

第六步：策略开发 – AI辅助代码生成与文件管理

在完成前期因子研究后，我们进入策略代码开发阶段。本阶段重点利用大语言模型（LLM）的协助来生成策略代码、优化参数，并通过良好的文件管理和模块化设计保证策略逻辑清晰可维护。开发者将与LLM协同编程，以更高效、模块化地实现策略逻辑，并引入**投资标的候选池**模块，确保策略不遗漏市场热点标的。

模块化代码结构与文件组织

首先，将策略代码按功能拆分为多个模块文件，存储在项目目录下，便于协作开发和版本管理。典型的模块划分包括：

- **主策略模块**（如 `strategy.py`）：包含平台所需的接口函数（例如聚宽平台的 `initialize()`、交易函数 `handle_data()` 或 `before_trading()` 等）。这里实现交易逻辑、调仓规则、仓位管理和风险控制。
- **因子计算模块**（如 `factors.py`）：封装第五步输出的因子计算代码，为策略提供选股打分依据。
- **工具函数模块**（如 `util.py`）：存放通用辅助函数（如日志、日期处理、数据加载等），提高代码复用性。
- **配置文件**（如 `config.yaml` 或 `.json`）：定义策略参数的初始设定（如指标阈值、仓位上限等）和基础设置（如回测时间段、交易标的的范围）。这些参数在需要时可由策略优化器批量调整。
- **候选池模块**：本步骤新增的模块，用于管理“投资标的候选池”。我们将其进一步细分为两个子模块，以清晰区分职责：
 - **候选标的获取模块**（负责多渠道搜集市场热点标的）：根据主线热点、技术突破、ETF轮动、外部推荐等来源定期抓取候选股票列表。
 - **策略候选池模块**（负责整合和筛选最终策略可交易池）：将获取的候选标的与策略既定的交易范围结合，应用一定过滤条件，产出策略最终考虑的标的池。

各模块文件通过项目文件系统组织，类似于乐高积木式的低耦合设计，确保修改某一部分时不会影响整体¹²。例如，可以在 `candidate_pool.py` 中定义候选池相关逻辑，在 `strategy.py` 中导入并调用，从而**按顺序衔接**：先更新候选池，再据此进行因子选股和交易决策。采用版本控制（如Git）或保存不同版本文件（如 `strategy_v1.py`，`strategy_v2.py` 等）的方式记录代码演进，确保每次优化迭代都有迹可循。良好的模块化和版本管理能够提高代码透明度和可维护性³。

AI辅助策略代码生成与调试

大语言模型在代码编写过程中扮演智能助手角色，帮助开发者快速构建和完善策略代码：

- **代码生成**：开发者可通过逐步提示，让LLM生成各模块的代码框架。例如，针对聚宽平台，可以提示：“请基于X因子和Y交易规则编写聚宽策略的初始化函数和每日调仓函数”。LLM会产出符合聚宽API接口的代码结构（如包含 `initialize(context)` 和调仓函数的代码片段），加速初始开发⁴。这种做法相当于让LLM充当“聚宽API查询和代码生成器”，为经验不足者提供模板，也为熟悉聚宽的开发者节省手动查询文档的时间。

- **自动调试**：当生成的代码在回测中出现错误时，开发者可以将错误日志反馈给LLM，让其协助定位问题并提出修改建议。LLM具备一定的代码理解和推理能力，能够根据异常信息快速定位bug所在。例如，如果回测日志显示某变量未定义或API调用参数错误，LLM会协助找出遗漏的全局变量声明或纠正API用法，从而缩短调试周期。
- **参数敏感性分析**：在多次回测后，LLM还能根据结果分析策略对参数的敏感度并提出优化建议。例如，若发现策略绩效对某阈值参数非常敏感，LLM可能建议缩小该参数的搜索范围重新网格优化，或者识别出某参数组合在特定市场条件下效果最佳。这种分析结合了LLM对语言和模式的理解，有助于发现人工难以及时察觉的改进方向。
- **自动生成文档**：LLM可以根据最终的策略代码生成策略说明文档。通过提示模型阅读代码并总结策略逻辑，输出Markdown格式的说明，包括策略理念、交易规则、因子说明、参数含义和使用方法等。这确保文档与代码实现一致，并极大地方便了后续维护。文档中也可以包含版本变更记录（changelog），说明每次优化调整了哪些内容。

值得注意的是，越来越多量化平台开始探索将LLM直接集成到策略开发流程中，例如看海量化平台（KHQuant）从设计之初就考虑了与AI结合的可能，使其开放的Python环境和标准化接口便于大模型辅助策略生成¹；又如QMT-MCP项目直接提供了“智能策略生成”功能，可自动生成并优化量化交易策略⁴。这些趋势表明AI协作编程在量化领域大有可为，我们在本策略开发中充分加以利用。

策略参数优化与自动调优

策略开发完成初版后，通常需要对关键参数进行优化。这里我们结合平台自带的优化工具和自定义方法，实现高效的参数调优：

- **平台内置参数优化引擎**：如果使用QMT等本地量化平台，其专业版自带参数优化功能，可指定多个参数范围自动遍历组合进行回测⁵。开发者在策略代码中将某些参数声明为外部可调参数，然后在QMT的回测界面上选择优化选项，设置参数范围和步长。平台将并行运行所有组合回测，输出各组合的绩效指标，帮助找到最佳参数。实际案例显示，利用QMT的多线程优化引擎，对50组参数组合进行遍历只需数分钟，相比纯Python单线程优化大幅提速⁶。
- **一键网格搜索**：借助开源框架的工具也可快速实现参数网格优化。例如BulletTrade框架提供了一条命令完成参数遍历优化的功能⁷。只需在命令行指定策略脚本和参数网格，BulletTrade会自动多进程回测所有组合并给出最优结果。这种“一行命令”式的优化极大地方便了开发者调参。例如，针对某策略的均线周期和阈值两项参数，可用如下命令网格搜索最优组合⁷：

```
bullet-trade optimize your_strategy.py \
  --params '{"ma_period": [5, 10, 20], "threshold": [0.01, 0.02, 0.03]}' \
  --start 2020-01-01 --end 2023-12-31
```

BulletTrade将自动遍历所有参数组合并输出表现最佳的参数组，使开发者迅速锁定较优配置。

- **自定义优化脚本**：对于更复杂的参数空间或自定义指标优化，开发者也可编写Python脚本调用策略回测函数，结合诸如网格搜索、随机搜索或贝叶斯优化算法进行参数调优⁸。例如，使用

`itertools.product` 生成参数组合、调用聚宽的 `run_query` 或QMT的回测API获取结果，再根据目标指标排序筛选。还可以利用开源的优化库（如Hyperopt、Optuna）进行更智能的寻优。这些方法能探索更大的参数空间并避免纯网格的低效，对多参数优化问题尤为有用。

- **LLM辅助解读优化结果**：完成参数优化后，可将不同参数组合的回测结果交由LLM分析。例如让模型查看每组参数的收益率、回撤、夏普比率等，归纳哪些因素导致绩效差异，并用自然语言给出结论。这种分析能帮助开发者理解策略对参数的依赖关系，避免陷入过拟合陷阱⁹。例如，如果LLM发现最优参数仅在历史数据上表现突出，但稍有变动绩效便恶化，它会提醒这是过拟合的征兆，需要谨慎使用优化结果¹⁰。LLM甚至可建议更合理的参数默认值或范围，使策略在未知数据上更健壮。

通过以上工具链，策略员能够在**较短时间内完成参数的自动化调优**。重要的是，在拿到优化结果后，仍需结合金融逻辑进行判断，防止机械套用过拟合的参数。LLM的引入，为参数调优提供了另一层智能建议，有助于找到在统计上显著且在经济上合理的参数配置。

候选池模块的集成与高级选股流程

为了确保策略不遗漏市场中有潜力但可能未被原有因子筛选捕捉到的标的，我们在策略开发环节融入“**投资标的候选池**”机制。这个机制通过独立的代码模块，从多种来源动态获取候选投资标的，并与策略的规则相衔接。具体方案如下：

候选标的获取模块 – 多来源智能抓取

我们设计一个“候选标的获取”模块，定期从**四大来源**收集潜在投资标的，并借助LLM和工具链提高抓取效率和精准度：

1. **主线强势股**：跟踪当前市场的主线热点题材和其中的龙头强势股。主线是指市场资金关注的主要投资逻辑或主题，强势股则是沿该主线持续走强的股票¹¹。例如，当人工智能、新能源等成为市场主线时，相应板块中涨幅领先、获得广泛认同的股票可视为强势股纳入候选池。实现方面，可以每日分析市场资讯和行情：
2. **LLM新闻解析**：利用LLM从财经新闻、研报摘要中提取当日**市场主流投资逻辑**和**出现频率最高的公司名称**。LLM擅长总结非结构化文本，例如输入当日市场评论文章，询问“当前市场的热点板块和龙头个股有哪些？”，模型将返回可能的热点概念（如“AI芯片板块强势，龙头为XYZ”）。通过聚合多篇资讯的结果，可得到主线题材及强势股名单。
3. **行情数据扫描**：直接从行情出发，筛选一段时间内相对强度突出的股票。例如计算过去一个月涨幅排名前1%的股票列表，结合成交额筛选出涨幅大且成交活跃的个股，这些往往对应市场主线中的强势票。还有一种经验做法是浏览每日涨幅榜前列股票，关注反复上榜者¹²。在实现中，可用聚宽 `get_fundamentals` 或QMT的API获取当日涨跌幅榜单，将前若干名股票记录并与主题分析结果交叉验证，提取重合的热点龙头。
4. **技术突破股**：捕捉出现关键技术形态突破的股票，补充单纯基本面筛选的不足。技术突破通常指股价突破重要阻力位、新高、或摆脱长期盘整区间等。这类股票往往在技术面上转强，具有短期趋势跟随的意义。获取方法包括：
5. **区间涨幅和高点突破**：编写算法扫描全市场股票，找出近期涨幅超过某阈值且创出**阶段新高**的标的。例如“从最近低点上涨超过25%且突破半年内高点”可视为一个突破条件¹³。符合条件的股票进入“突破股票池”候选列表¹³。也可使用技术指标如MACD金叉、均线多头发散等信号作为突破判断条件。

6. **量价配合监控**：选取最近出现**放量上涨**的个股作为突破线索。通过计算量比、换手率等，发现那些价格上升伴随成交量显著放大的股票。成交放大表示资金关注度提升，若同时股价突破重要阻力位，基本符合“技术突破”的定义。这类标的抓取可借助平台API批量获取日K线数据并计算技术指标，将满足条件者列入候选。
7. 对这些技术突破股，需要进一步**验证其持续性**。可以将它们加入候选池后持续观察几日，若能站稳突破位或继续放量上涨，则在策略选股时重点考虑；若很快回落，则从候选池剔除，以免策略被假突破迷惑。
8. **ETF轮动信号**：利用ETF基金的轮动揭示市场风格切换，进而选取受益板块的龙头股。ETF轮动策略通过分析各类宽基或行业ETF的强弱，追逐阶段领涨的资产⁸。我们将此思想融入候选池：
9. **ETF强度排名**：定期（如每周或每月）计算主要指数ETF和行业ETF的动量指标（如过去N日收益率、夏普比率等），挑选**强势上升**的ETF⁸。例如发现某段时间科技类ETF表现位居前列，则表明资金正青睐科技板块。
10. **板块轮动候选**：针对强势ETF对应的板块，提取其成份股或行业内龙头公司，纳入候选池。例如，若新能源车ETF近期领涨，则将新能源车板块的头部几只股票加入候选列表。聚宽的`get_index_stocks`或QMT的行业成分股API可用于获取ETF对应指数或行业的成份股票列表¹⁴¹⁵。
11. **轮动信号退出**：同样，当某主题ETF的动量开始衰减（排名下滑），应及时从候选池中移除相关板块的个股，避免策略追逐过气主题。通过设定动量阈值或排名滑落标准，实现候选池的**动态更新**。
12. **外部推荐**：综合外部智能推荐和研究员人工筛选的标的，确保候选池来源多元化。外部推荐包括：- 13. **AI模型推荐**：利用训练好的选股模型或LLM，输入大量股票特征让模型输出**看好的股票名单**。例如一些开源的因子选股模型、ChatGPT等可以根据财务和新闻数据生成股票评级。虽然这类推荐存在不确定性，但作为参考可以丰富思路。
- 14. **资讯研报线索**：定期抓取券商研报和投资组合推荐。如果多家机构都在研报中提及某些股票具备投资潜力，可将其列入观察名单。LLM可以被用来批量阅读研报摘要，提取其中反复出现的股票及理由，从而形成**研报推荐池**。
- 15. **策略员人工加入**：策略开发人员基于自身经验或直觉，额外指定一些候选标的（例如新上市的龙头公司、政策大力扶持的新兴产业公司等）。这些人工加入的标的可以通过配置文件列出，候选池模块启动时自动读取合并。这样一来，策略员的主观判断与客观算法形成互补。

通过上述多源头获取，我们每日/每周会得到若干子列表，如“主线强势股名单”、“技术突破股名单”、“热点板块龙头股名单”、“外部推荐股名单”等。**候选标的获取模块**可以将这些来源的结果存储为中间文件或数据结构（如Python列表、JSON文件），供下游使用。同时要注重**抓取频率和效率**，例如热点题材和强势股可以**每日**更新，ETF轮动和研报推荐则**每周**更新，以避免过度频繁导致噪音增加或性能开销过大。整个设计力求**有针对性**：每个来源都有明确目的，避免无效数据；**高效**：充分利用API和LLM的批处理能力，减少人工重复劳动；**合理**：对纳入候选的标的都设定基本门槛（如市值流动性要求、信噪比判断），确保候选池质量而非罗列过多杂乱股票。

策略候选池模块 – 合并筛选与流程衔接

在完成多来源的数据抓取后，接下来通过**策略候选池模块**将这些候选标的进行整合，形成最终供策略选股决策使用的**策略标的池**。关键步骤包括：

- **合并与去重**：将不同渠道获得的候选列表合并成统一的候选池。例如，将主线强势、技术突破、ETF板块龙头、外部推荐四类列表取并集。很多股票可能出现在多个列表中，通过集合运算去重，并可以统计每只股票来自哪些来源作为参考（例如某股票同时属于强势+突破，两项标签意味着其更值得关注）。

- **基础过滤**：对合并后的候选池应用基础过滤条件，剔除不符合策略基本要求的股票。例如，排除停牌股、日均成交额过低的股票、涨跌停被限制无法交易的股票等。这一步可利用平台API查询股票的实时状态和流动性指标，保持策略候选池的可交易性¹⁶。
- **与策略适配**：这里是模块衔接的重心：策略可能本身限定了投资范围（如只做沪深300成份股、或仅交易特定行业），因此需要将候选池与策略既定的**标准股票池**取交集，得到**最终策略标的池**。例如，如果策略明确只交易沪深300指数成份股，那么候选池中的股票也需限制为沪深300内的标的¹⁴。通过调用指数成份获取函数¹⁴交叉过滤，确保策略池符合投资范围规定。
- **策略适配的深度研究算法**：这里强调不能只是用硬编码的配置文件列出股票池，而应通过算法动态获取并调整。例如，若策略要求适配不同**标准池**（沪深300、中证500、创业板等指数成份），可在配置中指定池的类别，然后由代码查询相应成份股列表，实现**自动更新**（避免成份变动时配置过期）。又或者策略需要在不同市场环境切换投资池（比如震荡市采用价值股池，趋势市采用成长股池），可以内置**条件判断逻辑或因子筛选逻辑**，根据市场指标信号切换候选池构成。这些都是比简单配置文件更“聪明”的适配方式，体现深度研究和算法的作用。
- 举例来说，一种深度适配做法是**动态行业筛选**：策略平时运行在全市场范围，但如果某段时期市场呈现结构性行情，则策略模块可调用行业轮动信号，自动聚焦领先行业的股票池。当检测到当前市场仅有少数行业领涨时，策略标的池自动收缩到这些行业，以提高胜率；反之当行情全面开花时，再扩展回全市场池。这种算法化调整由代码根据指标触发，不需要人工修改配置文件，体现策略对标准池的智能适配。
- **候选打分和最终选股**：经过以上合并和过滤，我们得到当期的策略候选标的池。这一池即是策略接下来进行因子打分和选股决策的基础范围。此时策略会调用**因子计算模块**，但**仅对候选池内的股票**计算各项因子分数，避免无关股票的计算开销，并确保选股聚焦在精挑细选的范围。然后按策略逻辑（比如多因子综合评分、打分排序择前N名、或基于打分的买入信号）产生最终的交易决策。
- **调仓执行**：策略标的池输出买卖信号后，在交易执行部分，代码会针对这些标的下单，控制仓位和风控。若标的池为空或信号不强，则策略可以选择空仓休息。标的池模块也可以提供辅助信息给交易决策，例如某股票属于主线强势且基本面优异，策略可对其赋予更高权重或更大持仓上限；再如某股票仅因技术突破入池但基本面平平，策略可设定更谨慎的仓位。

通过这样的流程，我们将**候选池模块无缝融入策略开发与运行环节**：先由候选标的获取模块抓取多源信息，再经策略候选池模块融合筛选，最终为因子选股和交易提供优化过的标的范围。整个过程在策略代码的**初始化阶段或每日盘前**执行，保证在交易判断前候选池是最新鲜可靠的。例如在 `initialize(context)` 中调用候选池更新函数，或每日定时任务(`before_trading`)中刷新候选池数据。

需要强调的是，引入候选池模块后，策略逻辑变得更加**智能和全面**。它不仅依赖定量的因子规则，也结合了定性和外部信息，使策略拥有“扫描市场雷达”的功能。这在实战中可有效弥补量化模型的盲区——例如纯因子模型可能错过突发政策利好下快速上涨的概念股，但通过候选池的新闻抓取，此类标的会被及时纳入关注范围。正如有经验的基金经理所言，“**强势股+基本面**”是选股的重要框架：先跟随市场共识找出强势主线股票，再筛选基本面优异者作为投资候选¹⁷¹³。我们的候选池机制正是对此的自动化实现，确保策略既不盲从热点也不漏掉机会，在系统化和灵活性之间取得平衡。

输出成果：完整策略代码与配套文档

经过策略开发阶段，我们最终产出**完整的策略代码和配套文档**，为下一步的回测验证做好准备。具体包括：

- **可执行的策略脚本**：代码符合目标交易平台的接口规范，可以直接运行于回测或实盘环境。例如，针对聚宽平台，代码包含 `initialize(context)` 初始化函数和交易逻辑函数，如每日收盘前调仓的 `handle_data(context)` 或定时调仓的 `run_daily()` 调度¹⁸。对于QMT/PTrade平台，策略代码实现其所需的 `init()`、`handlebar()` 等接口。所有模块（因子、候选池、风控等）在主策略脚本中有机结合，逻辑清晰。代码已经过充分测试，能正确获取数据、计算指标并下单交易。

- **模块化的工程结构**：项目目录清晰包含各模块文件。比如：

```
strategy_project/
├─ strategy.py      # 主策略代码
├─ factors.py       # 因子计算模块
├─ candidate_pool.py # 候选池模块（含子模块功能）
├─ util.py          # 工具函数
├─ config.yaml      # 策略参数配置
├─ README.md        # 策略说明文档
└─ changelog.md     # 版本更新记录
```

这种结构方便后续移植到其他平台或共享给他人使用。例如借助BulletTrade框架，几乎无需修改即可将聚宽策略代码移植到本地实盘环境运行¹⁹²⁰。

- **版本管理与变更记录**：伴随代码，我们提供版本化的变更记录文档。例如，从v1.0到v2.0优化了哪些参数、调整了哪些逻辑、提高了哪些性能指标，一一记录在 `changelog.md` 中。这种版本记录便于团队协作时了解历史修改，也方便将来回溯某次改动对策略效果的影响。在大型更新前，我们也可保留旧版本代码文件（如 `strategy_v1.py`），以防新版本不及预期时快速回退。遵循Git版本控制最佳实践，每次提交附带清晰注释³，保证策略开发过程可追溯、可审计。
- **策略说明文档**：提供一份详细的Markdown文档（如 `README.md`），内容包括：策略背景与设计思路、策略逻辑概述（交易规则、选股因子、调仓频率等）、**候选池机制**介绍、主要参数列表及含义、回测主要绩效指标摘要，以及使用方法指南（如何在平台上部署运行、本地运行需满足的依赖等）。说明文档还应体现我们的开发创新点，例如如何利用LLM辅助开发、如何引入候选池提高策略有效性等，使读者充分理解策略的特色和优势。

通过第六步的努力，策略代码已经**高质量完成**：逻辑经由AI和人类多次校验，模块划分明确且易扩展移植；参数优化得当不overfit；文档完善方便交流和复审。这为后续步骤（第七步的历史回测和第八步的实盘仿真/上线）奠定了坚实基础。开发者和策略员此时对策略的方方面面（从模型原理到代码实现）都有了深入掌握，配合完善的文件管理和说明文档，整个策略开发流程实现了专业化、智能化和可控化。

最后值得一提的是，这一阶段我们有效结合了**人机智能**：利用大模型的创意和效率，加上开发者的专业判断和经验，两者优势互补，产出比纯人工或纯机器都更优的成果。这正是AI时代策略开发的新范式，也将持续提升量化投

资的研发效率和策略表现。随着完整策略代码和文档的交付，我们即将进入策略验证阶段，通过严格的历史回测来检验本策略在不同市场环境下的表现，并据此进行最后的调优完善。

1 3 1.1 引言 - khQuant看海量化交易平台

<https://khsci.com/khQuant/chapter1/>

2 7 18 19 20 聚宽策略想实盘？这个开源项目让你一行代码不改直接跑 - bigbencat - 博客园

<https://www.cnblogs.com/bulletrade/p/19308512>

4 GitHub - guangxiangdebizi/QMT-MCP: QMT-MCP 模块化量化交易助手

<https://github.com/guangxiangdebizi/QMT-MCP>

5 6 震惊！策略回测参数优化从 500 多秒提升到 1 秒，竟然这么简单？ - QMT投研数据服务 - 迅投QMT社区 - Powered by Discuz!

<https://www.xuntou.net/forum.php?mod=viewthread&tid=397&mobile=2>

8 ETF轮动策略创建教程 - 次方量化

<https://quant.lindercube.com/strategy-guide>

9 10 Hikyuu教程 | 如何进行策略回测参数优化-CSDN博客

<https://blog.csdn.net/KongDong/article/details/143061204>

11 12 13 17 蜂巢基金徐增：善用市场共识的力量 | 蜂巢 | 强势股 | 基金 | 蜂巢 | 蜂巢基金_手机新浪网

https://finance.sina.cn/fund/jjyj/2024-03-25/detail-inapnwew1868630.d.html?vt=4&cid=77033&node_id=77033

14 基于聚宽量化交易平台实现量化交易策略 - 休耕 - 博客园

<https://www.cnblogs.com/xiugeng/p/12251232.html>

15 16 迅投QMT Python文档

https://zilch Yao.github.io/xuntou_yao/