

# TradingAgents 多 Agent 投资研究框架

## 手把手配置与使用手册

面向个人投资者的零基础中文自学版

本手册只讲解 **TradingAgents**：从安装 Python、建立独立环境、配置模型与 API Key，到运行第一只股票、阅读多 Agent 辩论、分析 A 股、控制成本、保存研究记录和排查错误。

全书只保留 **TradingAgents** 的配置、运行、研究解读与故障排查。

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](https://don-huang.com)

版本：TradingAgents 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 开始之前：先理解这个框架         | 4  |
| 1 本手册适合谁                  | 4  |
| 2 TradingAgents 是什么       | 4  |
| 3 它能做什么，不能做什么             | 4  |
| 4 多 Agent 的角色结构           | 4  |
| 第二部分 电脑与账号准备              | 5  |
| 5 你需要准备什么                 | 5  |
| 6 先确认 Git、Conda 和 Python  | 6  |
| 7 终端的基本规则                 | 6  |
| 8 API Key 的安全规则           | 6  |
| 第三部分 安装 TradingAgents     | 7  |
| 9 macOS/Linux: Conda 安装路线 | 7  |
| 10 Windows: Conda 安装路线    | 7  |
| 11 不使用 Conda: venv 备选路线   | 8  |
| 12 Docker 路线：可选           | 8  |
| 13 安装后先冻结一个可运行版本          | 8  |
| 第四部分 配置模型与数据源             | 9  |
| 14 只选择一个模型 Provider 开始    | 9  |
| 15 编辑.env                 | 9  |
| 16 模型怎么选                  | 9  |
| 17 数据源的基本认识               | 10 |

|                              |           |
|------------------------------|-----------|
| <b>第五部分 第一次完整运行</b>          | <b>10</b> |
| 18 先用美股做 Smoke Test          | 10        |
| 19 为什么分析日期必须谨慎               | 11        |
| 20 运行时你会看到什么                 | 11        |
| 21 不要因为界面停顿而重复启动             | 12        |
| <b>第六部分 运行 A 股、港股与其他市场</b>   | <b>12</b> |
| 22 Ticker 的交易所后缀             | 12        |
| 23 第一次 A 股运行                 | 13        |
| 24 A 股结果为什么更需要人工检查           | 13        |
| <b>第七部分 如何阅读多 Agent 研究结果</b> | <b>13</b> |
| 25 正确阅读顺序                    | 13        |
| 26 五个必问问题                    | 14        |
| 27 识别“伪辩论”                   | 14        |
| 28 最终评级应该怎样理解                | 14        |
| <b>第八部分 控制成本、保存记录与恢复运行</b>   | <b>14</b> |
| 29 API 成本如何产生                | 14        |
| 30 启用 Checkpoint             | 15        |
| 31 Decision Log              | 15        |
| 32 建立自己的研究记录                 | 15        |
| 33 同一输入为什么会出现不同结果            | 16        |
| <b>第九部分 使用 Python 自定义运行</b>  | <b>16</b> |
| 34 最小 Python 示例              | 16        |
| 35 配置默认 yfinance 数据源         | 17        |

|                             |               |
|-----------------------------|---------------|
| <b>36 修改参数时一次只改一个变量</b>     | <b>17</b>     |
| <b>37 一个可复用的研究任务模板</b>      | <b>17</b>     |
| <br><b>第十部分 常见问题与故障排查</b>   | <br><b>18</b> |
| <b>38 安装与环境问题</b>           | <b>18</b>     |
| <b>39 API 与模型问题</b>         | <b>18</b>     |
| <b>40 数据与 ticker 问题</b>     | <b>18</b>     |
| <b>41 运行中断与结果异常</b>         | <b>19</b>     |
| <br><b>第十一部分 最终检查表与速查命令</b> | <br><b>19</b> |
| <b>42 最终自检清单</b>            | <b>19</b>     |
| <b>43 常用命令速查</b>            | <b>19</b>     |
| 43.1 安装 . . . . .           | 19            |
| 43.2 启动 . . . . .           | 20            |
| 43.3 Checkpoint . . . . .   | 20            |
| 43.4 检查环境 . . . . .         | 20            |
| 43.5 更新前先备份 . . . . .       | 20            |
| <b>44 官方资料与版本说明</b>         | <b>20</b>     |
| <b>45 结语</b>                | <b>20</b>     |

# 第 I 部分 开始之前：先理解这个框架

## 1 本手册适合谁

### 本节目标

读者不需要具备专业编程背景。完成本手册后，你应能独立安装 TradingAgents、运行一只股票的完整研究流程、理解每个 Agent 的职责、判断输出是否可信，并知道哪些内容绝对不能直接当成交易指令。

适合以下读者：

- 看过投资科普直播，希望在自己的电脑上复现多 Agent 研究流程；
- 想了解 AI 如何分别扮演基本面、技术面、新闻、情绪、风险与组合管理角色；
- 能够复制命令、打开终端、编辑配置文件，但不希望一开始编写复杂程序；
- 愿意把 Agent 输出视为研究材料，而不是自动赚钱机器。

## 2 TradingAgents 是什么

TradingAgents 是一个开源的多 Agent 金融研究框架。它把一项股票研究任务拆成多个专业角色，让不同 Agent 分别收集证据、提出观点、互相辩论，再由风险团队和组合经理给出最终研究结论。

### 核心概念

单一聊天机器人通常从一个角度直接给答案；TradingAgents 更像一个虚拟研究部门。它的价值不在于“多说几遍”，而在于把研究拆成不同职责，并把观点冲突显性化。

## 3 它能做什么，不能做什么

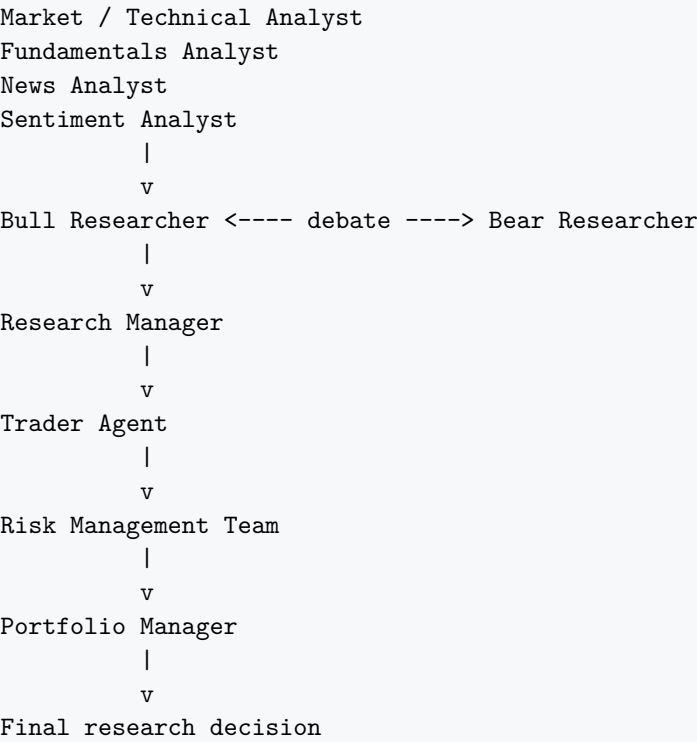
| 可以辅助                 | 不能替代           |
|----------------------|----------------|
| 整理市场、基本面、技术面、新闻和情绪信息 | 真实券商账户中的交易授权   |
| 组织 Bull/Bear 正反方研究辩论 | 对未来收益的保证       |
| 生成研究摘要、风险条件和最终评级     | 个人风险承受能力与适合性评估 |
| 保存决策记录，供未来回顾         | 经审计财务数据或监管披露原文 |
| 帮助投资者发现遗漏的反方证据       | 投资者本人对损失承担的责任  |

### 风险与注意事项

TradingAgents 是研究框架，不是经过实盘验证的自动交易系统。其输出会受到模型、数据源、分析日期、提示词、参数、网络状态和随机性的影响，不构成证券推荐、投资建议或收益承诺。

## 4 多 Agent 的角色结构

典型工作流如下：



主要角色：

- **Market/Technical Analyst**：分析价格、成交量和技术指标；
- **Fundamentals Analyst**：分析财务、估值和公司经营；
- **News Analyst**：整理公司新闻和宏观事件；
- **Sentiment Analyst**：分析市场情绪和相关文本；
- **Bull/Bear Researchers**：分别构建看多与看空论点并展开辩论；
- **Research Manager**：综合研究员观点；
- **Trader Agent**：形成可执行但仍属模拟的研究方案；
- **Risk Management Team**：从不同风险偏好审查方案；
- **Portfolio Manager**：给出最终评级、理由和风险条件。

## 第 II 部分 电脑与账号准备

### 5 你需要准备什么

| 项目              | 是否必需 | 说明                                |
|-----------------|------|-----------------------------------|
| 电脑              | 必需   | macOS、Windows 或 Linux；建议至少 8GB 内存 |
| Git             | 必需   | 用于下载官方代码仓库                        |
| Python          | 必需   | 官方项目要求 Python 3.10 或以上；本手册推荐 3.13 |
| Conda/Miniconda | 建议   | 建立独立环境，避免与其他 Python 项目冲突          |
| VS Code         | 建议   | 方便打开项目、终端和配置文件                    |

|             |         |                    |
|-------------|---------|--------------------|
| LLM API Key | 云端模型必需  | 只需选择一个 Provider 开始 |
| Ollama      | 本地模型才需要 | 不使用云端 API 时的可选方案   |
| 券商账户        | 不需要     | 本手册不连接券商，不执行真实交易   |

## 6 先确认 Git、Conda 和 Python

打开终端，逐行运行：

```
git --version
conda --version
python --version
```

Windows 也可以尝试：

```
py --version
```

### 成功标志

三条命令至少应显示版本号，而不是“command not found”或“不是内部或外部命令”。如果 Conda 不存在，先安装 Miniconda，再重新打开终端。

## 7 终端的基本规则

- macOS：使用 Terminal，或 VS Code 的 `Terminal → New Terminal`；
- Windows：使用 Anaconda Prompt、PowerShell，或 VS Code 内置终端；
- 命令应逐行执行，每行按一次 Enter；
- 看到错误时，先阅读最后 20 行，不要连续重复运行；
- 路径中尽量避免中文、空格和特殊符号。

建议项目目录：

```
macOS/Linux: ~/Projects/TradingAgents
Windows:      C:\Projects\TradingAgents
```

## 8 API Key 的安全规则

1. Key 只写入本机的 `.env`；
2. 不要把 Key 写进截图、微信群、Notebook 或公开 GitHub；
3. 不要把券商密码、身份证、银行卡或完整持仓上传给 Agent；
4. 投屏或录屏前关闭 `.env`；
5. 怀疑泄露时，立即在 Provider 后台撤销旧 Key。

### 为什么这样做

API Key 类似一张可以自动扣费的门禁卡。公开上传后，最先发现它的未必是你的学生，也可能是互联网机器人。

## 第 III 部分 安装 TradingAgents

### 9 macOS/Linux: Conda 安装路线

#### 照着做

1. 打开 Terminal。
2. 进入准备存放项目的目录。
3. 下载官方代码并建立独立环境：

```
mkdir -p ~/Projects
cd ~/Projects
git clone https://github.com/TauricResearch/TradingAgents.git
cd TradingAgents
conda create -n tradingagents python=3.13 -y
conda activate tradingagents
python -m pip install --upgrade pip
pip install .
cp .env.example .env
```

检查安装：

```
tradingagents --help
```

如果该命令不可用：

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

### 10 Windows: Conda 安装路线

建议打开 `Anaconda Prompt` 或已启用 Conda 的 PowerShell：

```
mkdir C:\Projects
cd C:\Projects
git clone https://github.com/TauricResearch/TradingAgents.git
cd TradingAgents
conda create -n tradingagents python=3.13 -y
conda activate tradingagents
python -m pip install --upgrade pip
pip install .
Copy-Item .env.example .env
```

在传统 Command Prompt 中，复制文件也可以使用：

```
copy .env.example .env
```

#### 成功标志

终端行首出现 (tradingagents)，并且 `tradingagents --help` 或 `python -m cli.main --help` 能显示帮助信息。

## 11 不使用 Conda: venv 备选路线

macOS/Linux:

```
python3 -m venv .venv
source .venv/bin/activate
python -m pip install --upgrade pip
pip install .
```

Windows PowerShell:

```
py -m venv .venv
.\.venv\Scripts\Activate.ps1
python -m pip install --upgrade pip
pip install .
```

### 风险与注意事项

若 PowerShell 阻止激活脚本，可改用 Anaconda Prompt，或在了解公司安全政策后处理 Execution Policy。不要从不明来源复制“永久关闭安全限制”的命令。

## 12 Docker 路线：可选

已经熟悉 Docker 的读者可以使用：

```
cp .env.example .env
# edit .env and add one provider API key
docker compose run --rm tradingagents
```

本地 Ollama 模型：

```
docker compose --profile ollama run --rm tradingagents-ollama
```

### 核心概念

零基础读者优先使用 Conda。Docker 能减少环境差异，但会增加容器、文件权限和网络映射等新概念，不一定更“简单”。

## 13 安装后先冻结一个可运行版本

第一次成功运行后，记录当前版本：

```
git rev-parse --short HEAD
python --version
pip show tradingagents
pip freeze > requirements_working.txt
```

重要演示前不要直接执行未经测试的：

```
git pull
pip install --upgrade --upgrade-strategy eager .
```

## 第 IV 部分 配置模型与数据源

### 14 只选择一个模型 Provider 开始

TradingAgents 支持多种 Provider。初学者不要同时配置全部平台。

| 使用环境     | 可选 Provider                                 | 建议                    |
|----------|---------------------------------------------|-----------------------|
| 中国大陆网络   | DeepSeek、Qwen China、GLM China、MiniMax China | 优先选择官方中国区端点,降低跨境网络不稳定 |
| 香港、台湾或海外 | OpenAI、Anthropic、Google、xAI、OpenRouter 等    | 比较价格、地区可用性和速率限制       |
| 本地运行     | Ollama                                      | 隐私较好,但模型能力、内存和速度取决于本机 |
| 企业环境     | Azure OpenAI、AWS Bedrock 等                  | 需要额外企业配置文件和权限         |

### 15 编辑.env

在 TradingAgents 项目根目录打开 .env，只填写正在使用的平台：

```
# Fill only the provider you use
OPENAI_API_KEY=
GOOGLE_API_KEY=
ANTHROPIC_API_KEY=
XAI_API_KEY=
DEEPSEEK_API_KEY=
DASHSCOPE_API_KEY=
DASHSCOPE_CN_API_KEY=
ZHIPU_API_KEY=
ZHIPU_CN_API_KEY=
MINIMAX_API_KEY=
MINIMAX_CN_API_KEY=
OPENROUTER_API_KEY=
ALPHA_VANTAGE_API_KEY=
```

例如只使用 DeepSeek：

```
DEEPSEEK_API_KEY=your_real_key_here
```

#### 风险与注意事项

不要在 Key 两侧加入中文引号；不要保留 `your_real_key_here` 这样的占位符；不要把 Key 直接写入 Python 文件。

### 16 模型怎么选

TradingAgents 通常会区分：

- **Quick-thinking model**: 负责较轻量、较频繁的任务；
- **Deep-thinking model**: 负责研究经理、交易与组合决策等复杂推理；

- **Research depth:** 控制整体研究复杂度；
- **Debate rounds:** 控制 Bull/Bear 和风险团队讨论轮数。

第一次运行建议：

- Quick model 选择成本较低的模型；
- Deep model 选择同一 Provider 中较稳定的模型；
- Research depth 选择浅层；
- Debate rounds 设为 1；
- 一次只研究一只股票。

#### 风险与注意事项

模型 ID 和价格会变化。以你当前 CLI 列出的模型为准，不要从旧教程复制已经下线的模型名称。

## 17 数据源的基本认识

默认配置通常使用 yfinance 获取价格、技术指标、基本面和新闻数据。价格数据不需要额外 Key；部分替代数据源可能需要 Alpha Vantage Key。

#### 如何阅读结果

不同数据源对不同市场的覆盖并不均匀。美股的价格、公司信息和新闻通常更完整；A 股可能只有价格覆盖相对稳定，而基本面、新闻和情绪存在缺口。数据为空时，正确答案是“数据不足”，不是让大模型自行补齐。

## 第 V 部分 第一次完整运行

### 18 先用美股做 Smoke Test

第一次不要直接挑战冷门 A 股。先用数据覆盖通常较好的 AAPL 或 NVDA 检查软件、模型和网络是否正常。

### 照着做

1. 打开终端并进入项目：

```
conda activate tradingagents  
cd ~/Projects/TradingAgents
```

Windows 可使用：

```
conda activate tradingagents  
cd C:\Projects\TradingAgents
```

2. 启动交互式 CLI：

```
tradingagents
```

若命令不可用：

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

3. 在界面中输入 AAPL。
4. Analysis Date 选择一个已经收盘的过去交易日。
5. 选择已配置 Key 的 Provider。
6. Quick 和 Deep Model 先选成本较低、CLI 明确支持的模型。
7. Output Language 选择 Chinese 或简体中文。
8. Research Depth 选浅层，Debate Rounds 设为 1。
9. 确认参数并开始运行。

## 19 为什么分析日期必须谨慎

建议选择：

- 已经过去的交易日；
- 不是周末或交易所假期；
- 不晚于数据供应商可获得的最新日期；
- 做历史研究时，明确区分“分析日期”和“今天抓取新闻的日期”。

### 风险与注意事项

历史价格可以按日期截取，但新闻和网页数据未必能完整还原当时的信息集。历史回看很容易混入未来信息，这就是前视偏差。

## 20 运行时你会看到什么

界面通常会依次显示：

1. 数据工具调用；
2. 市场、基本面、新闻和情绪分析师报告；
3. Bull 与 Bear 的正反方辩论；

- 4. Research Manager 的综合；
- 5. Trader Agent 的研究方案；
- 6. 风险团队讨论；
- 7. Portfolio Manager 的最终评级和理由。

成功标志

一次成功的 Smoke Test 应至少出现分析师报告、Bull/Bear 讨论、风险审查和最终组合经理结论，并且没有 API Key、模型名称、ticker 或数据下载错误。

## 21 不要因为界面停顿而重复启动

多 Agent 流程会连续调用模型和数据工具，某个节点可能停留较久。处理顺序：

- 1. 观察终端是否仍有新日志；
- 2. 等待当前 API 调用返回；
- 3. 检查 Provider 后台是否出现请求；
- 4. 确认没有 401、429、timeout 或 connection error；
- 5. 不要连续按 Enter 或开启第二个相同任务。

## 第 VI 部分 运行 A 股、港股与其他市场

### 22 Ticker 的交易所后缀

TradingAgents 默认数据接口使用 Yahoo Finance 风格 ticker：

| 市场     | 示例        | 说明                |
|--------|-----------|-------------------|
| 上海 A 股 | 600519.SS | 贵州茅台；Yahoo 常用 .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      | 一般不加交易所后缀         |
| 日本股票   | 7203.T    | 丰田汽车              |

特别注意 Tushare 与 Yahoo 的上海后缀不同：

|                |           |
|----------------|-----------|
| Tushare:       | 600519.SH |
| Yahoo Finance: | 600519.SS |

## 23 第一次 A 股运行

### 照着做

1. 重新启动 TradingAgents。
2. Ticker 输入 600519.SS 或 000858.SZ。
3. 选择一个已经收盘的交易日。
4. 继续使用浅层研究和 1 轮辩论。
5. 检查公司名称是否与代码一致。
6. 检查价格、基本面、新闻和情绪是否都有有效数据。
7. 若某项数据明显缺失，在自己的研究记录中标注“低置信度”。

## 24 A 股结果为什么更需要人工检查

常见限制包括：

- 海外数据源对中国公司公告覆盖不完整；
- 中文新闻可能被摘要、翻译或延迟；
- 财务字段口径可能与国内数据库不同；
- 公司名称、交易所后缀或历史证券代码可能发生变化；
- 情绪分析可能基于很少的样本。

### 风险与注意事项

价格数据能够下载，不代表基本面、新闻和情绪数据同样完整。某个 Agent 输出很长，也不代表它掌握的证据很多。

## 第 VII 部分 如何阅读多 Agent 研究结果

### 25 正确阅读顺序

不要先跳到最后一行看 Buy 或 Sell。建议按以下顺序：

1. **身份**：ticker、公司名称和交易所是否正确；
2. **日期**：分析日期、财务期和新闻时间是否一致；
3. **市场数据**：价格、成交量和技术指标是否合理；
4. **基本面**：财务与估值数据是否有来源和时间戳；
5. **新闻与情绪**：样本是否足够，是否混入分析日期之后的信息；
6. **Bull/Bear**：双方是否回应了对方的核心证据；
7. **风险**：是否讨论波动、流动性、估值、数据缺口和仓位；
8. **最终评级**：是否附带失效条件，而不是只有一个方向词。

## 26 五个必问问题

每次运行后，自己写下：

1. 哪三项是可以回到原始数据核实的事实？
2. 哪三项只是模型推断或叙事？
3. 哪个 Agent 缺少数据却仍然说得很肯定？
4. Bull 与 Bear 是否使用相同日期和相同口径？
5. 什么新证据出现时，最终结论必须改变？

## 27 识别“伪辩论”

真正有价值的 Bull/Bear 辩论应当：

- 引用对方刚才提出的具体证据；
- 指出证据口径、时间或因果关系的问题；
- 承认本方观点的弱点；
- 提出可验证的失效条件；
- 最终缩小分歧，而不是重复两篇独立作文。

### 如何阅读结果

如果 Bull 说“公司长期优秀”，Bear 说“市场存在风险”，双方都没有讨论数字、日期和估值，这不是研究辩论，只是两位 Agent 很有礼貌地各说各话。

## 28 最终评级应该怎样理解

当前框架可能使用 Buy、Overweight、Hold、Underweight、Sell 等分级。它们表示 Agent 在特定输入下的研究判断，不等于：

- 适合你的仓位建议；
- 对未来收益的概率保证；
- 可以忽略价格、估值和交易成本；
- 可以直接发送给券商执行的订单。

## 第 VIII 部分 控制成本、保存记录与恢复运行

### 29 API 成本如何产生

成本大致随以下项目上升：

$$\text{成本} \approx \text{Agent 数量} \times \text{辩论轮数} \times \text{上下文长度} \times \text{模型单价}$$

第一次练习建议：

- 一次只研究一只股票；
- Research Depth 选择浅层；
- Debate Rounds 设为 1；
- Quick Model 使用便宜模型；
- 不因界面停顿而重复启动；
- 在 Provider 后台设置用量上限或告警。

为什么这样做

多 Agent 开会不收会议室租金，但每个发言都可能按 token 计费。让十个 Agent 开三轮会，通常比人类会议更准时，也更会花钱。

30 启用 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/
```

成功完成后，旧 checkpoint 通常会自动清理。中断后再次运行相同任务时，日志应显示从已完成步骤继续，而不是全部重来。

31 Decision Log

TradingAgents 会保存历史决策，并在后续同一 ticker 的研究中加入过去结果和反思。默认日志常见位置：

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

风险与注意事项

历史记忆不是“模型学会赚钱”的证明。它可能改善一致性，也可能把过去的错误叙事带入下一次研究。定期阅读并备份 Decision Log，而不是把它当成不可质疑的长期记忆。

32 建立自己的研究记录

每次运行至少记录：

| 字段            | 示例                         |
|---------------|----------------------------|
| Ticker        | 600519.SS                  |
| Analysis date | 2026-06-30                 |
| 运行时间          | 2026-07-2120:30            |
| 代码版本          | Git commit 短码              |
| Provider      | DeepSeek / OpenAI / Qwen 等 |

|                  |                         |
|------------------|-------------------------|
| Quick/Deep model | 运行时实际模型 ID              |
| Research depth   | Shallow / Medium / Deep |
| Debate rounds    | 1                       |
| 数据缺口             | 新闻不足、基本面字段缺失等           |
| 最终评级             | Agent 原始输出              |
| 人工结论             | 接受、保留、拒绝及理由             |

### 33 同一输入为什么会出现不同结果

即使 ticker 和日期相同，结果仍可能变化，原因包括：

- 大模型生成具有随机性；
- 实时新闻和网页内容发生变化；
- Provider 模型后台版本更新；
- 数据缓存或供应商返回内容变化；
- Decision Log 给后续运行加入了不同历史上下文。

正确比较方式是比较**事实、证据和关键假设**，而不是逐字比较两篇报告。

## 第 IX 部分 使用 Python 自定义运行

### 34 最小 Python 示例

在项目根目录建立 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["checkpoint_enabled"] = True
config["output_language"] = "Chinese"

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

运行：

```
conda activate tradingagents
cd ~/Projects/TradingAgents
python my_analysis.py
```

Windows：

```
conda activate tradingagents
cd C:\Projects\TradingAgents
python my_analysis.py
```

### 风险与注意事项

示例中的模型 ID 是占位符，必须替换成当前 Provider 实际支持的模型。API Key 仍写在 `.env`，不要硬编码进 Python 文件。

## 35 配置默认 yfinance 数据源

```
config["data_vendors"] = {
    "core_stock_apis": "yfinance",
    "technical_indicators": "yfinance",
    "fundamental_data": "yfinance",
    "news_data": "yfinance",
}
```

如果切换到其他数据源，先确认：

- 是否需要额外 API Key；
- 是否支持目标市场和历史日期；
- 财务字段和复权口径是否一致；
- 速率限制是否足以支持多 Agent 连续调用。

## 36 修改参数时一次只改一个变量

推荐实验顺序：

1. 固定 ticker、日期和数据源；
2. 先改变 Debate Rounds；
3. 再改变 Deep Model；
4. 最后改变 Research Depth；
5. 每次保存输出和参数；
6. 判断变化来自更多证据，还是只是语言更长。

## 37 一个可复用的研究任务模板

每次分析前先写明：

Research question:

Is the current evidence sufficient to support a positive, neutral, or negative view on [TICKER] as of [DATE]?

Required checks:

1. Verify ticker, company name, exchange, and date.
2. Separate facts, estimates, and model inference.
3. Identify missing or stale data.
4. State the strongest bull and bear evidence.
5. Provide invalidation conditions.
6. Do not convert the conclusion into a real trade order.

## 第 X 部分 常见问题与故障排查

### 38 安装与环境问题

| 现象                | 常见原因               | 处理                                                       |
|-------------------|--------------------|----------------------------------------------------------|
| git 不存在           | Git 未安装或终端未重开      | 安装 Git，关闭并重新打开终端                                         |
| conda 不存在         | Miniconda 未安装或未初始化 | 使用 Anaconda Prompt，或重新初始化 Conda                          |
| 终端仍显示 base        | 没有激活项目环境           | 运行 <code>conda activate tradingagents</code>             |
| tradingagents 不存在 | 环境未激活或安装失败         | 在项目目录运行 <code>pip install .</code>                       |
| Module not found  | 使用了错误 Python       | 检查 <code>which python</code> 或 <code>where python</code> |
| 依赖冲突              | 与其他项目共用环境          | 删除旧环境并单独重建 tradingagents                                 |
| Windows 中文乱码      | 终端编码或旧版本问题         | 更新项目，使用 UTF-8 终端并重开窗口                                    |

### 39 API 与模型问题

| 现象               | 常见原因                        | 处理                     |
|------------------|-----------------------------|------------------------|
| API Key missing  | <code>.env</code> 未填写或变量名错误 | 使用官方变量名，保存后重开终端        |
| 401 Unauthorized | Key 错误、失效或端点不匹配             | 重新生成 Key，确认国内/国际端点     |
| 429 Rate limit   | 请求过多、余额不足或并发过高              | 降低深度和轮数，检查额度           |
| Model not found  | 模型 ID 已下线或拼写错误              | 使用 CLI 当前列出的模型         |
| 连接超时             | 网络、代理或 Provider 服务异常        | 测试网络，切换 Provider 或稍后重试 |
| 本地 Ollama 很慢     | 模型过大或硬件不足                   | 换小模型、减少深度或使用云端模型       |

### 40 数据与 ticker 问题

| 现象                        | 常见原因                                         | 处理                                             |
|---------------------------|----------------------------------------------|------------------------------------------------|
| A 股 ticker 无数据<br>公司名称不匹配 | 上海误用了 <code>.SH</code><br>ticker 输入错误或证券代码变更 | Yahoo 风格上海改为 <code>.SS</code><br>立即停止，先核对交易所官网 |
| 历史日期失败                    | 周末、未来日期或数据源缺失                                | 改为已收盘的交易日                                      |
| 角色报告为空                    | 对应数据源覆盖不足                                    | 标记缺失，换 ticker 或数据源                             |

|           |               |              |
|-----------|---------------|--------------|
| 新闻出现未来信息  | 历史新闻没有严格按日期冻结 | 降低置信度并人工核对时间 |
| 价格与其他平台不同 | 复权、时区或币种口径不同  | 对齐日期、复权方式和币种 |

## 41 运行中断与结果异常

| 现象             | 常见原因                               | 处理                                                       |
|----------------|------------------------------------|----------------------------------------------------------|
| 流程中断           | 网络或长流程失败                           | 使用 <code>--checkpoint</code> 续跑                          |
| 再次运行从头开始       | 未启用 <code>checkpoint</code> 或参数不一致 | 使用相同 <code>ticker</code> 、日期和 <code>checkpoint</code> 设置 |
| 结果每次不同         | 模型和实时数据非确定性                        | 固定输入并保存版本；比较证据                                           |
| 报告很长但没有数字      | 数据工具失败后模型继续叙述                      | 将该部分标记为不可用                                               |
| Bull/Bear 各说各话 | 辩论未真正回应对方证据                        | 人工提取分歧，不增加轮数掩盖问题                                         |
| 最终评级过度肯定       | 组合经理忽略数据缺口                         | 降低置信度，补充失效条件                                             |

# 第 XI 部分 最终检查表与速查命令

## 42 最终自检清单

| 我已经做到 |                                                       |
|-------|-------------------------------------------------------|
| ✓     | TradingAgents 使用独立 Conda 或 venv 环境。                   |
| ✓     | 我只在 <code>.env</code> 中保存 API Key，没有公开上传。             |
| ✓     | 我能用 AAPL 或 NVDA 完成 Smoke Test。                        |
| ✓     | 我知道如何在 CLI 中选择中文输出、模型、深度和辩论轮数。                        |
| ✓     | 我知道上海 A 股使用 <code>.SS</code> ，深圳使用 <code>.SZ</code> 。 |
| ✓     | 我会先检查 <code>ticker</code> 、日期和数据，再看最终评级。              |
| ✓     | 我能区分可核实事实、模型推断和未知信息。                                  |
| ✓     | 我会控制研究深度、辩论轮数和 API 成本。                                |
| ✓     | 我知道如何启用 <code>checkpoint</code> ，并能找到 Decision Log。   |
| ✓     | 我会保存 Provider、模型、参数、版本和运行日期。                          |
| ✓     | 我不会把 Buy、Hold 或 Sell 直接当成真实交易指令。                      |

## 43 常用命令速查

### 43.1 安装

```
git clone https://github.com/TauricResearch/TradingAgents.git
cd TradingAgents
conda create -n tradingagents python=3.13 -y
conda activate tradingagents
python -m pip install --upgrade pip
pip install .
```

## 43.2 启动

```
tradingagents
# alternative
python -m cli.main
```

## 43.3 Checkpoint

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

## 43.4 检查环境

```
python --version
pip show tradingagents
git rev-parse --short HEAD
```

## 43.5 更新前先备份

```
pip freeze > requirements_working.txt
git status
git log -1 --oneline
```

# 44 官方资料与版本说明

- 官方仓库: <https://github.com/TauricResearch/TradingAgents>
- 官方论文: <https://arxiv.org/abs/2412.20138>
- 版本记录: <https://github.com/TauricResearch/TradingAgents/releases>
- 主要配置: 项目中的 `tradingagents/default_config.py`
- 官方示例: 项目根目录的 `main.py`

截至 2026 年 7 月, 本手册依据 TradingAgents v0.2.5 系列官方资料编写。该项目更新较快; 若菜单、模型列表或参数发生变化, 以官方 README、CHANGELOG.md、Releases 和当前代码为准。

# 45 结语

### 核心概念

正确使用多 Agent 的关键, 不是让更多模型一起替你决定, 而是把研究拆成可核验的步骤: 数据来源明确、日期口径一致、正反观点可追溯、风险条件可执行、最终由人负责。

### 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。