可解释 Alpha 因子生成智能体

点时间对齐、受限表达式引擎与确定性回测的可信底座

An Explainable Alpha-Factor Generation Agent: Point-in-Time Alignment, a Restricted Expression Engine, and a Fully Deterministic Backtest

cumulative rank IC · out-of-sample

RESEARCH TEAM

周梓煜（组长） 张心湉 许艺馨 张修博 刘迅羽 王佳祺

LLM 提出假设，确定性代码作出裁决。

The LLM proposes hypotheses; deterministic code renders the verdict.

2026 年 7 月 21 日

v1.0 · synthetic-data build

摘 要

本文提出并实现一个**可解释 Alpha 因子生成智能体**，用以回答一个具体的研究问题: 大语言模型（LLM）能否在*不参与任何数值计算*的前提下，稳定地生成具备经济解释的 A 股 Alpha 因子。系统遵循“**LLM 提出假设、确定性代码负责计算与验证**”的分工范式——语言模型仅用于生成候选因子、辅助语义查重与阅读回测结果，所有关于收益、IC 与稳定性的数字一律由本地 Python 程序产生，从架构上杜绝“数值幻觉”与回测泄漏。我们的贡献有三:(i) 给出行情、财务与股票池三表的**数据契约**，并把防前视（no-look-ahead）约束形式化为点时间（point-in-time, PIT）合并算子 `pit_merge`，配以可证伪的单元测试;(ii) 设计一个仅允许白名单算子的**受限表达式引擎**，在支持因子可组合的同时禁止任意代码执行，使“可解释”与“可审计”落到实处;(iii) 构建一条**完全无 LLM 的确定性回测流水线**，产出 Rank IC、ICIR、分组收益、多空收益、换手率与成本后收益，并按日期滚动完成样本外划分。在聚源 SQL Server 登录验证与 JYDB 元数据快照（2739 张表、81757 个字段）完成后，本文进一步导出行情、财务与股票池三张 raw 表，并在 2020–2021 年沪深 A 股 price-observed 股票池上重跑确定性流水线。初步样本外结果显示，部分基线因子在 2021 年具备正 Rank IC 与 ICIR，但两只 mock 生成候选因子未表现出稳健正向样本外收益，其中 `value_profit_blend` 还与既有基线高度相关。因此，本文的实证结论被限定为: 可信底座已经能够在聚源真实数据上可复现运行，并能拒绝证据不足或新颖性不足的候选因子，而非宣称发现稳定 alpha。

关键词： Alpha 因子; 可解释性; 点时间对齐; 样本外回测; 大语言模型; 确定性计算; 防前视

目 录

1 引言

Alpha 因子研究的核心难点，从来不是让模型“说出一句听起来很聪明的话”，而是能否把一个候选假设转化为可计算、可验证、可复现的研究对象。对量化研究而言，任何关于收益、信息系数（IC）或稳定性的结论，都必须来自可审计的数值流程，而不能是语言模型的口头裁定。近年来大语言模型被大量引入因子挖掘 [3, 4]，但一旦让模型直接“判断因子好坏”或“参与回测计算”，就会同时引入两类风险: **数值幻觉**（模型编造并不成立的统计量）与**回测泄漏**（在 T 日的因子里混入了 T 日之后才可得的信息）。这两类风险恰恰是既有实证文献反复强调的可复现性危机的放大器 [1, 2]。

本项目从设计之初就以一条铁律划清边界: **回测里绝不调用 LLM**。语言模型负责提出因子想法、解释经济含义、阅读回测摘要; 确定性代码负责字段可得性检查、点时间合并、表达式求值、样本外检验与报告生成（图 1）。这样既保留了 LLM 的探索效率，又把所有数值判断交还给适合做计算、且可被逐行审计的模块。

本文贡献。

1. **可证伪的防前视契约。**我们把 no-look-ahead 约束形式化为点时间合并算子（§4，定义 1），并给出其防泄漏性质（命题 1）与对应的可证伪单元测试。
2. **可审计的受限表达式引擎。**因子被表示为携带经济解释与字段依赖的结构化对象，求值走白名单 AST，禁止任意代码执行（§5.1），使“可解释”不止于自然语言，而是可回溯到字段与算子。
3. **完全确定性的回测流水线与端到端闭环。**生成—校验—回测—反馈四步闭环中，唯有回测一步不碰 LLM（§5.4），并在聚源真实 raw 数据上完成初步样本外实证（§6）。

2 相关工作与定位

机器学习驱动的资产定价已被系统研究 [2]，而学术因子被公开后收益衰减的现象 [1] 提示我们: 因子研究的价值高度依赖诚实的样本外评估。近期一类工作尝试用 LLM 自动搜寻策略 [3] 或以 Transformer 做 Alpha 因子的符号回归 [4]，展示了“语言/序列模型提出因子”的潜力。

与这些工作相比，本项目的定位不在“更强的搜索器”，而在**可信底座**: 我们主张先把数据契约、防前视、样本外划分、确定性评价这套地基做扎实、做成可测试的工程约束，再谈生成能力。换言之，LLM 的角色被刻意收敛为“假设提出者与结果阅读者”，其产出必须穿过确定性校验与回测才能被接受。这一保守设计以牺牲部分自动化程度为代价，换取了结论的可审计性——我们认为这正是当前 LLM 因子研究最稀缺的性质。

3 系统架构

系统由六层组成: 配置与数据契约、数据管线、因子引擎、回测模块、LLM 封装、Agent 闭环。核心链路是一条“生成 → 校验 → 回测 → 反馈”的闭环, 其中**只有回测一步是纯代码、不含 LLM**(图 1)。数据自 `data/raw` 经清洗对齐得到 `panel.parquet`(`data/processed`), 再逐层沉淀因子与 LLM 中间结果至 `data/cache`。

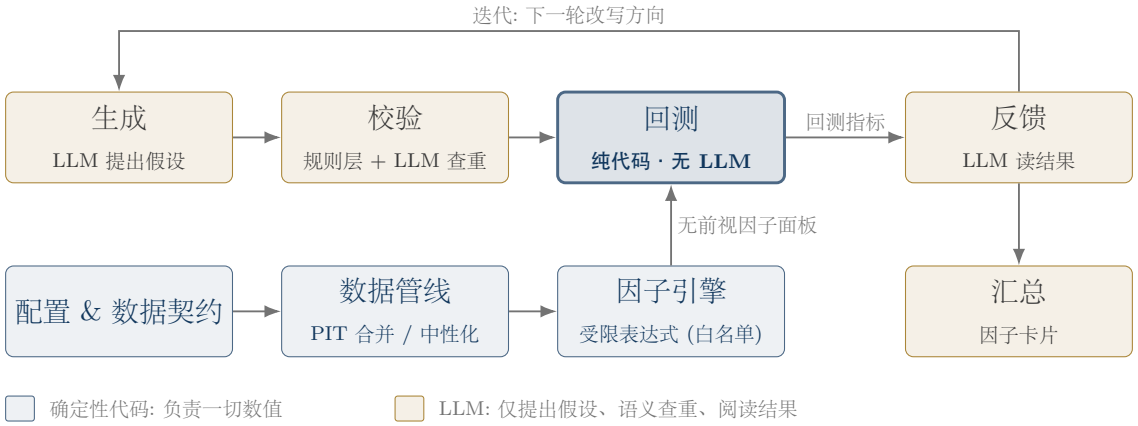

图 1: Alpha 因子生成智能体的主链路。上行是“生成 → 校验 → 回测 → 反馈”的四步闭环, 下行是支撑它的确定性底座。蓝色为确定性代码, 负责一切数值; 金色为 LLM, 仅参与提出假设、语义查重与阅读结果。**回测一步完全不触碰 LLM**, 两层之间只以结构化对象交互。

4 数据契约与点时间对齐

数据契约由三张表构成:`prices`(日频行情与复权因子 `adj_factor`)、`fundamentals`(财务字段与公告日 `ann_date`)、`universe`(历史成分股, 用于避免幸存者偏差)。防前视的关键在于: 财务数据必须按*公告日* `ann_date` 而非*报告期* `report_period` 对齐——一条报告期更早、但在 T 日之后才公告的记录, 在 T 日是不可得的。

定义 1 (点时间可见集与因子面板). 对交易日 T 与个股 i, 记其*点时间可见财务集*为

$$\mathcal{F}_i(T) \;=\; \big\{\, r \in \text{fundamentals}_i \;:\; \texttt{ann\_date}(r) \le T \,\big\},$$

点时间合并算子 `pit_merge` 取 $\mathcal{F}_i(T)$ 中*公告日最新*的一条与 T 日行情拼接, 得到因子面板单元 $x_{i,T} = \phi\big(\text{prices}_{i,T},\, \arg\max_{r\in\mathcal{F}_i(T)} \texttt{ann\_date}(r)\big)$。因子在 T 日的取值只能是 $\{x_{j,T}\}_{j\in\text{universe}(T)}$ 的函数。

命题 1 (防泄漏性). 若因子求值仅依赖 $\{x_{j,T}\}_j$, 则对任意 T, 其取值与所有 $\texttt{ann\_date} > T$ 的记录*条件独立*, 即不存在前视信息通道。

证明思路。 由定义 1, $x_{i,T}$ 的构造在 $\mathcal{F}_i(T)$ 上取上确界, 而 $\mathcal{F}_i(T)$ 按“$\texttt{ann\_date} \le T$”硬过滤, 故任何公告日晚于 T 的记录都不在拼接来源中; 又因子被限定为面板单元的函数, 泄漏通道被完全切断。 □

该性质在实现中由 `pit_merge` 强制执行（算法 1），并配以*可证伪*的单元测试（`test_pit_merge_excludes_future`）：构造一条公告日晚于 T 的“未来”记录，断言它**绝不**出现在 T 日面板中。测试失败即视为防线失守。

算法 1 `pit_merge`: 点时间合并（防前视核心）

输入： 行情表 `prices`、财务表 `fundamentals`（含 `ann_date`）
输出： 无前视的因子面板 `panel`

for all 交易日 T，个股 $i \in \text{universe}(T)$ **do**
  $\mathcal{F}_i(T) \leftarrow \{\, r \in \text{fundamentals}_i : \texttt{ann\_date}(r) \le T \,\}$
  if $\mathcal{F}_i(T) = \varnothing$ **then**
    该单元置空并跳过
  end if
  $r^\star \leftarrow \arg\max_{r \in \mathcal{F}_i(T)} \texttt{ann\_date}(r)$ ▷ 取最新已公告记录
  $\texttt{panel}[i, T] \leftarrow \text{join}\big(\text{prices}_{i,T},\, r^\star\big)$
end for
return `panel`

5 方法

5.1 受限因子表达式引擎

因子以结构化对象 `FactorExpr` 表示，携带名称、表达式、*经济解释*、字段依赖与元数据。表达式求值走**受限 AST 解析**：只允许基础算术与一组注册的白名单算子——`rank`/`cs_rank`（截面排序）、`ts_mean`/`ts_std`（时序统计）、`delay`/`delta`（滞后与差分）、`safe_div`（防零除）、`signed_log`（保号对数）。白名单之外的一切节点（函数调用、属性访问、下标等）在解析期即被拒绝，从而**禁止任意代码执行**。这一设计的意义在于：因子的“可解释性”不再停留于自然语言的经济故事，而是可以一路回溯到*具体字段*与*有限算子*，做到真正可审计。

5.2 基线因子库

实现内置五类基线因子，覆盖经典风格：*价值*（估值比率）、*质量*（盈利能力）、*动量*（短期趋势）、*低波动*（收益波动）与*流动性*（交易活跃度）。基线一方面构成最小可用研究集合，另一方面作为 Agent 语义查重与效果比较的参照系——任何 LLM 候选因子都要先与基线比对，避免“换皮重复”。

5.3 回测与评价指标

回测完全由本地代码实现。设 $f_{i,t}$ 为个股 i 在 t 日的因子值，$R_{i,t+1}$ 为其前向收益，N_t 为当日截面样本数。核心指标定义如下。

Rank IC 与 ICIR。 逐日计算因子与前向收益的秩相关，再对时间求稳定性:

$$\mathrm{RankIC}_t \;=\; \mathrm{corr}\big(\mathrm{rank}(f_{\cdot,t}),\, \mathrm{rank}(R_{\cdot,t+1})\big), \qquad \mathrm{ICIR} \;=\; \frac{\overline{\mathrm{RankIC}}}{\sigma(\mathrm{RankIC})}.$$

分组与多空收益。 按截面分位数将股票分为 Q 组，$G_k(t)$ 为第 k 组; 多空组合取最高减最低组:

$$r^{\mathrm{LS}}_{t+1} \;=\; \frac{1}{|G_Q(t)|}\sum_{i\in G_Q(t)} R_{i,t+1} \;-\; \frac{1}{|G_1(t)|}\sum_{i\in G_1(t)} R_{i,t+1}.$$

换手率与成本后收益。 以持仓权重 $w_{i,t}$ 的逐期变动近似交易摩擦，成本后收益按 `cost_bps` 扣减交易成本:

$$\mathrm{turnover}_t \;=\; \tfrac{1}{2}\sum_i \big|w_{i,t} - w_{i,t-1}\big|, \qquad r^{\mathrm{net}}_{t+1} \;=\; r^{\mathrm{LS}}_{t+1} - \mathrm{turnover}_t \cdot \texttt{cost\_bps}.$$

样本切分严格按日期滚动:`train_end` 之后统一作为样本外区间，**禁止随机打乱**，以杜绝跨期信息混入。

5.4 LLM 封装与 Agent 闭环

LLM 经统一客户端 `LLMClient.generate()` 封装，后端可切换（默认 OpenAI 兼容 API，测试用 mock），并对 prompt 做哈希缓存、限制 `max_tokens` 以节省开销。闭环由三个 Agent 串起（算法 2）：**生成** Agent 产出结构化候选因子; **校验** Agent 先过**确定性规则层**（字段存在、表达式可解析、是否遗漏 `delay` 等未来字段引用），再由 LLM 做**语义查重**; **反馈** Agent 依据回测摘要建议改写方向。无论哪一步，最终数值一律以确定性回测为准。

模型调参的交付重心不是训练一个直接预测收益的黑箱，而是让“候选因子提案器”更稳定、更合规。当前优先调 prompt 模板、JSON 输出约束、`temperature`、`max_tokens` 与规则校验; 当积累足够多人工审核样本且格式仍不稳定时，才考虑 LoRA 等参数高效微调。LoRA 在本项目中的目标是学习因子表达风格与结构化输出习惯，而不是替代回测计算 IC 或收益。

算法 2 生成—校验—回测—反馈闭环（数值均由确定性代码产生）

输入: 数据面板 `panel`、基线因子集 $\mathcal{B}$、迭代上限 K

输出: 通过校验并完成样本外回测的因子卡片集合 $\mathcal{C}$

$\mathcal{C} \leftarrow \varnothing$
for $k = 1$ **to** K **do**
  $F \leftarrow \textsc{GenerateAgent}()$ ▷ LLM: 产出结构化候选因子
  if not $\textsc{RuleCheck}(F)$ **then**
    continue ▷ 确定性规则层: 字段/可解析/防前视
  end if
  if $\textsc{SemanticDup}(F, \mathcal{B} \cup \mathcal{C})$ **then**
    continue ▷ LLM: 语义查重
  end if
  $m \leftarrow \textsc{Backtest}(F, \texttt{panel})$ ▷ **纯代码**:IC/ICIR/多空/换手
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{(F, m)\}$; $\mathcal{B} \leftarrow \mathcal{B} \cup \{F\}$
  $\textsc{FeedbackAgent}(m)$ ▷ LLM: 读摘要，建议下一轮改写
end for
return $\mathcal{C}$

6 实验: 聚源真实数据下的初步实证

实验目的与边界。 本节使用从聚源 JYDB 导出的真实 raw 数据重跑确定性流水线，检验数据契约、防前视合并、表达式求值、样本外回测与报告落盘能否在真实 A 股数据上闭环。由于当前只覆盖 2020–2021 年、股票池为基于行情观测构造的沪深 A 股 price-observed universe，且尚未纳入严格停牌、涨跌停、ST、行业中性化与完整复权处理，本文将结果定位为*初步实证与系统验收*，不把单次运行解释为稳健 alpha 发现。

真实数据接入状态。 项目已完成聚源 SQL Server 本地接入、脱敏诊断与 JYDB 元数据快照流程，真实账号、密码、主机与库名仅保存在本地忽略文件或环境变量中，不进入代码仓库与论文正文。元数据快照覆盖 2739 张表与 81757 个字段; 在此基础上，本文导出三张 raw 表: 行情表 `prices.pkl` 共 1,904,369 行、4,261 只股票，日期范围为 2020-01-02 至 2021-12-31; 财务表 `fundamentals.csv` 共 164,881 行、5,705 只股票，公告日范围为 2018-04-04 至 2021-12-31，报告期范围为 2018-03-31 至 2021-09-30，且 `(code, report_period, ann_date)` 重复键为 0; 股票池表 `universe.csv` 共 1,904,369 行、4,261 只股票，与行情日期范围一致。经 `pit_merge` 对齐后的面板共 1,904,369 行、18 列，其中 `mktcap` 非空 1,904,328 行。当前 `mktcap` 由 PIT 可见的 `shares_outstanding` 与当日收盘价相乘得到，行情复权因子暂取 `adj_factor=1.0`，相关限制在讨论节中单独列出。

实验设置。 关键配置集中于 `src/config.py`（表 1），所有模块从此处取参，不散落硬编码。

表 1: 真实数据初步实证的关键配置（集中于 `src/config.py`）

配置项	取值 / 说明
数据来源	聚源 JYDB raw 表，price-observed 沪深 A 股股票池
频率 `freq`	日频
样本内/外划分	2020 年用于前期计算与候选形成，2021 年为样本外，禁止随机打乱
交易成本 `cost_bps`	10 bps，以基点计入成本后收益
LLM 后端	mock，仅用于提出候选与阅读摘要; 回测中无 LLM 调用

验证清单。 端到端验证聚焦三点:

1. **防前视:** `pit_merge` 不把 `ann_date` $> T$ 的记录并入 T 日面板（对应单测断言）;
2. **接口一致:** 基线因子与 LLM 候选因子在统一接口下稳定求值;
3. **可落盘:** 回测、报告与因子汇总可端到端写入 `results/`。

表 2 先给出五个手工基线因子在 2021 年样本外区间的结果。低波动、流动性换手与价值类基线在本次数据口径下取得正 Rank IC 和正 ICIR; 质量 ROE 与 20 日动量方向为负。该结果说明基线模块能够在真实数据上生成有区分度的排序信号，但不构成跨年份稳健性结论。

表 2: 聚源真实 raw 数据下的基线因子样本外结果

因子	Rank IC	ICIR	换手率	成本后多空	样本数
`baseline_low_volatility_20d`	0.0795	0.6035	0.2604	0.0042	960478
`baseline_liquidity_turnover`	0.0662	0.5046	0.8799	0.0031	966784
`baseline_value_ep_bp`	0.0356	0.2967	0.1030	0.0029	958573
`baseline_quality_roe`	-0.0179	-0.1736	0.0721	-0.0033	958677
`baseline_momentum_20d`	-0.0305	-0.2142	0.6799	-0.0017	960478

表 3 给出两只 mock 生成候选因子经同一闭环后的样本外结果。两者的 Rank IC 与 ICIR 均为负，成本后多空收益也未通过正向证据门槛; 其中 `llm_mock_value_profit_blend` 与既有基线最大绝对相关性达到 0.8949，说明其新颖性不足。换言之，真实数据运行并没有把 LLM 候选因子“自动包装成成功案例”，而是通过确定性评价给出了可拒绝的负结果。

表 3: 聚源真实 raw 数据下的候选因子闭环结果

候选因子	Rank IC	ICIR	换手率	成本后多空	基线相关性
`llm_mock_value_profit_blend`	-0.0045	-0.0536	0.0892	-0.0021	0.8949
`llm_mock_cashflow_quality`	-0.0043	-0.0677	0.1187	-0.0006	0.3653

图 2 展示候选因子回测报告产物的一个例子，用于确认报告管线、图表导出与目录组织正确。

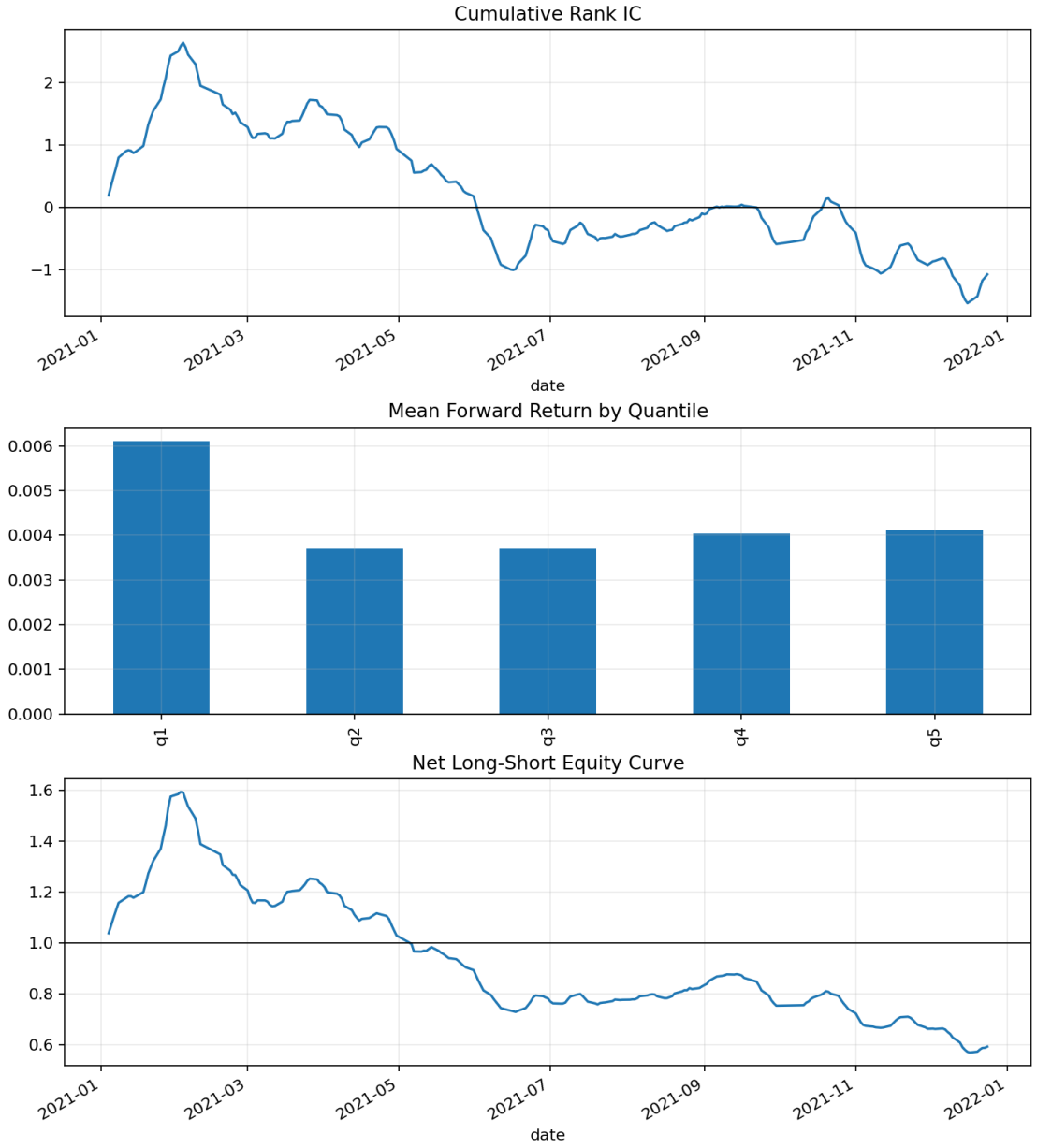

图 2: 单个候选因子（`llm_mock_value_profit_blend`）的真实数据回测报告示例。图表由 `report.py` 自动导出，数值来自确定性回测流水线。

7 可复现性声明

全流程可在聚源 raw 数据落地后由确定性脚本复现。核心防线由 pytest 守护，尤以防前视核心测试 `test_align.py` 为要:

```
pytest                          # run all unit tests
pytest tests/test_align.py -v # no-look-ahead core test
python scripts/paper_update_round.py --run-pipeline
```

原始表位于 `data/raw/prices.pkl`、`data/raw/fundamentals.csv` 与 `data/raw/universe.csv`; 结果摘要与论文表格由 `scripts/paper_update_round.py` 生成并落盘于 `results/paper_update/latest/`，回测报告与图表落盘于 `results/reports/`。所有数值均由上述确定性代码产生，LLM 不进入回测过程; 连接配置与凭据不写入论文正文。

8 讨论、局限与后续工作

价值定位。 当前版本的价值在于把**可信底座**推进到真实聚源数据上，而非宣称已发现稳定 alpha。实验表明: 程序链路正确、接口稳定、防前视与样本外约束有效; 同时，候选因子若缺乏样本外表现或与基线过度重合，会被确定性评价明确拒绝。

有效性威胁（threats to validity）。 (i) *外部有效性*: 当前只覆盖 2021 年一个样本外年度，且股票池为 price-observed 沪深 A 股，并非官方指数成分或严格可交易股票池;(ii) *构造有效性*: 暂未加入 ST、停牌、涨跌停、冲击成本、行业/市值中性化等更严格交易约束，`adj_factor` 目前取 1.0，`mktcap` 由 PIT 可见股本与收盘价近似;(iii) *设计取舍*:LLM 仅“提想法/读结果”，因子优劣不由其裁定——这一保守设计牺牲了自动化程度，换取对数值幻觉与回测泄漏的最大抑制。

后续工作。 下一阶段的三项主线:(1) 用官方成分股、ST/停牌/涨跌停过滤与完整复权字段替换当前简化口径;(2) 引入更严格的稳健性检验，包括多期滚动、分年度、行业与市值中性化后的稳定性;(3) 在缓存、`max_tokens` 与审计日志约束下接入真实 LLM 后端，扩大候选生成空间，同时保持回测完全确定性。

9 结论

本文给出可解释 Alpha 因子生成智能体的初版实现与聚源真实数据下的初步实证。系统在*数据层*以点时间对齐防止 look-ahead（并给出可证伪测试），在*方法层*以受限表达式引

擎与纯代码回测保证可审计，在**交互层**以 LLM 封装与三 Agent 闭环提升探索效率——同时以铁律确保回测绝不触碰 LLM。最新实验已完成 JYDB 三张 raw 表导出、真实数据面板构建与 2021 年样本外回测: 部分基线因子取得正向样本外表现，而 mock 候选因子未通过收益与新颖性检验。这一结果并不宣称发现稳定 alpha，却证明系统已经能够在真实数据上诚实地产生、检验并拒绝候选因子。我们相信，**先地基、后生成**的路线，是让“LLM 生成因子”这件事从“看起来聪明”走向“可被相信”的必要一步。

参考文献

[1] McLean, R. D., & Pontiff, J. (2016). Does academic research destroy stock return predictability? *Journal of Finance*, 71(1), 5–32.

[2] Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *Review of Financial Studies*, 33(5), 2223–2273.

[3] Kou, Z., et al. (2025). Automate Strategy Finding with LLM in Quant Investment. *Findings of EMNLP*.

[4] Huang, Y., et al. (2026). AlphaFormer: End-to-End Symbolic Regression of Alpha Factors with Transformers. *Proceedings of Machine Learning Research (PMLR)*.