

AI Agent 投资研究实操手册

Local Agent Lab 与 TradingAgents

面向个人投资者的零基础中文自学版

路线一：Local Agent Lab

从课程包中的冻结数据、因子结果和回测结果出发，学习数据质量检查、因子解释、盘前研究、盘后归因、回测审计和多角色投资委员会。

路线二：TradingAgents

独立安装开源多 Agent 研究框架，让基本面、技术面、新闻、情绪、Bull/Bear、风险经理和组合经理协同完成一次研究流程。

不包含： Dify、Activepieces、OpenBB、FinRobot、RD-Agent 或其他 Demo。

主讲人：黄耀东博士

AQUMON 首席科学家 | 数学家 | 量化投资研究与实践

个人网站：don-huang.com

版本：Audience Self-Study Manual v1, 2026 年 7 月

本手册仅用于投资科普、软件教学与研究流程展示

目录

第一部分 先理解：两个工具分别解决什么问题	4
1 本手册适合谁	4
2 两个工具的分工	4
3 开始前的四条原则	4
第二部分 共同准备：电脑、终端与安全设置	5
4 需要准备的软件与账号	5
5 终端是什么	5
6 API Key 的安全规则	5
第三部分 Local Agent Lab：从零启动	6
7 认识课程包目录	6
8 macOS：建立 Local Agent Lab 环境	6
8.1 自动安装路线	6
8.2 手工安装路线	7
9 Windows：建立 Local Agent Lab 环境	7
10 第一遍先跑离线示例	7
11 配置真实数据与 Live LLM	8
11.1 创建.env	8
11.2 一键建立真实 Cache	8
11.3 检查真实数据是否就绪	8
12 Local Agent Lab 页面怎么用	9
12.1 侧边栏设置	9
12.2 三种模式	9
第四部分 Local Agent Lab：六项核心功能	9
13 功能一：Data Quality 数据质量审计	9

14 功能二：Factor Agent 因子筛选与风险解释	10
15 功能三：Historical Premarket 历史盘前研究	10
16 功能四：Postmarket 盘后归因与行为复盘	11
17 功能五：Backtest Auditor 回测审计	11
18 功能六：Investment Committee 多角色策略委员会	12
19 在哪里找生成结果	13
20 怎样安全修改 Prompt	13
 第五部分 TradingAgents：独立安装与首次运行	 13
21 TradingAgents 的角色结构	13
22 为什么必须建立独立环境	14
23 macOS/Linux 安装	14
24 Windows PowerShell 安装	14
25 配置模型 Provider	15
26 第一次运行：先用美股做 Smoke Test	15
27 第二次运行：中国 A 股	16
 第六部分 阅读 TradingAgents 的多 Agent 过程	 16
28 不要只看最终决定	16
29 五个必问问题	17
30 控制成本与运行时间	17
31 中断后继续：Checkpoint	17
32 Decision Log 与重复研究	18
33 用 Python 运行 TradingAgents	18
 第七部分 把两个工具结合起来学习	 18
34 推荐研究顺序	18

35 对照表：出现分歧怎么办	19
36 三项课后练习	19
36.1 练习一：同一只股票，两种框架	19
36.2 练习二：同一日期，两次 TradingAgents	19
36.3 练习三：给结论加失效条件	19
第八部分 故障排查与最终检查	20
37 Local Agent Lab 常见问题	20
38 TradingAgents 常见问题	20
39 最终自检清单	21
40 常用命令速查	21
40.1 Local Agent Lab	21
40.2 TradingAgents	21
41 官方资料与版本说明	22
42 结语	22

第 I 部分 先理解：两个工具分别解决什么问题

1 本手册适合谁

本节目标

读者不需要是程序员，但应当能够完成复制命令、打开终端、编辑配置文件和阅读表格。完成本手册后，你应能独立启动两个工具、运行一个完整案例、判断输出是否可信，并知道什么时候必须停止相信 Agent。

适合以下读者：

- 想了解 AI Agent 如何辅助投资研究，但不希望一开始就写复杂代码；
- 已经看过投资科普直播，希望课后在自己的电脑上复现；
- 希望把“让模型给答案”升级为“让多个角色检查证据、提出反例并保留人工审批”；
- 愿意把软件输出视为研究材料，而不是自动赚钱机器。

2 两个工具的分工

比较维度	Local Agent Lab	TradingAgents
定位	课程包内的可审计投资研究实验室	通用开源多 Agent 金融研究框架
数据	读取课程包已处理数据和冻结 Cache	默认通过 Yahoo Finance 等数据源获取市场信息
核心优势	数字来自确定性程序，证据路径清晰	角色丰富、辩论过程完整、可扩展性强
最适合学习	数据质量、因子、回测、归因、审计	多角色协作、Bull/Bear 辩论、风险与组合决策
安装难度	较低，随课程包提供	中等，需要单独克隆和配置模型
真实交易	不连接券商，不自动下单	模拟研究框架，不应直接连接真实账户
建议顺序	先学	后学

核心概念

Local Agent Lab 更像“有底稿的审计会议”；TradingAgents 更像“由多个 AI 角色组成的研究部”。前者帮助你建立正确流程，后者帮助你观察多 Agent 如何分工。顺序反过来，很容易先被热闹的讨论吸引，却忘了核对数字。

3 开始前的四条原则

1. **数字优先看原始表格。**收益率、权重、估值、因子、回撤和换手应由程序计算，不能由大模型“心算”。
2. **先看输入，再看结论。**不知道 Agent 看了什么，就无法判断结论是否可信。
3. **把未知写成未知。**缺失数据不能由模型补成“合理值”。

4. 始终保留人工决定。Agent 可以提出证据、风险和反例，但不能替你承担投资损失。

风险与注意事项

本手册中的任何 Buy、Hold、Sell、Overweight 或 Underweight 都是软件研究输出示例，不构成证券推荐、投资建议、收益承诺或适合性判断。

第 II 部分 共同准备：电脑、终端与安全设置

4 需要准备的软件与账号

项目	是否必需	说明
课程包 ZIP	Local Lab 必需	由课程或讲师提供，内含 Local Agent Lab、脚本、示例数据和 Cache
Python 3.12	建议	本手册统一推荐；两个项目应使用各自独立环境
VS Code	建议	方便打开文件夹、终端和配置文件
Git	TradingAgents 必需	用于克隆官方仓库
Conda/Miniconda	TradingAgents 建议	用于创建独立环境；也可以使用 venv
Tushare Token	真实 Local Lab 数据才需要	只跑 fixture 和离线模式时不需要
LLM API Key	Live 模式需要	可选择 OpenAI、DeepSeek、Qwen、GLM、MiniMax 等受支持 Provider
真实券商账户	不需要	两个工具都不应直接替你实盘下单

5 终端是什么

终端是输入命令的窗口：

- macOS：打开 `Terminal`，或在 VS Code 中选择 `Terminal → New Terminal`；
- Windows：使用 `PowerShell`，或 VS Code 内置终端；
- 命令应逐行输入，每行按一次 Enter；
- 看到报错时不要连续重复点击，先复制最后 20 行错误信息。

6 API Key 的安全规则

1. Key 只写入本机的 `.env`，不要写入 Notebook、截图或微信群；
2. `.env` 不应上传 GitHub；
3. 投屏或录屏前关闭 `.env`；
4. 不要把券商账号、身份证、完整持仓或银行信息输入 Agent；
5. 发现泄露后立即在 Provider 后台撤销并重新生成 Key。

为什么这样做

API Key 类似一张可以自动扣费的门禁卡。把它贴到公开仓库，常常比把银行卡放桌上更快收到“惊喜账单”。

第 III 部分 Local Agent Lab：从零启动

7 认识课程包目录

解压后，课程包根目录通常类似：

```
AI_Agent_Demo_China_Tushare_Ashare_2025_v9_Growth_IC/  
|-- .env.example  
|-- instructor_setup.py  
|-- run_pipeline.py  
|-- requirements_core.txt  
|-- requirements_tested_lock.txt  
|-- config/  
|-- data/  
|-- local_agent_lab/app.py  
|-- classroom_cache/snapshots/  
|-- prompts/  
|-- scripts/  
`-- reports/
```

成功标志

你打开的文件夹中应直接看见 `instructor_setup.py` 和 `local_agent_lab`。如果外面还套了一层同名文件夹，请继续进入下一层再打开。

8 macOS：建立 Local Agent Lab 环境

8.1 自动安装路线

照着做

1. 在 VS Code 选择 `File → Open Folder`，打开课程包根目录。
2. 打开 VS Code Terminal。
3. 运行：

```
bash scripts/bootstrap_mac.sh
```

4. 安装结束后关闭旧 Terminal，再打开一个新 Terminal。
5. 检查：

```
python --version  
which python
```

正确路径应包含：

```
.../AI_Agent_Demo.../.venv/bin/python
```

8.2 手工安装路线

自动脚本失败时运行：

```
python3.12 -m venv .venv
source .venv/bin/activate
python -m pip install --upgrade pip
python -m pip install -r requirements_core.txt
```

若出现依赖冲突，改用课程包测试过的锁定版本：

```
python -m pip install -r requirements_tested_lock.txt
```

9 Windows：建立 Local Agent Lab 环境

在课程包根目录打开 PowerShell：

```
Set-ExecutionPolicy -Scope Process Bypass
.\scripts\bootstrap_windows.ps1
.\.venv\Scripts\Activate.ps1
python --version
where.exe python
```

成功标志

终端提示符通常会出现 (.venv)，而 where.exe python 的第一条路径应指向课程包中的 .venv。

10 第一遍先跑离线示例

本节目标

第一次运行不要急着配置 Tushare 或付费模型。先证明软件环境能够工作，再增加外部账号和网络变量。

在课程包根目录运行：

```
python instructor_setup.py --fixture --offline-agents
```

这条命令使用合成 fixture 数据并生成离线规则报告，不需要 Tushare Token，也不消耗 LLM API 费用。完成后运行：

```
bash scripts/run_local_agent_lab.sh
```

Windows 使用：

```
.\.venv\Scripts\Activate.ps1
streamlit run local_agent_lab\app.py
```

浏览器打开：

```
http://127.0.0.1:8501
```

成功标志

页面能够打开；侧边栏可选择 fixture Cache；页面明显提示 Synthetic/Fixture；任一 Tab 能生成离线报告。此时只说明软件可运行，不说明投资数据真实。

11 配置真实数据与 Live LLM

11.1 创建.env

macOS/Linux:

```
cp .env.example .env
```

Windows PowerShell:

```
Copy-Item .env.example .env
```

用 VS Code 打开 .env，填写：

```
TUSHARE_TOKEN=your_real_tushare_token

LLM_API_KEY=your_llm_key
LLM_BASE_URL=https://your-provider-compatible-endpoint
LLM_MODEL=your-tested-model-name
LLM_TEMPERATURE=0.1
AGENT_MODE=auto
```

11.2 一键建立真实 Cache

```
python instructor_setup.py --cache-id real_2025_course
```

该向导通常会依次完成：环境检查、Tushare 权限测试、数据下载、数据处理、因子与回测、Cache 冻结、Agent 报告和最终 Preflight。

风险与注意事项

Tushare 不同接口可能需要不同积分或权限。能够调用 stock_basic，不代表 fina_indicator、daily_basic 和 index_daily 一定可用。

11.3 检查真实数据是否就绪

```
python scripts/preflight.py --level full \
  --cache-id real_2025_course
```

打开：

```
reports/preflight_latest.md
```

必须确认：

- Cache ID 是 real_2025_course；
- Provenance 显示真实 Tushare 派生数据；
- 不是 fixture 或 synthetic；

- Cache 哈希验证通过；
- 关键数据和 Agent 报告存在。

12 Local Agent Lab 页面怎么用

12.1 侧边栏设置

建议第一次真实运行使用：

设置	建议值与含义
Cache ID	real_2025_course；练习软件时才选 fixture
Agent Mode	先选 Offline Deterministic，再测试 Auto Fallback，最后尝试 Live LLM
Temperature	约 0.1；较低值通常减少文字波动，但不能保证完全一致
Input Preview	每次生成前展开，确认日期、文件、股票数量和数据来源
Generate	输入确认后再点击；不要连续重复点击造成重复扣费

12.2 三种模式

Offline Deterministic 不调用大模型，按固定规则生成结构化报告；最稳定，适合核对数字。

Live LLM 调用外部模型；表达更自然，但可能超时、产生差异或幻觉。

Auto Fallback 优先 Live；失败后自动切换 Offline，适合一般使用。

如何阅读结果

先比较 Offline 与 Live 的数字是否一致，再比较文字是否更有洞察。若关键数字不同，应停止阅读结论，回到 Input Preview 和 CSV 查错。

第 IV 部分 Local Agent Lab：六项核心功能

13 功能一：Data Quality 数据质量审计

本节目标

先回答“数据能不能用”，再讨论“策略有没有用”。这是整个流程最重要的一步。

照着做

1. 打开 `Data Quality` Tab。
2. 选择 `real_2025_course`。
3. 展开 Input Preview，确认数据来源、日期范围、ticker 数量和关键接口。
4. 点击 `Generate`。
5. 在输出中寻找 PASS、WARN、BLOCKER 或同等分类。
6. 重点检查缺失字段、日期覆盖、财务公告日期、基准对齐和 fixture 警告。

如何阅读结果

一份好的数据质量报告至少应回答：数据从哪里来、覆盖哪段时间、缺哪些股票或字段、是否存在前视偏差、哪些问题会阻止后续回测。若报告只说“数据质量良好”，却不给证据，价值接近一句礼貌用语。

练习：观察 `fina_indicator` 少一只股票时，Agent 是否明确标记缺失，而不是自行估算。

14 功能二：Factor Agent 因子筛选与风险解释

本节目标

理解成长、质量、动量、低波和价值如何共同形成综合评分，以及为什么高分股票仍可能被流动性或行业约束过滤。

照着做

1. 打开 `Factor Agent` 或因子解释 Tab。
2. 在 Input Preview 查看最新 signal date、eligible 数量和 selected 数量。
3. 核对五因子权重是否与课程配置一致。
4. 点击 Generate。
5. 找出至少一只具有明显因子冲突的股票，例如成长高但估值昂贵。
6. 检查 Agent 是否区分“分数不够”“数据缺失”“流动性不足”和“行业上限”。

如何阅读结果

Agent 的任务是解释已经计算好的排名，不是悄悄重算排名。最终选股名单应以 `data/processed/selected_portfolios_2025.csv` 等确定性文件为准。

15 功能三：Historical Premarket 历史盘前研究

本节目标

在某个历史交易日前，只使用当时可获得的信息，形成风险清单和待确认问题，而不是事后解释行情。

照着做

1. 打开 **Premarket** Tab。
2. 选择一个课程包支持的历史日期。
3. 检查 Input Preview 中是否只包含该日期之前可得信息。
4. 点击 Generate。
5. 阅读组合暴露、异常波动、流动性、资料缺口和计划内动作。
6. 把“预测涨跌”改写为“列出触发复核的条件”。

如何阅读结果

好的盘前报告会告诉你“今天应该核对什么”；差的盘前报告会用宏大叙事假装知道收盘价。盘前研究的价值是纪律，不是水晶球。

16 功能四：Postmarket 盘后归因与行为复盘

本节目标

把组合收益拆成可解释来源，并区分策略本身、交易成本、基准环境和投资者行为。

照着做

1. 打开 **Postmarket** Tab。
2. 在 Input Preview 查看组合收益、基准收益、成本、持仓和调仓记录。
3. 点击 Generate。
4. 检查报告是否区分 gross return 与 net return。
5. 找出贡献最大和拖累最大的持仓。
6. 检查是否存在计划外交易、追涨杀跌或偏离规则。

如何阅读结果

“今天亏了，所以策略错了”不是归因；“今天赚了，所以决策英明”也不是。归因要回答：收益来自市场、因子、个股、成本还是行为偏差。

17 功能五：Backtest Auditor 回测审计

本节目标

不只看收益曲线，而是检查前视偏差、Point-in-Time、交易成本、换手、基准、样本长度和过度拟合。

照着做

1. 打开 `Backtest Auditor` Tab。
2. 展开策略规格、回测指标和交易记录。
3. 点击 `Generate`。
4. 检查信号日、执行日和收益计算是否错位。
5. 检查财务数据是否按公告日进入模型。
6. 检查 `gross/net`、成本、换手和基准是否完整。
7. 找出固定 `Universe`、单年度样本和幸存者偏差等限制。
8. 阅读“下一步必须补做的测试”，不要只看最终评级。

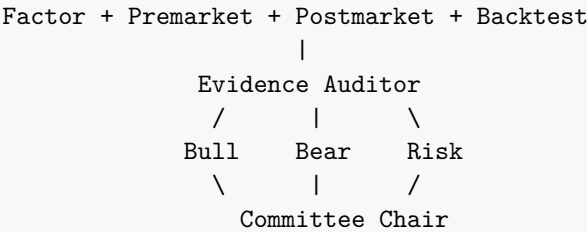
风险与注意事项

一次跑赢基准的回测，最多证明“这套规则在这段历史样本里有效”。它不能单独证明未来有效，更不能证明适合你的资金、风险承受能力和投资期限。

18 功能六：Investment Committee 多角色策略委员会

本节目标

让 `Evidence Auditor`、`Bull`、`Bear`、`Risk` 和 `Chair` 围绕同一套底稿进行结构化反证，而不是让几个 `Agent` 轮流发表相似观点。



照着做

1. 先阅读前五项功能的关键结论。
2. 打开 `Investment Committee` Tab。
3. 在运行前，自己先写下判断：有效、条件性有效、证据不足或不可行。
4. 点击 `Generate`。
5. 检查 `Evidence Auditor` 是否发现前序报告与 `CSV` 不一致。
6. 检查 `Bull` 是否提出反证条件，而不是只讲优点。
7. 检查 `Bear` 与 `Risk` 是否区分可修复风险和致命问题。
8. 阅读 `Chair` 的条件、失效情景、补充测试和人工复核要求。
9. 比较委员会结论与你的预判，记录差异来自哪项证据。

核心概念

多 Agent 的价值不是“投票人数更多”，而是把证据核对、乐观论证、悲观反证、风险约束和最终裁决拆开。五个角色一起产生幻觉，仍然只是规模更大的幻觉。

19 在哪里找生成结果

常见路径：

```
classroom_cache/snapshots/real_2025_course/agent_reports/
reports/
data/processed/
```

建议每次保存三类材料：

1. 原始 CSV 或 Parquet；
2. Agent 输入预览或证据摘要；
3. Agent 输出 Markdown。

文件名中加入日期和 Cache ID，例如：

```
2026-07-21_real_2025_course_backtest_auditor.md
```

20 怎样安全修改 Prompt

1. 先复制原 Prompt，再修改副本；
2. 保留“不重算、不补数据、不输出买卖指令”等边界；
3. 明确要求引用字段、日期和文件来源；
4. 要求把未知标成 Unknown；
5. 每次只改一个变量，比较前后输出；
6. 发现数字改变时，回到原始 CSV 核对。

一个稳健的追加指令示例：

```
For every conclusion, cite the supporting input field and date.
Label unsupported claims as UNKNOWN.
Do not recalculate portfolio weights or performance figures.
List at least three conditions that would invalidate the conclusion.
```

第 V 部分 TradingAgents：独立安装与首次运行

21 TradingAgents 的角色结构

官方框架将研究拆成多个角色：

- Fundamentals Analyst：基本面；
- Technical/Market Analyst：价格和技术指标；

- News Analyst: 新闻与宏观信息;
- Sentiment Analyst: 市场情绪;
- Bull 与 Bear Researcher: 正反方辩论;
- Trader Agent: 形成交易方案;
- Risk Management Team: 从不同风险偏好审查;
- Portfolio Manager: 给出最终研究决定。

风险与注意事项

TradingAgents 是研究框架。它的最终决定受到模型、数据源、温度、研究日期和随机性的影响，不是稳定可复制的交易信号。

22 为什么必须建立独立环境

不要把 TradingAgents 安装进 Local Agent Lab 的 `.venv`。两个项目依赖更新节奏不同，共用环境容易出现 `pandas`、`LangChain`、`LangGraph` 或其他包版本冲突。

建议目录：

```
~/Projects/  
|-- AI_Agent_Demo_China_.../      # Local Agent Lab  
`-- TradingAgents/               # independent project
```

23 macOS/Linux 安装

照着做

1. 打开新的终端。
2. 进入准备存放项目的目录。
3. 运行：

```
git clone https://github.com/TauricResearch/TradingAgents.git  
cd TradingAgents  
conda create -n tradingagents python=3.12 -y  
conda activate tradingagents  
python -m pip install --upgrade pip  
pip install .  
cp .env.example .env
```

24 Windows PowerShell 安装

```
git clone https://github.com/TauricResearch/TradingAgents.git  
cd TradingAgents  
conda create -n tradingagents python=3.12 -y  
conda activate tradingagents  
python -m pip install --upgrade pip  
pip install .  
Copy-Item .env.example .env
```

成功标志

运行 `tradingagents --help` 或 `python -m cli.main` 不再提示 `module not found`；终端前方显示 `(tradingagents)`。

25 配置模型 Provider

只选择一个 Provider 开始。将对应 Key 写入 `.env`：

```
# Choose only the provider you use
OPENAI_API_KEY=...
DEEPSEEK_API_KEY=...
DASHSCOPE_CN_API_KEY=...
ZHIPU_CN_API_KEY=...
MINIMAX_CN_API_KEY=...
OPENROUTER_API_KEY=...
```

常用选择：

场景	Provider	建议
中国大陆网络环境	DeepSeek、Qwen China、GLM China、MiniMax China	先用官方支持的中国区端点，减少跨境网络不稳定
海外或国际网络	OpenAI、Anthropic、Google、OpenRouter 等	关注模型费用、速率限制和地区可用性
本地模型	Ollama	隐私较好，但需要足够内存和本地模型能力
自建兼容服务	openai_compatible	配置 backend URL；适合高级用户

风险与注意事项

模型名称和价格变化很快。优先使用 CLI 当前列出的模型，避免从旧教程复制已经下线的模型 ID。

26 第一次运行：先用美股做 Smoke Test

先选择数据覆盖通常较稳定的美股，例如 AAPL 或 NVDA：

```
conda activate tradingagents
cd TradingAgents
tradingagents
```

若命令不可用：

```
python -m cli.main
```

在交互界面中：

- 1. Ticker 输入 AAPL；
- 2. Analysis Date 选择一个过去交易日；
- 3. 选择你已配置 Key 的 Provider；
- 4. Quick/Deep Model 先选成本较低且已测试的模型；

5. Research Depth 先选较浅；
6. Debate Rounds 先设为 1；
7. 启动分析并观察各角色逐步输出。

成功标志

至少有市场/基本面等分析产物、Bull/Bear 讨论、风险审查和最终 Portfolio Manager 结论；整个流程没有 API Key、ticker 或数据下载错误。

27 第二次运行：中国 A 股

TradingAgents 使用 Yahoo Finance 风格 ticker：

市场	示例	说明
上海 A 股	600519.SS	贵州茅台；注意 Yahoo 使用.SS
深圳 A 股	000858.SZ	五粮液
香港股票	0700.HK	腾讯控股
美国股票	AAPL	不加交易所后缀

Tushare 与 Yahoo 的上海后缀不同：

Tushare: 600519.SH
Yahoo Finance: 600519.SS

照着做

1. 先测试 600519.SS 能否取得价格数据。
2. 选择过去交易日，不要选择未来日期或周末。
3. 使用较浅研究深度完成第一次 A 股运行。
4. 检查公司名称是否与 ticker 一致。
5. 检查基本面、新闻和情绪数据是否有缺失或地域覆盖不足。
6. 将最终结论与 Local Agent Lab 的因子/回测底稿对照。

风险与注意事项

Yahoo 对 A 股的价格覆盖通常比新闻、情绪和基本面覆盖更稳定。某个角色没有足够数据时，应把输出视为低置信度，而不是让其他角色用语言填满空白。

第 VI 部分 阅读 TradingAgents 的多 Agent 过程

28 不要只看最终决定

建议按以下顺序阅读：

1. **身份与日期**：ticker、公司名称、分析日期是否正确；
2. **数据证据**：价格、指标、财务和新闻来自什么时间；

3. **分析师报告**：各角色是否真的使用不同证据；
4. **Bull/Bear 辩论**：是否回应对方，而非各说各话；
5. **Trader**：方案是否建立在前述证据上；
6. **Risk**：是否讨论波动、流动性、数据缺口和仓位风险；
7. **Portfolio Manager**：最终评级是否附带条件和失效情景。

29 五个必问问题

每次运行后写下：

1. 哪三项是可核实事实？
2. 哪三项只是模型推断？
3. 哪个角色缺少数据却仍然说得很肯定？
4. Bull 与 Bear 是否使用了同一套时间截面？
5. 什么新证据出现时，最终结论必须改变？

30 控制成本与运行时间

第一次练习建议：

- 一次只分析一只股票；
- Research Depth 选浅；
- Debate Rounds 设为 1；
- Quick Model 使用便宜模型；
- Deep Model 只给关键角色使用；
- 同一输入不要因页面没有立即响应而重复启动。

为什么这样做

多 Agent 成本大致随“角色数量 × 辩论轮数 × 上下文长度 × 模型单价”上升。让十个 Agent 开三轮会，常常比人类会议更准时，也更会花钱。

31 中断后继续：Checkpoint

长流程建议启用 checkpoint：

```
tradingagents analyze --checkpoint
```

若要清除旧 checkpoint：

```
tradingagents analyze --clear-checkpoints
```

默认 checkpoint 常见位置：

```
~/.tradingagents/cache/checkpoints/
```

32 Decision Log 与重复研究

框架会保存历史决定，用于后续反思。默认记录路径常见为：

```
~/tradingagents/memory/trading_memory.md
```

建议做法：

1. 每次固定 ticker 和 analysis date；
2. 记录 Provider、模型、研究深度和辩论轮数；
3. 将输出另存为带时间戳的研究记录；
4. 比较两次运行的共同证据与分歧；
5. 不把文字差异误认为市场发生变化。

33 用 Python 运行 TradingAgents

建立 my_analysis.py：

```
from tradingagents.graph.trading_graph import TradingAgentsGraph
from tradingagents.default_config import DEFAULT_CONFIG

config = DEFAULT_CONFIG.copy()
config["llm_provider"] = "deepseek"
config["deep_think_llm"] = "YOUR_CURRENT_DEEP_MODEL_ID"
config["quick_think_llm"] = "YOUR_CURRENT_QUICK_MODEL_ID"
config["max_debate_rounds"] = 1
config["temperature"] = 0.1
config["checkpoint_enabled"] = True

ta = TradingAgentsGraph(debug=True, config=config)
_, decision = ta.propagate("600519.SS", "2026-06-30")
print(decision)
```

运行：

```
conda activate tradingagents
python my_analysis.py
```

风险与注意事项

模型 ID 应以当前 Provider 和 CLI 实际支持列表为准。不要把示例占位符原样运行，也不要再在 Python 文件中硬编码 API Key。

第 VII 部分 把两个工具结合起来学习

34 推荐研究顺序

以贵州茅台为例：

1. 在 Local Agent Lab 运行 Data Quality，确认课程数据底稿；
2. 查看 Factor Agent，理解其在五因子中的位置；

3. 查看 Backtest Auditor，理解策略层面的证据限制；
4. 运行 Local Investment Committee，得到策略审计结论；
5. 在 TradingAgents 输入 600519.SS 和同一历史日期；
6. 比较 TradingAgents 的基本面、技术面、新闻和风险观点；
7. 把冲突分成“数据不同”“日期不同”“方法不同”“模型推断不同”；
8. 最终只保留能够回到原始证据核实的结论。

35 对照表：出现分歧怎么办

分歧类型	可能原因	处理方法
价格或收益不同	日期、复权方式、时区、数据来源不同	对齐交易日和复权口径，回到原始行情
基本面不同	公告日、财报版本或供应商覆盖不同	核对 ann date、period 和原始公告
因子观点不同	Local 是量化横截面，TradingAgents 是公司研究	不要求两者结论完全相同，先确认研究问题
新闻观点不同	实时新闻会变化，历史日期不等于历史新闻快照	记录抓取时间，降低历史新闻结论置信度
最终评级不同	模型、Prompt、随机性、角色权重不同	比较证据链，不比较谁的语气更肯定

36 三项课后练习

36.1 练习一：同一只股票，两种框架

选择一只课程 Universe 股票，保存：

- Local Factor Agent 报告；
- Local Backtest Auditor 报告；
- TradingAgents 完整输出；
- 一页自己的证据对照表。

36.2 练习二：同一日期，两次 TradingAgents

保持 ticker、日期、模型和参数不变，运行两次，标记：

- 完全一致的事实；
- 表述不同但含义一致的部分；
- 真正改变结论的差异；
- 可能来自实时新闻变化的内容。

36.3 练习三：给结论加失效条件

对任何“看多”或“看空”结论，至少补充：

1. 一项数据失效条件；
2. 一项估值或基本面条件；
3. 一项市场状态条件；
4. 一项流动性或交易成本条件。

第 VIII 部分 故障排查与最终检查

37 Local Agent Lab 常见问题

现象	常见原因	处理
终端仍显示 base	旧 Terminal 未重开	激活 <code>.venv</code> , 关闭旧 Terminal 后重开
找不到模块	Python 不是项目 <code>.venv</code>	检查 <code>which python</code> 或 <code>where.exe python</code>
8501 页面打不开	Streamlit 未启动或端口占用	终端直接运行 app; 检查错误和端口
只有 fixture	尚未生成真实 Cache	运行真实 <code>setup</code> , 并检查 <code>Pre-flight</code>
Live LLM 超时	网络、限流、Key 或模型名错误	测试 LLM; 切 Auto/Offline
Agent 数字与 CSV 不同	Prompt 重算或模型幻觉	停止使用结论, 回到 CSV 和 Offline 报告
Tushare 部分接口失败 中文乱码	账号权限或积分不同 终端或文件编码问题	逐接口检查, 不用模型补齐数据 使用 UTF-8, 重开终端

38 TradingAgents 常见问题

现象	常见原因	处理
tradingagents 不存在	环境未激活或安装失败	激活 Conda; 运行 <code>pip install .</code>
API Key missing	<code>.env</code> 未填写或变量名错误	使用官方变量名, 重开终端
429/Rate limit A 股 ticker 无数据 历史日期失败	请求过多或余额不足 使用了 <code>.SH</code> 或错误代码 周末、未来日期或数据源缺失	降低深度和轮数, 等待额度恢复 上海改为 <code>.SS</code> , 深圳用 <code>.SZ</code> 改为已收盘的交易日
角色报告为空	对应数据源覆盖不足	明确标记缺失, 换 ticker 或数据源
结果每次不同	LLM 和实时数据非确定性	固定输入并保存版本; 比较证据而非逐字一致
运行中断 安装依赖冲突	网络或长流程失败 与其他项目共用环境	使用 <code>--checkpoint</code> 续跑 删除环境并单独重建 tradingagents

39 最终自检清单

我已经做到	
✓	Local Agent Lab 使用独立 .venv, TradingAgents 使用独立 Conda 环境。
✓	我能区分 fixture、真实 Cache 和实时外部数据。
✓	我没有把 API Key、券商账号或个人敏感信息输入公开界面。
✓	我会在生成前查看 Input Preview。
✓	我知道数字冲突时应以确定性底稿为准。
✓	我完成了 Local Data Quality、Factor、Backtest Auditor 和 Investment Committee。
✓	我用 AAPL/NVDA 完成了 TradingAgents smoke test。
✓	我知道上海 A 股在 Yahoo 使用 .SS, 不是 Tushare 的 .SH。
✓	我会控制研究深度、辩论轮数和模型成本。
✓	我能区分事实、模型推断和未知。
✓	我不会把 Agent 最终评级直接当成真实交易指令。

40 常用命令速查

40.1 Local Agent Lab

```
# Offline fixture setup
python instructor_setup.py --fixture --offline-agents

# Real data and cache setup
python instructor_setup.py --cache-id real_2025_course

# Readiness check
python scripts/preflight.py --level full \
    --cache-id real_2025_course

# Launch app on macOS/Linux
bash scripts/run_local_agent_lab.sh

# Launch app on Windows
streamlit run local_agent_lab\app.py
```

40.2 TradingAgents

```
# Install
git clone https://github.com/TauricResearch/TradingAgents.git
cd TradingAgents
conda create -n tradingagents python=3.12 -y
conda activate tradingagents
pip install .

# Launch
tradingagents
# or
python -m cli.main

# Resume long workflow
```

```
tradingagents analyze --checkpoint
```

41 官方资料与版本说明

- TradingAgents 官方仓库: <https://github.com/TauricResearch/TradingAgents>
- TradingAgents 论文: <https://arxiv.org/abs/2412.20138>
- Tushare Pro 文档: <https://tushare.pro/document/2>

截至 2026 年 7 月, 本手册参考 TradingAgents v0.2.5 系列官方文档。开源项目更新较快; 若命令或菜单变化, 以官方 README、CHANGELOG.md 和仓库当前版本为准。升级前应保存可运行环境, 不要在重要演示当天执行未经测试的 `git pull` 或依赖升级。

42 结语

核心概念

正确使用 AI Agent 的关键, 不是让模型替你做决定, 而是把研究拆成可核验的步骤: 数据来源明确、数字可复算、观点有反证、风险有边界、最终由人负责。工具可以让研究更快; 审计决定它是否值得相信。

风险与注意事项

所有案例仅用于投资科普、软件操作和研究方法演示, 不构成任何投资建议、证券推荐、招揽、收益保证或适合性意见。真实投资还需考虑个人目标、风险承受能力、投资期限、费用、税务、流动性、监管要求及独立专业意见。