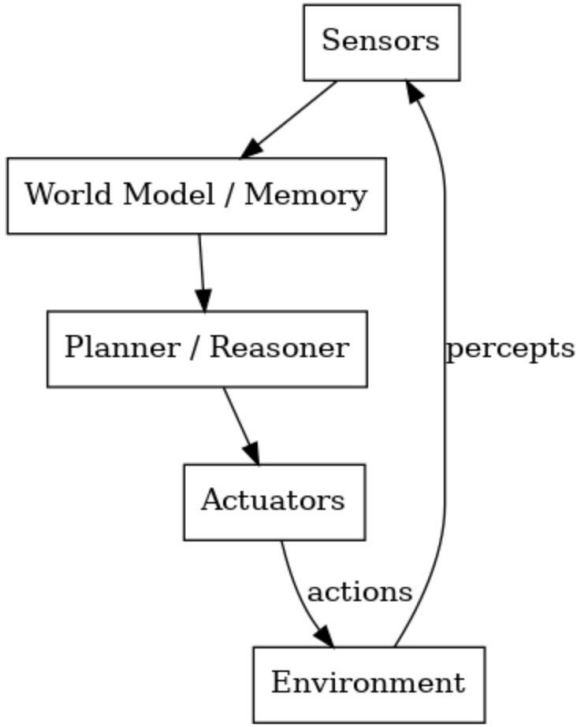
research_report_20250509_230702.txt

打卡：深思熟虑式 Agent

以智能投研助手（生成报告）为例，搭建Deliberative Agent

为投资研究场景设计，基于LangGraph实现。

能够整合市场数据，进行多步骤分析和推理，生成投资观点和研究报告。



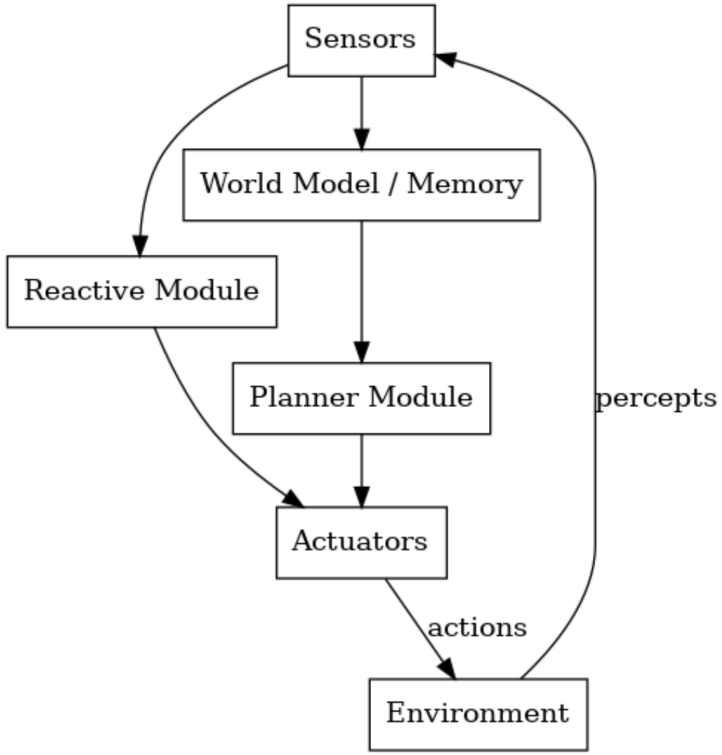
智能体分类（混合 Hybrid）

混合智能体架构（Hybrid Agent Architecture）

特点：结合反应式的"快速本能"和深思熟虑的"战略规划"，实现智能与效率的平衡。

三层设计：

- 底层（反应式）：即时处理紧急任务（如避障）
- 中层（协调）：管理任务优先级（可选）
- 顶层（深思熟虑）：进行长期目标规划（如路径优化）



运作机制：

通过仲裁系统（如监督器）动态切换模式：

- 紧急情况 → 启用反应式快速响应
- 常规情况 → 启动深思熟虑规划

智能体分类（混合 Hybrid）

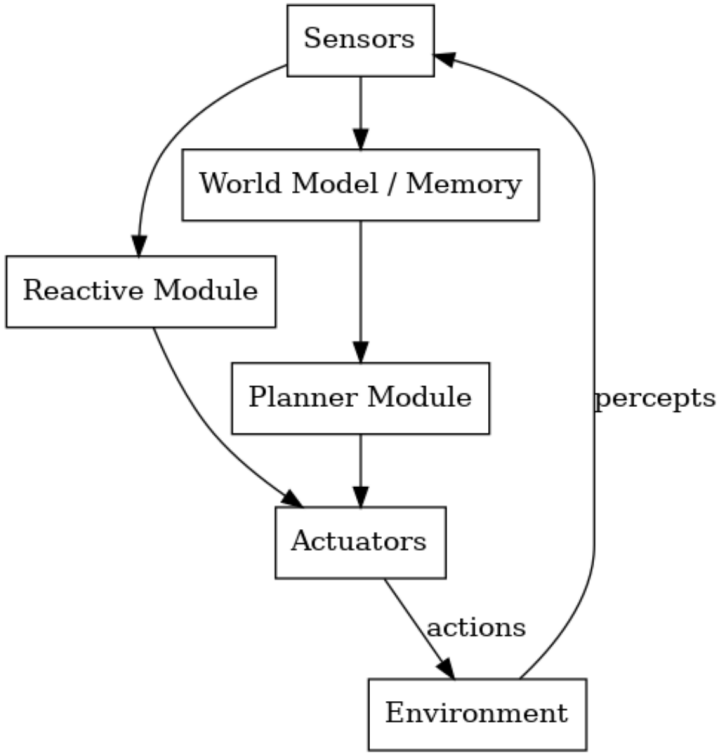
典型示例：

自动驾驶车辆：

- 突发障碍 → 立即刹车（反应式）
- 正常行驶 → 规划最优路线（深思熟虑）

核心优势：

- 兼具实时响应能力（毫秒级）
- 保留战略规划优势（长期目标）



CASE：投顾AI助手

TO DO：投顾AI助手

基于混合智能体架构设计，结合了反应式架构的即时响应能力和深思熟虑架构的长期规划能力，通过协调层动态选择最合适的处理模式，为客户提供智能化、个性化的财富管理咨询服务。

混合智能体采用三层架构设计：

1. 底层（反应式层）：

- 处理简单直接的查询，如市场状况、账户信息等
- 毫秒级响应速度，提供快速反馈
- 基于预设规则和简单逻辑作出决策

2. 中层（协调层）：

- 评估任务类型和优先级
- 动态选择处理模式（反应式或深思熟虑）
- 管理系统资源分配

3. 顶层（深思熟虑层）：

- 处理复杂的投资分析和长期财务规划
- 多步骤、深度思考过程
- 构建内部模型并生成多个备选方案

CASE：投顾AI助手（处理流程）

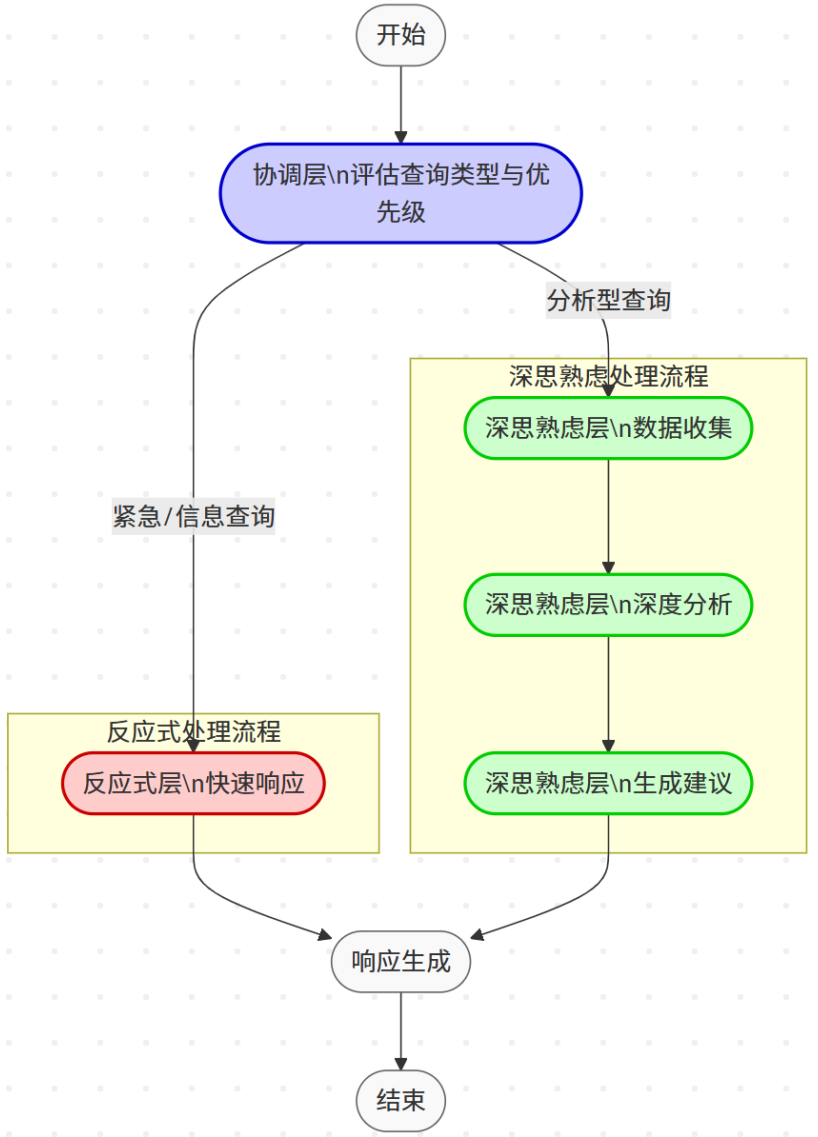
Step1. 查询评估阶段

协调层首先评估客户查询，确定：

- 查询类型：紧急型、信息型或分析型
- 处理模式：反应式或深思熟虑

评估基于以下因素：

- 查询的复杂性和时效性
- 所需数据和计算资源
- 客户期望的响应时间



CASE：投顾AI助手（处理流程）

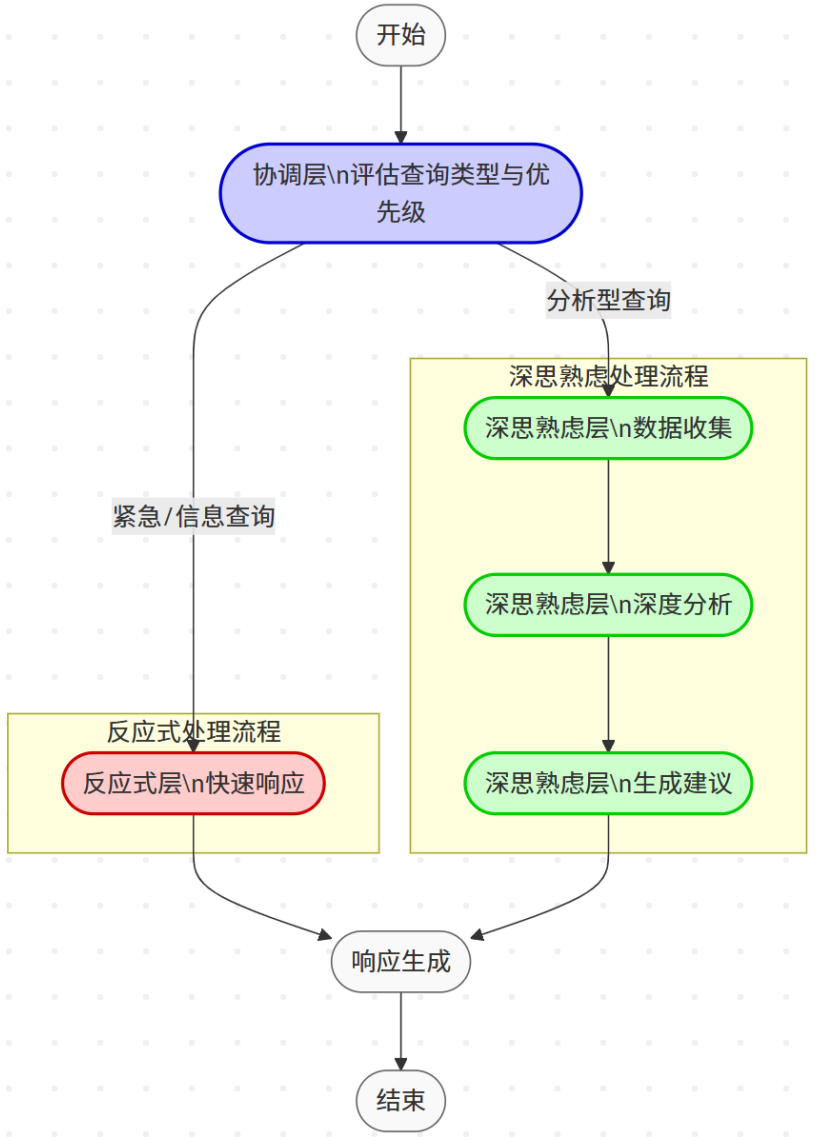
Step2A. 反应式处理流程

针对简单直接的查询：

- 市场状况查询（如"今天上证指数表现如何"）
- 账户信息查询（如"我的投资组合中科技股占比"）
- 基础概念解释（如"什么是ETF"）

处理特点：

- 低延迟、高响应速度
- 直接调用数据和预设回答
- 简洁明了的输出格式



CASE：投顾AI助手（处理流程）

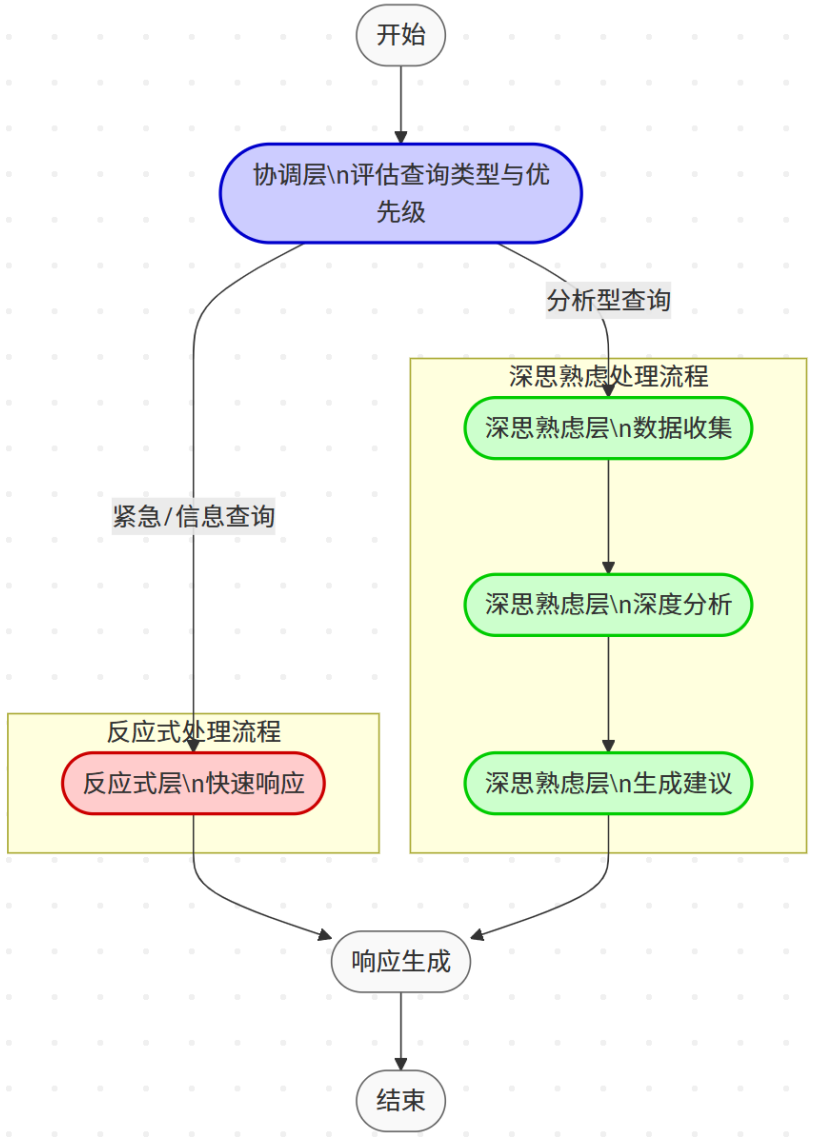
Step2B. 深思熟虑处理流程

针对复杂分析型查询：

- 投资组合调整建议
- 长期财务规划
- 风险评估和应对策略

处理步骤：

1. 数据收集：整合市场数据、客户画像、历史数据等
2. 深度分析：构建市场模型，分析多种可能情景
3. 生成建议：形成多个方案并选择最优解



CASE：投顾AI助手（状态管理）

智能体通过WealthAdvisorState维护完整状态：

```
class WealthAdvisorState(TypedDict):
    # 输入
    user_query: str  # 用户查询
    customer_profile: Optional[Dict[str, Any]]  # 客户画像

    # 处理状态
    query_type: Optional[Literal["emergency", "informational", "analytical"]]
# 查询类型
    processing_mode: Optional[Literal["reactive", "deliberative"]]  # 处理模式
    emergency_response: Optional[Dict[str, Any]]  # 紧急响应结果
    market_data: Optional[Dict[str, Any]]  # 市场数据
    analysis_results: Optional[Dict[str, Any]]  # 分析结果

    # 输出
    final_response: Optional[str]  # 最终响应
    # 控制流
    current_phase: Literal["assess", "reactive",
"collect_data", "analyze", "recommend", "respond"]
    error: Optional[str]  # 错误信息
```

CASE：投顾AI助手（使用场景-反应式）

场景1：市场信息查询（反应式处理）

今天上证指数的表现如何?

用户查询: 今天上证指数的表现如何?
选择客户: 平衡型 投资者

正在处理...

[DEBUG] 进入节点: assess_query
[DEBUG] LLM评估输出: {'query_type': 'emergency', 'processing_mode': 'reactive', 'reasoning': "用户的查询是关于当天上证指数的表现，这是一个需要实时市场数据的紧急查询。用户期望快速获取最新的市场信息，因此需要立即响应。这种类型的查询适合采用'reactive'处理模式，以便迅速提供准确的市场数据。"}
[DEBUG] 分支判断: processing_mode=reactive, query_type=emergency
[DEBUG] 进入节点: reactive_processing
【处理模式: 反应式】- 快速响应简单查询

=== 响应结果 ===

我需要查询今天的上证指数表现。让我查询一下最新的数据...根据最新数据，今天上证指数收于3200.12点，上涨了15.67点，涨幅为0.49%。请注意，这是截至今天的市场数据，股市有风险，投资需谨慎。如果您需要更详细的信息，可以关注证券市场的实时报道或者财经新闻。

处理用时: 11.34秒

CASE：投顾AI助手（使用场景-深思熟虑）

场景2：投资组合优化（深思熟虑处理）

根据当前市场情况，我应该如何调整投资组合以应对可能的经济衰退?

用户查询: 根据当前市场情况，我应该如何调整投资组合以应对可能的经济衰退?
选择客户: 平衡型 投资者

正在处理...

[DEBUG] 进入节点: assess_query
[DEBUG] LLM评估输出: {'query_type': 'analytical', 'processing_mode': 'deliberative', 'reasoning': '用户的查询涉及根据当前市场情况调整投资组合以应对经济衰退，这需要深入分析市场趋势、经济指标以及用户的投资目标和风险承受能力。因此，这是一个需要深度思考和分析的查询，适合采用 deliberative 处理模式。'}
[DEBUG] 分支判断: processing_mode=deliberative, query_type=analytical
[DEBUG] 进入节点: collect_data
[DEBUG] 进入节点: analyze_data
[DEBUG] 进入节点: generate_recommendations
【处理模式: 深思熟虑】- 深度分析复杂查询

=== 响应结果 ===

.....

CASE：投顾AI助手（使用场景-深思熟虑）

尊敬的客户，

感谢您选择我们的财富管理服务。根据当前市场环境和您的个人情况，我们为您制定了以下详细的投资建议，帮助您更好地应对可能的经济衰退，同时实现中期财务目标（如退休规划和子女教育金）。以下是具体的内容：

---

1. 总体投资策略

在当前经济可能进入衰退的情况下，我们需要采取一种“防御性增长”的策略。这意味着在保护现有资本的同时，继续寻找具备长期潜力的投资机会。我们将通过优化资产配置、降低波动性和定期再平衡来实现这一目标。

由于您是“平衡型”风险承受能力的投资者，新的投资组合将更加注重收益与稳定性的平衡。调整后，您可以更安心地面对市场波动，同时为未来的财务目标打下坚实基础。

---

2. 具体行动步骤

以下是具体的调整建议，分为四个主要类别：股票、债券、现金和另类投资。

(1) 股票部分

- **调整比例**：从40%降至30%。
- **原因**：股票市场的波动性较高，尤其是在经济衰退期间。减少股票配置可以降低整体风险。

CASE：投顾AI助手（使用场景-深思熟虑）

- **选择方向**：
  - 优先考虑具有防御属性的行业，例如医疗保健、消费品和可再生能源领域。
  - 继续关注ESG投资，但要确保标的具备良好的基本面和较低估值。
  - 在科技行业中，选择与人工智能、云计算等高成长领域相关的优质公司，同时避免过度依赖高估值个股。

(2) 债券部分

- **调整比例**：从30%提高至40%。
- **原因**：债券在经济衰退时通常表现更为稳定，能够有效分散风险。
- **选择方向**：
  - 优先配置高质量的政府债券或投资级公司债券。
  - 避免对利率敏感的长期债券，更多选择短期或中期债券以降低利率风险。

(3) 现金部分

- **调整比例**：从10%提高至15%。
- **原因**：增加现金储备可以为您提供更高的流动性和灵活性，以便在市场出现机会时快速行动。
- **用途**：
  - 作为应急资金，应对短期内可能出现的资金需求。
  - 在市场下跌时，用于抄底优质资产。

CASE：投顾AI助手（使用场景-深思熟虑）

(4) 另类投资部分

- **调整比例**：从20%降至15%。
- **原因**：另类投资虽然具备一定的增长潜力，但在经济衰退期间流动性较低，风险较高。
- **选择方向**：
  - 关注与科技行业相关的低相关性资产，例如私募股权或基础设施基金。
  - 确保所选标的具备稳健的现金流和长期增值潜力。

---

3. 资产配置建议

以下是调整后的资产配置比例（基于您的150万元投资组合）：

资产类别	**当前比例**	**调整后比例**	**金额（元）**
股票	40%	30%	450,000
债券	30%	40%	600,000
现金	10%	15%	225,000
另类投资	20%	15%	225,000

……

处理用时: 112.00秒

基于深思熟虑模式给用户的回复
collect_data => analyze_data => generate_recommendations

LangGraph使用

1. ChatTongyi的作用

ChatTongyi是LangChain社区提供的通义千问（Qwen）模型的封装类，用于与大语言模型进行交互。

```
# 导入ChatTongyi
from langchain_community.chat_models import ChatTongyi


# 创建LLM实例
llm = ChatTongyi(
    model_name="qwen-turbo-latest",
    dashscope_api_key=DASHSCOPE_API_KEY
)


# 绑定工具到LLM（关键步骤）
llm_with_tools = llm.bind_tools(tools)
```

ChatTongyi支持工具调用功能，这是实现Agent自主决策调用工具的基础。通过bind_tools()方法，LLM可以理解可用工具的描述，并在需要时生成工具调用请求。

LangGraph使用

2. LangGraph中的工具调用

使用@tool装饰器定义工具函数：

```
# 定义查询上证指数的工具
@tool
def query_shanghai_index() -> str:
    """查询上证指数实时行情，获取当前点位、涨跌和涨跌幅信息"""
    name = "上证指数"
    price = "3125.62"
    change = "6.32"
    pct = "0.20"
    result = f"{name} 当前点位: {price}，涨跌: {change}，涨跌幅: {pct}%"
    return result
# 工具列表
tools = [query_shanghai_index, query_portfolio_allocation, query_market_news]
```

工具调用流程：

用户查询 => LLM分析 => 决定调用工具 => ToolNode执行 => 返回结果 => LLM生成响应

LangGraph使用

3. 代码实现细节

```
# 绑定工具到LLM
llm_with_tools = llm.bind_tools(tools)


# 反应式Agent中使用工具
def reactive_agent(state: WealthAdvisorState) -> Dict[str, Any]:
    # 准备消息
    messages = state.get("messages", [])
    if not messages:
        messages = [HumanMessage(content=f"{system_prompt}\n\n用户问题:
{state['user_query']}")]
    # 调用带工具的LLM（LLM可能返回工具调用请求）
    response = llm_with_tools.invoke(messages)
    return {"messages": [response]}
```

```
# 判断是否需要继续调用工具
def should_continue_tools(state: WealthAdvisorState) -> str:
    messages = state.get("messages", [])
    last_message = messages[-1]
    # 检查最后一条消息是否包含工具调用
    if hasattr(last_message, "tool_calls") and last_message.tool_calls:
        return "tools"  # 需要执行工具


    return "end"  # 不需要工具，结束
```

工具调用是一个循环过程。LLM可能先调用一个工具获取数据，然后基于结果再调用其他工具，直到获得足够信息生成最终响应。

LangGraph使用

4. ToolNode的作用

ToolNode是LangGraph提供的预构建节点，专门用于执行工具调用。它接收包含工具调用请求的消息，执行相应的工具函数，并返回工具执行结果。

```
# 导入ToolNode
from langgraph.prebuilt import ToolNode

# 创建ToolNode实例
tool_node = ToolNode(tools)

# 将ToolNode添加到工作流
workflow.add_node("tools", tool_node)
```

ToolNode的工作机制

- 接收消息：从状态中获取最后一条AIMessage，该消息包含tool_calls属性
- 解析工具调用：提取工具名称和参数
- 执行工具：根据工具名称找到对应的工具函数并执行
- 生成ToolMessage：将工具执行结果封装成ToolMessage
- 更新状态：将ToolMessage添加到消息历史中

LangGraph使用

5. StateGraph的作用

StateGraph是LangGraph的核心类，用于定义和管理Agent的状态转换图。

=> 允许你构建复杂的、有状态的工作流，其中每个节点可以读取和更新共享状态。

功能	说明	代码
状态管理	维护整个工作流的共享状态，节点可以读取和更新状态	WealthAdvisorState定义
节点定义	将函数封装为工作流节点	add_node()
条件路由	根据状态动态决定下一个节点	add_conditional_edges()
固定路由	节点之间的固定连接	add_edge()
入口点	定义工作流的起始节点	set_entry_point()

StateGraph支持有状态的工作流，这意味着每个节点都可以访问和修改共享状态，非常适合实现复杂的Agent逻辑。状态通过TypedDict定义，提供了类型安全。

Summary

LangGraph的使用步骤

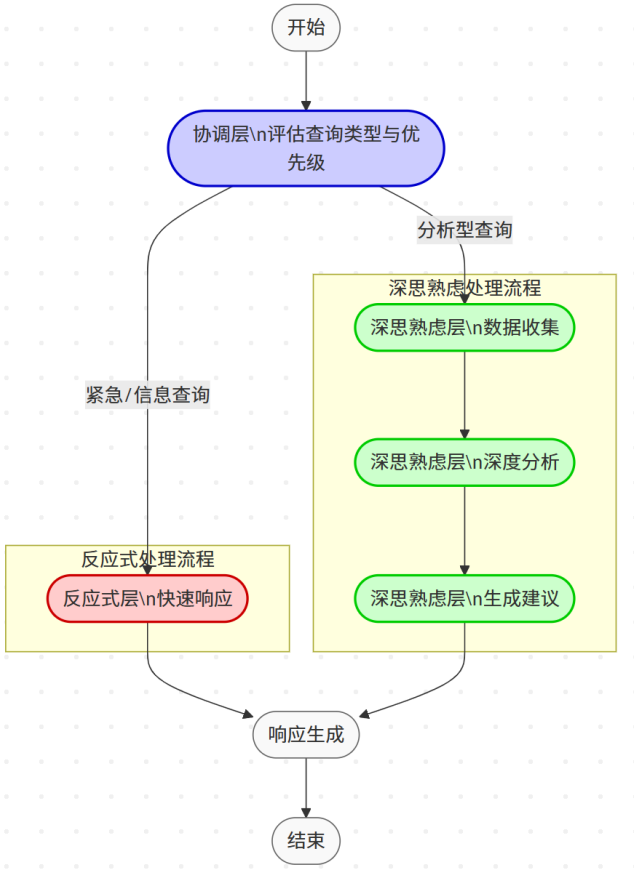
- 定义状态：使用TypedDict定义工作流的状态结构
- 创建StateGraph：实例化StateGraph并传入状态类型
- 定义节点函数：编写处理逻辑函数，接收状态并返回状态更新
- 添加节点：使用add_node()将函数添加到工作流
- 设置入口点：使用set_entry_point()指定起始节点
- 添加边：使用add_edge()或add_conditional_edges()连接节点
- 编译工作流：调用compile()方法编译工作流
- 执行工作流：调用invoke()方法传入初始状态执行

打卡：混合Hybrid Agent

以投顾AI助手为例，搭建混合Hybrid Agent

基于混合智能体架构设计，结合了反应式架构的即时响应能力和深思熟虑架构的长期规划能力

通过协调层动态选择最合适的处理模式，为客户提供智能化、个性化的财富管理咨询服务。



Summary（构建Agent的核心思想）

构建Agent的三个核心思想总结

1. 不要为所有任务构建Agent

适用场景：Agent适合处理复杂、模糊且高价值的任务，而非所有场景。

判断标准：

- 任务复杂性：若决策树可明确规划，直接构建工作流更高效。
- 任务价值：高成本（如大量Token消耗）需由高回报任务承担。
- 关键能力验证：确保Agent能处理核心子任务（如代码生成、调试）。
- 错误成本：高风险的错误需通过限制权限或人工介入来缓解。

案例：代码生成是理想场景，因其复杂性高、价值大且输出易验证（如通过单元测试）。

Summary（构建Agent的核心思想）

2. 保持简洁

Agent的核心组件：

- 环境：Agent的操作系统。
- 工具集：提供行动接口和反馈机制。
- 系统提示：定义目标、约束和预期行为。

设计原则：

- 初期避免过度复杂化，优先迭代核心组件。
- 优化（如成本、延迟）可在基础行为稳定后进行。

案例：不同功能的Agent可共享相同代码框架，仅调整工具和提示。

Summary（构建Agent的核心思想）

3. 像Agent一样思考

理解Agent的局限性：

- Agent仅基于有限上下文（10-20k Token）做决策。
- 需模拟Agent的视角（如仅通过静态截图操作电脑）以发现设计缺陷。

改进方法：

- 直接询问模型（如Claude）以验证指令清晰度或工具使用合理性。
- 分析轨迹日志，优化上下文提供方式（如分辨率信息、操作建议）。

优先评估：任务是否值得使用Agent。
简单起步：聚焦环境、工具、提示三大组件。
换位思考：通过模拟和日志分析优化Agent设计。
核心目标：在提升Agent能力的同时，平衡成本、风险与用户体验。


Thank You
Using data to solve problems