《多模态大模型》

# Lecture 14
# 高阶多模态大模型

王广润
中山大学
人机物智能融合实验室（HCP Lab）
wanggrun@gmail.com

# 三个层次

- operation-level
- subnet-level
- agent-level

人类的大脑的聪明与否，与大脑的并行性非常相关

As you read, different parts of your brain handle different tasks simultaneously. Some areas decode the words, others recall related knowledge or emotional context, and yet others help form an overall understanding. All these processes feed into each other—your memory, expectations, and reasoning all run in parallel, making sense of what you’ve read and integrating it into your broader understanding. It’s a constant, dynamic interplay.

While some parts process the words and others tap into memory, other areas work on predicting what’s coming next, interpreting the emotional tone, or assessing whether the content aligns with your goals. Simultaneously, some regions handle attention—deciding what to focus on more—and others keep track of your overall comprehension. In essence, it’s a symphony: prediction, emotion, focus, and understanding all run in parallel.

本节内容

CONTENTS

一、金融量化

二、物理AI

# 金融量化

- **AlphaAgent: LLM-Driven Alpha Mining with Regularized Exploration to Counteract Alpha Decay**
- QuantaAlpha: An Evolutionary Framework for LLM-Driven Alpha Mining
- TradingAgents: Multi-Agents LLM Financial Trading Framework
- TradeFM: A Generative Foundation Model for Trade-flow and Market Microstructure (J.P. Morgan AI Research)
- FactorEngine: A Program-level Knowledge-Infused Factor Mining Framework for Quantitative Investment

# 核心变量

在因子挖掘任务中，给定一系列基础金融数据，目标是寻找能够预测未来收益的数学表达式，即 Alpha 因子。论文首先定义了以下核心变量：

- **股票集合**：$S = \{s_1, \ldots, s_N\}$
- **时间窗口**：$\mathcal{T} = \{t_1, \ldots, t_T\}$
- **输入特征矩阵**：$X \in \mathbb{R}^{N \times T \times D}$，其中 $D$ 表示原始特征的维度，例如开盘价、收盘价、成交量等。
- **因子映射**：因子 $f$ 将某一时刻的特征切片 $X_t$ 映射为预测信号 $r_{t+1}$，即下一期的资产收益率。形式化表示为：$f(X_t) \to r_{t+1}$。

# 因子挖掘被定义

$$f^* = \arg\max_{f \in \mathcal{F}} \mathcal{L}(f(X), y) - \lambda \mathcal{R}(f)$$

- $\mathcal{F}$ 代表所有可能的因子表达式空间。
- $y$ 表示真实的未来收益，即 ground-truth future returns。
- $\mathcal{L}$ 用于衡量因子的预测有效性，例如信息比率 IR 等性能指标。
- $\mathcal{R}$ 是鼓励因子保持简单或新颖的正则化项，$\lambda$ 是权衡性能与复杂度的平衡参数。

# 引入市场假设的优化目标

$$f^* = \arg\max_{f \in \mathcal{F}} \mathcal{L}(f(X), y) - \lambda \mathcal{R}_g(f, h)$$

此处的 $\mathcal{R}_g(f, h)$ 是一个全新的正则化项。它不仅约束了因子的表达式复杂度，还强制因子与市场假设 $h$ 保持一致，并确保其相对于现有因子的原创性。由于该目标是非凸的，AlphaAgent 采用了在 $\mathcal{L}$ 和 $\mathcal{R}_g$ 之间交替优化的策略，直到收敛到局部最优解。

# 正则化约束项的数学拆解

$$\mathcal{R}_g(f,h) = \alpha_1 \cdot SL(f) + \alpha_2 \cdot PC(f) + \alpha_3 \cdot ER(f,h)$$

- $SL(f)$：符号长度（Symbolic length），衡量公式本身的冗长程度。
- $PC(f)$：自由参数计数（Free parameters count，例如滚动窗口的天数），防止参数过拟合。
- $ER(f,h)$：综合评估因子相对现有因子的**新颖性（Novelty）**以及与假设的**对齐度（Alignment）**。

为了计算 $ER(f,h)$，论文将因子代码解析为**抽象语法树（AST）** $T(f)$，并定义了以下严密的量化指标。

# 基于子树同构的原创性度量

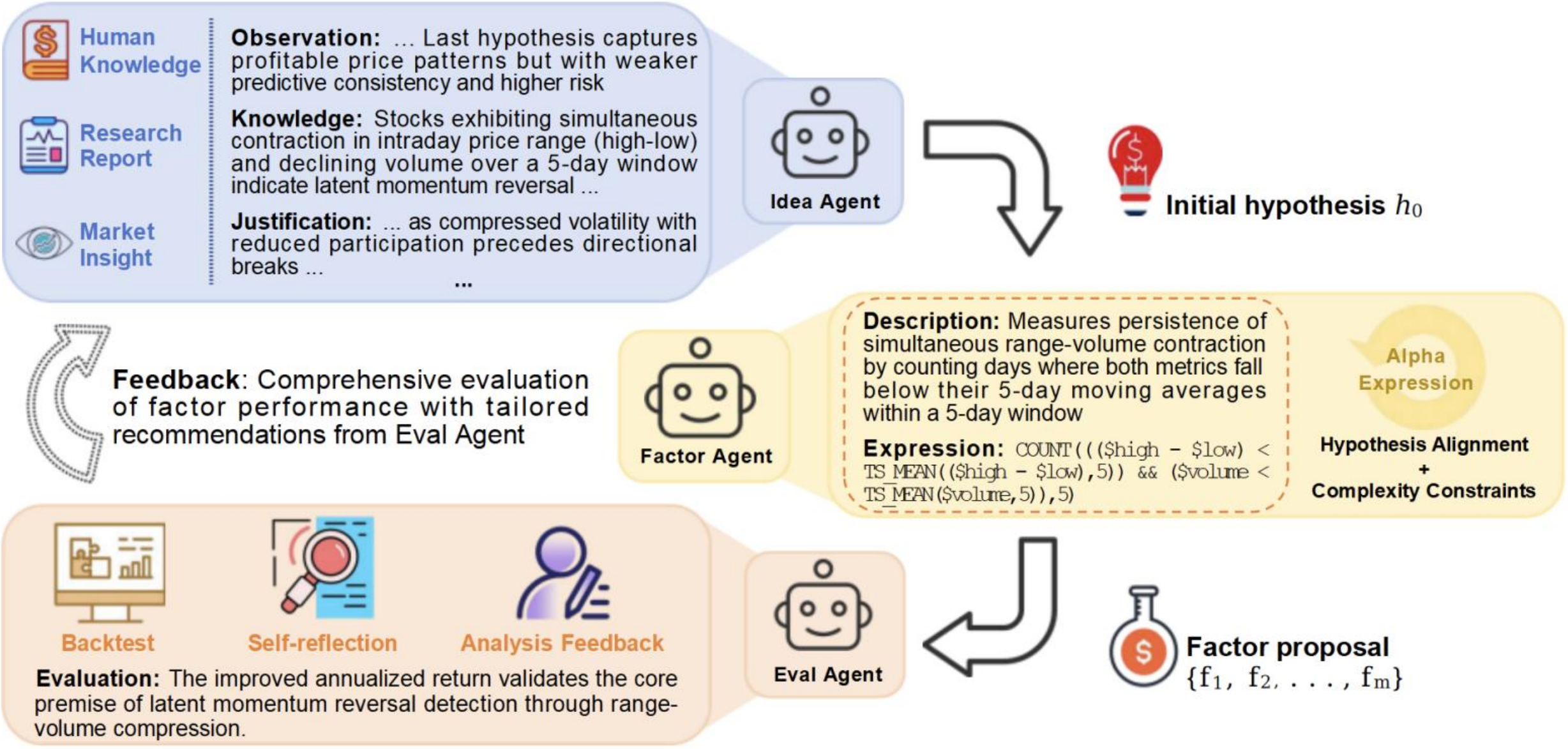
为了避免因子拥挤（Factor Crowding），系统必须计算新因子 $f_i$ 与现有因子 $f_j$ 的相似度。相似度 $s$ 是通过递归比较它们的 AST 结构 $T(f_i)$ 和 $T(f_j)$，找出其**最大公共子树**来计算的：

$$s(f_i, f_j) = \max_{t_i \subseteq T(f_i), t_j \subseteq T(f_j)} \{|t_i| : t_i \cong t_j\}$$

- $t_i$ 和 $t_j$ 分别是 $T(f_i)$ 和 $T(f_j)$ 的子树，$|t_i|$ 表示子树的节点数量。
- $t_i \cong t_j$ 表示两棵子树在结构上是同构的。

随后，将其与现有的基准因子库（Alpha Zoo）$\mathcal{Z} = \{\phi_1, \phi_2, \ldots, \phi_N\}$，如 Alpha101，进行比对，得到该因子的最大相似度得分 $S(f)$，作为其缺乏原创性的惩罚依据：

# 自主多智能体工作流


Human Knowledge
Research Report
Market Insight
Observation: ... Last hypothesis captures profitable price patterns but with weaker predictive consistency and higher risk
Knowledge: Stocks exhibiting simultaneous contraction in intraday price range (high-low) and declining volume over a 5-day window indicate latent momentum reversal ...
Justification: ... as compressed volatility with reduced participation precedes directional breaks ...
...
Idea Agent
Initial hypothesis $h_0$
Feedback: Comprehensive evaluation of factor performance with tailored recommendations from Eval Agent
Factor Agent
Description: Measures persistence of simultaneous range-volume contraction by counting days where both metrics fall below their 5-day moving averages within a 5-day window
Expression: COUNT(($high - $low) < TS_MEAN(($high - $low),5)) && ($volume < TS_MEAN($volume,5)),5)
Alpha Expression
Hypothesis Alignment + Complexity Constraints
Backtest
Self-reflection
Analysis Feedback
Evaluation: The improved annualized return validates the core premise of latent momentum reversal detection through range-volume compression.
Eval Agent
Factor proposal
{f1, f2, . . . , fm}

# 总体框架

该框架以市场假设生成为起点，以量化因子构建为核心，以历史数据回测评估为反馈信号，形成一个能够不断自我修正、自我迭代的自主多智能体系统。 三类智能体分别承担**创意生成**、**因子实现**和**性能评估**三项关键职责，从而将金融知识、数学表达式和市场验证过程统一到一个闭环工作流中。

# 三个专业智能体

Idea Agent

## 创意智能体

作为框架的起点，创意智能体负责综合人类知识、研究报告和市场洞察来提出市场假设。 它通过思维链（Chain-of-Thought）推理构建包含四个部分的系统性假设： **经验观察结果**、**金融知识综合**、**经济学机制论证**以及**实施规范**。

# 三个专业智能体

Factor Agent

## 因子智能体

因子智能体充当理论假设与量化表达式之间的桥梁。 它为每个假设生成多个候选因子实现方案，包括自然语言描述和数学表达式， 并利用其维护的成功/失败历史知识库，结合三种正则化约束，不断过滤和优化因子。

# 三个专业智能体

Eval Agent

## 评估智能体

评估智能体负责将因子放入真实框架中，利用历史数据进行多维度回测评估。它主要检验三个方面：**预测能力指标**（如 IC、RankIC）、**收益表现指标**（如年化收益 AR）和**风险控制指标**（如最大回撤 MDD）。

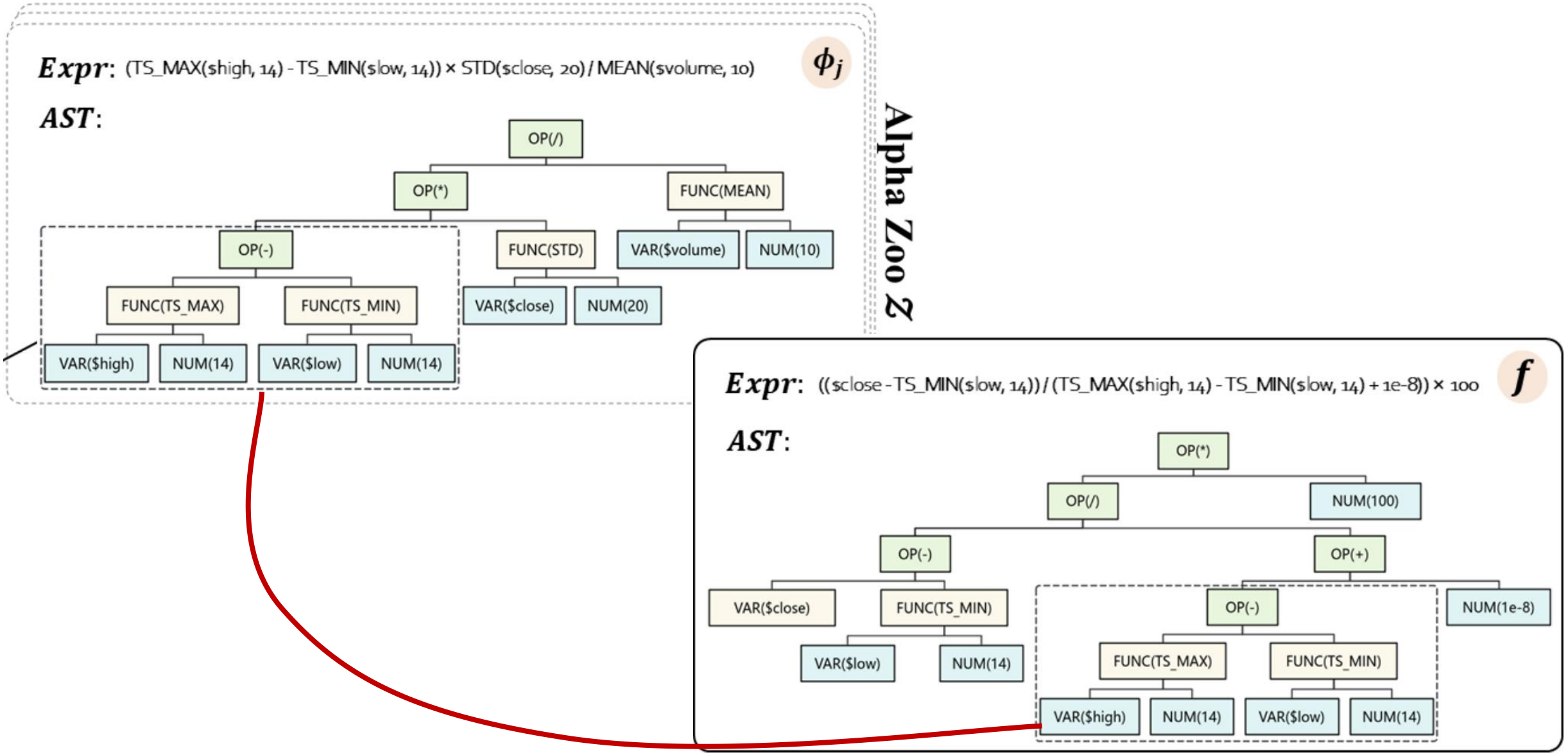
# 闭环

## 闭环反馈机制

评估智能体会分析因子的性能表现，识别其成功或失败的规律，并将这些经验知识反馈给创意智能体。 这些反馈进一步指导下一轮市场假设的细化和因子表达式的优化，使系统能够根据市场变化持续调整搜索方向。

> 通过这种闭环机制，AlphaAgent 不仅能够生成单个因子，还能够在持续迭代中积累经验， 自适应不断变化的市场条件，并挖掘出更具抗衰退能力的新因子。

# AST


Expr: (TS_MAX($high, 14) - TS_MIN($low, 14)) × STD($close, 20) / MEAN($volume, 10)
$\phi_j$
AST:
OP(/)
OP(*)
FUNC(MEAN)
OP(-)
FUNC(STD)
VAR($volume)
NUM(10)
FUNC(TS_MAX)
FUNC(TS_MIN)
VAR($close)
NUM(20)
VAR($high)
NUM(14)
VAR($low)
NUM(14)
Alpha Zoo Z
Expr: (($close - TS_MIN($low, 14)) / (TS_MAX($high, 14) - TS_MIN($low, 14) + 1e-8)) × 100
f
AST:
OP(*)
OP(/)
NUM(100)
OP(-)
OP(+)
VAR($close)
FUNC(TS_MIN)
OP(-)
NUM(1e-8)
VAR($low)
NUM(14)
FUNC(TS_MAX)
FUNC(TS_MIN)
VAR($high)
NUM(14)
VAR($low)
NUM(14)

# 金融量化

- AlphaAgent: LLM-Driven Alpha Mining with Regularized Exploration to Counteract Alpha Decay
- **QuantaAlpha: An Evolutionary Framework for LLM-Driven Alpha Mining**
- TradingAgents: Multi-Agents LLM Financial Trading Framework
- TradeFM: A Generative Foundation Model for Trade-flow and Market Microstructure (J.P. Morgan AI Research)
- FactorEngine: A Program-level Knowledge-Infused Factor Mining Framework for Quantitative Investment

# 核心视角：挖掘轨迹的数学抽象

不同于纯数据驱动的方法，QuantaAlpha 将每一次因子挖掘过程定义为一个有序的轨迹序列 $\tau = (s_0, a_0, s_1, a_1, \ldots, s_n)$，其中 $s_0$ 是初始环境（如市场背景或种子因子），$a_i$ 是智能体在第 $i$ 步采取的动作，而 $s_n$ 是包含回测结果的终端状态。

框架的核心目标是学习一个能够最大化轨迹终端奖励（Terminal Reward）的策略：

$$f^* = \arg\max_{f \in \mathcal{F}} \mathcal{L}(f(X), y) - \lambda \mathcal{R}(f)$$

# 多样化规划初始化

- **为了避免模型陷入局部最优解，框架在初始阶段不会盲目搜索，而是提供一个多样化的规划初始化。**
  - ✓系统使用公共因子库（如 Alpha158 的低相关性子集）作为初始种子池。
  - ✓初始化智能体（Initialization Agent）会被提示在语义和结构层面尽可能生成互补的市场假设，例如区分量价信号、长短期窗口、动量或均值回归机制等。
- **这极大地拓宽了有效搜索的边界，使得后续的进化过程可以并行探索多个有潜力的区域。**

# 可控的因子构建与多重约束门

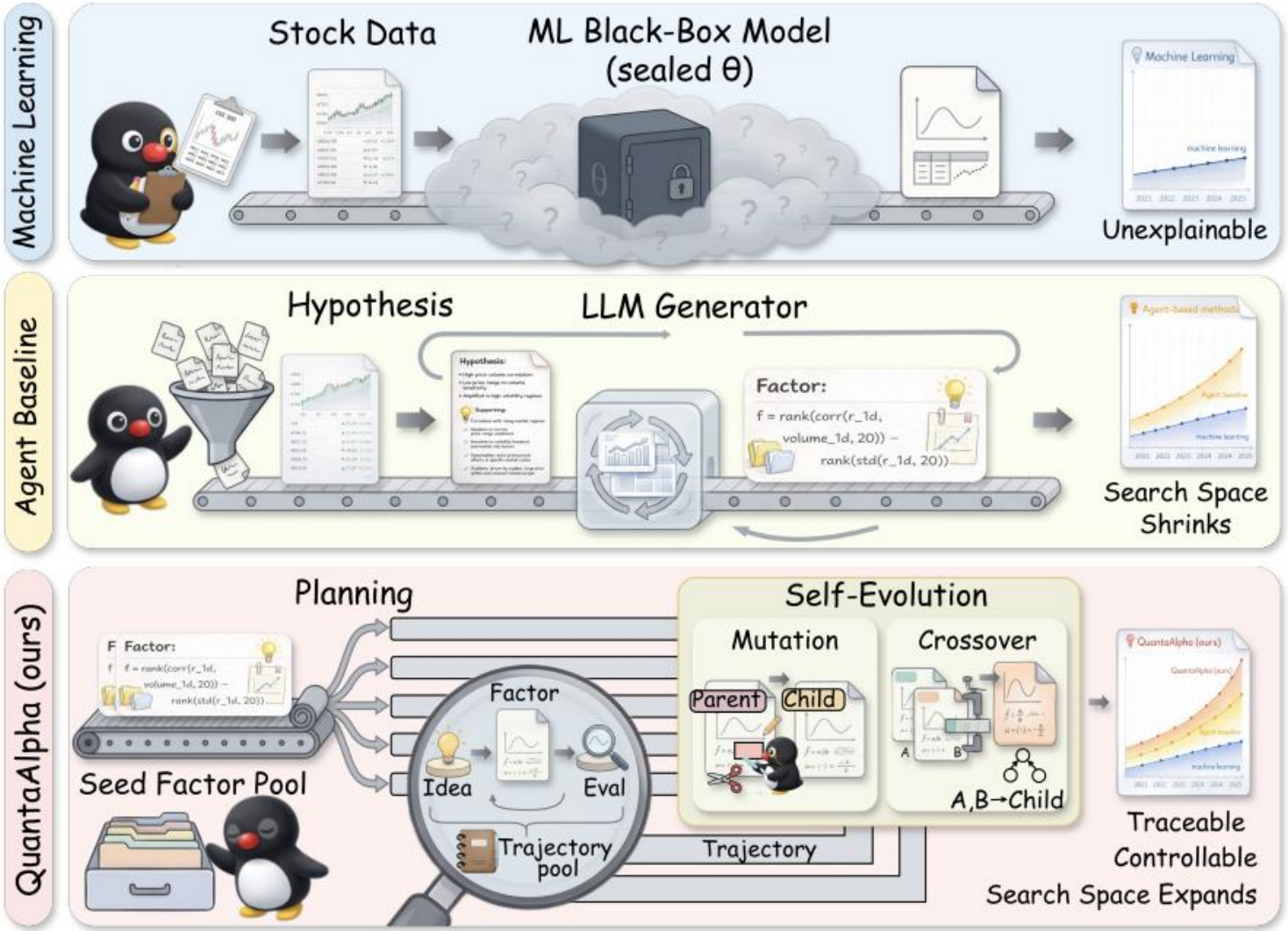
- **抽象语法树（AST）解析：** 智能体首先将自然语言假设转化为结构化的语义描述，然后调用标准化的操作符库（如 TS_MIN、RANK 等）将其组装成符号表达式，并最终解析为抽象语法树。这使得因子的计算依赖和数据流变得完全透明。
- **一致性验证：** 利用 LLM 作为验证器，强制检查“市场假设”、“因子表达式”以及“可执行代码”三者之间的语义一致性，防止生成过程发生语义漂移。
- **复杂度控制：** 通过计算符号长度 自由参数数量 以及使用到的特征数量 对过于臃肿的因子进行惩罚与重写
- **冗余度过滤：** 为了缓解“因子拥挤（Factor Crowding）”，系统会通过 AST 子树匹配算法计算新因子与现有因子库的最大公共子树，评估其结构相似度

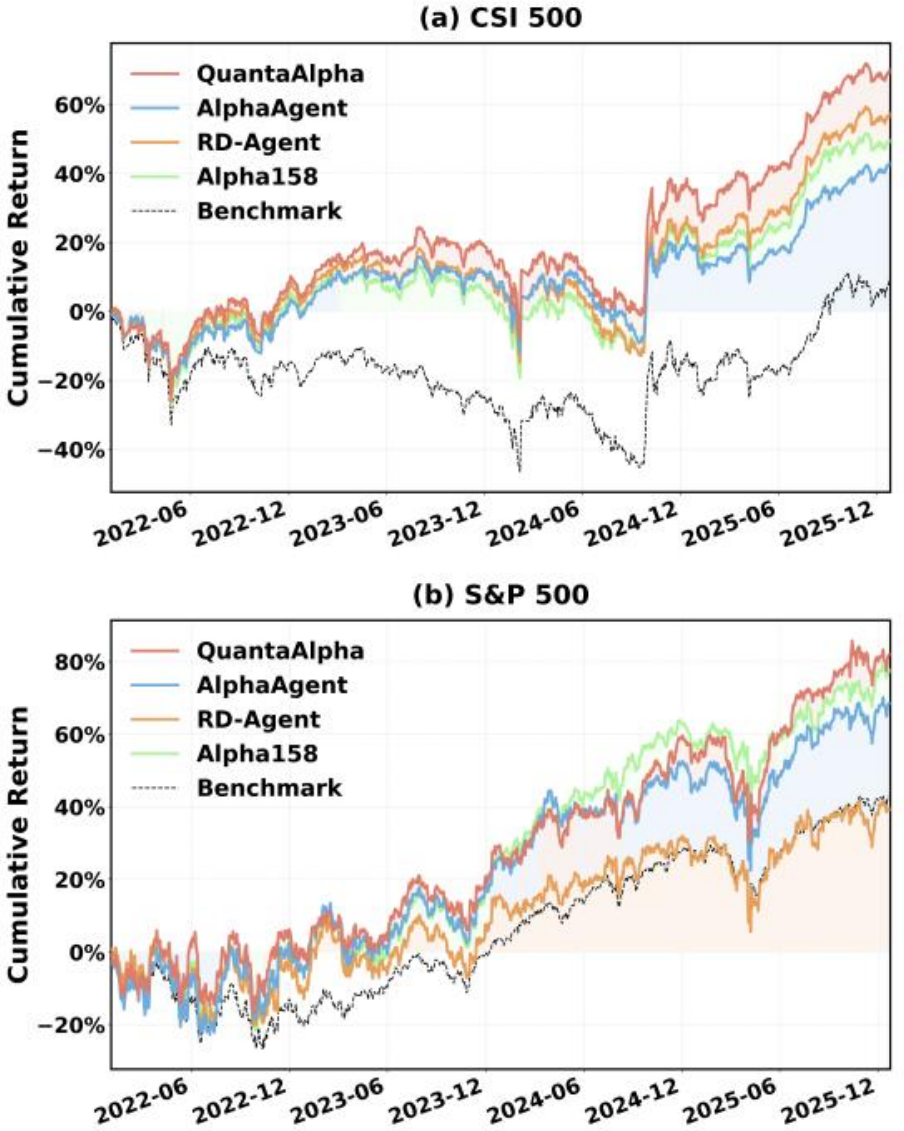
# 轨迹级别的自我进化

**变异算子（Mutation - 靶向修正）：** 给定一条效果不佳的历史轨迹，智能体首先进行“自我反思”，定位导致终端奖励低下的**特定故障节点**，例如某个逻辑步骤出现偏差。系统只对该局部段落进行重写修改，例如改变时间尺度或添加市场状态条件，而保持轨迹的其余部分完好无损，从而在保持整体连贯性的同时定向探索新空间。

**交叉算子（Crossover - 经验重组）：** 系统选取表现优异的几条父代轨迹，将它们互补的有效片段，如成熟的假设模板、构建模式或修复策略，进行重组。这模仿了人类量化研究员将不同策略的优势进行结合的思维模式，使得新生成的后代轨迹能够显式地继承已经被市场验证过的成功经验，大幅提升了因子的可靠性和可追溯性。

# QuantaAlpha



# QuantaAlpha

## Hypothesis Operations

### A. Diversified Planning Initialization

Seed Library
$close/Min($low, 5)-1
$Std($volume, 5)/Mean($volume, 5)
($high-$low)/$open
...

LLM Hypothesis Generator

Hypothesis h1
Hypothesis h2
Hypothesis h3
...

### B. Factor Realization

Factor Instantiation×3

des → expr → AST

Revision

Feedback

Constraint Gate

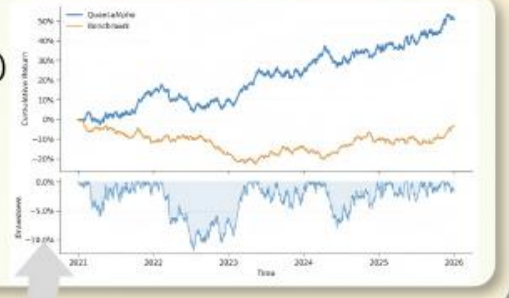
Consistency
Complexity (SL≦250|ER≦6)
Redundancy

Backtest

IC
ARR

## D. Final Factor Pool

**VASRF_Stealth_Accumulation_20D**:
★[CORR(c,v,20) * MEAN(TR,14) / ( (CORR(rng,v,10) +ε) * (MEAN(c,14)+ε) )]
**Stealth_Efficiency_Rank_15D**:
★[RANK(CORR(c,v,15)) - RANK(CORR(rng,v,15))]
**Volatility_Regime_Synergy_10D**:
★[CORR(c,v,10) * ZSCORE(rng/(c+ε), 20)]



## C. Self-Evolution Strategy

[1] Parent = trajectory (t)
[2] Child = new trajectory (t+1)

Parent[1]: Combines **volume-pressure × price-volume coupling** and **regime-damps** it with a volatility-to-range filter, ...

Mutation

{targeted revision}

Child[2]: A signed **return-volume momentum** proxy (not true ownership coordination), ...

Parent: Validates a **microstructure efficiency regime** using range-volume alignment and range-per-volume cost, ...

Crossover

Parent: Targets **overbought mean-reversion** by combining attention, RSI, institutional/fundamental deterioration, ...

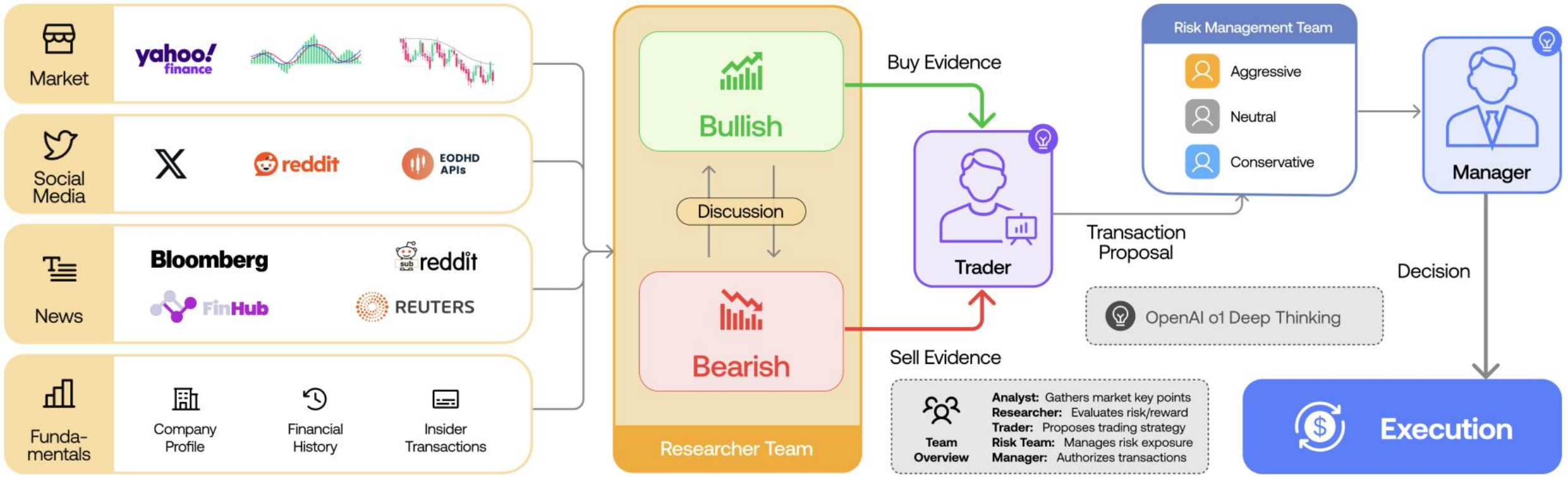
{segment recombination}

Child: A minimal **institutional-momentum proxy** using price-volume-change correlation scaled, ...

# 金融量化

- AlphaAgent: LLM-Driven Alpha Mining with Regularized Exploration to Counteract Alpha Decay
- QuantaAlpha: An Evolutionary Framework for LLM-Driven Alpha Mining
- **TradingAgents: Multi-Agents LLM Financial Trading Framework**
- TradeFM: A Generative Foundation Model for Trade-flow and Market Microstructure (J.P. Morgan AI Research)
- FactorEngine: A Program-level Knowledge-Infused Factor Mining Framework for Quantitative Investment

# TradingAgents



# 累积收益率

$$CR = \left( \frac{V_{end} - V_{start}}{V_{start}} \right) \times 100\%$$

其中：

- $V_{start}$ 表示期初的投资组合总价值。
- $V_{end}$ 表示期末（模拟结束时）的投资组合总价值。

# 年化收益率

$$AR = \left( \left( \frac{V_{end}}{V_{start}} \right)^{\frac{1}{N}} - 1 \right) \times 100\%$$

其中：

- $N$ 表示回测周期所包含的年数（例如，如果回测了 3 个月，则 $N = 3/12 = 0.25$）。

# 夏普比率

$$SR = \frac{\overline{R} - R_f}{\sigma}$$

- $\overline{R}$ 表示投资组合的平均预期收益率（Average portfolio return）。
- $R_f$ 表示无风险利率（Risk-free rate，通常使用短期国债收益率作为基准）。
- $\sigma$ 表示投资组合收益率的标准差（Standard deviation of the portfolio returns），代表了系统交易收益的波动率或风险。

# 最大回撤

$$MDD = \max_{t \in [0,T]} \left( \frac{Peak - Trough}{Peak} \right) \times 100\%$$

- $Peak$ 是资产组合净值在下跌前的最高点。
- $Trough$ 是该次下跌过程中的最低点。

最大回撤越小，说明模型中的风控智能体（如“保守型风控智能体”）对亏损的干预和交易计划修改机制越有效。

# 金融量化

- AlphaAgent: LLM-Driven Alpha Mining with Regularized Exploration to Counteract Alpha Decay
- QuantaAlpha: An Evolutionary Framework for LLM-Driven Alpha Mining
- TradingAgents: Multi-Agents LLM Financial Trading Framework
- **TradeFM: A Generative Foundation Model for Trade-flow and Market Microstructure (J.P. Morgan AI Research)**
- FactorEngine: A Program-level Knowledge-Infused Factor Mining Framework for Quantitative Investment

# TradeFM

- 这篇论文提出了一种名为 **TradeFM** 的生成式基础模型（**Generative Foundation Model**），专门用于预测交易流和市场微观结构。为了处理极其复杂的金融数据并实现强大的泛化能力

# 核心方法

- **尺度不变特征：** 金融市场中的不同资产（如不同股票）在价格、交易量和最小变动价位（Tick size）上差异巨大。为了实现跨资产的泛化（Cross-asset generalization），研究人员开发了尺度不变特征。这种方法使得模型能够学习到市场微观结构的通用规律，而不被特定资产的绝对数值所局限。
- **通用分词机制：** 传统的金融模型往往需要针对特定资产进行单独的校准。TradeFM 引入了一种通用的分词方案，将包含异构、多模态信息的订单流（Order flow）事件流映射为一个统一的离散序列（Unified discrete sequence）。这一机制彻底消除了针对特定资产进行模型校准的需求，使得数据可以直接输入到大语言模型架构中。

# 核心方法

- **生成式 Transformer 架构：** 模型的核心是一个拥有 5.24 亿参数的生成式 Transformer。与自然语言处理中的大模型类似，TradeFM 直接从超过 9000 只股票的数十亿次真实交易事件中进行学习，通过序列生成的方式预测未来的市场事件。

# 核心方法

## ❑结合确定性市场模拟器

- ✓TradeFM 并不是孤立运行的，而是与一个确定性的市场模拟器相结合。在这个系统中：
- ✓TradeFM 的角色：作为一个逼真的“背景”市场，生成合理且具有反应性的对手方订单流（Counterparty order flow）。
- ✓模拟器的作用：结合模型生成的订单，精确计算价格并复现金融回报的关键典型特征，如厚尾分布（Heavy tails）、波动率聚集（Volatility clustering）以及无回报自相关性等。
- ✓这种结合甚至支持进行反事实的压力测试（Counterfactual stress testing），例如向系统中注入频率高达正常情况 10 倍的异常订单流，来观察模型生成逼真的价格路径反馈，以此进行影响分析。

# 核心方法

- 参与者级别的条件生成与多智能体支持
  - 强化学习（RL）微调：RL 智能体可以在这个模拟市场中通过交互来学习最优的大额订单执行策略，从而最小化价格冲击和买卖价差等交易成本。
  - 多智能体系统（MAS）：模拟器可以接入多个异构的学习型智能体，研究它们交互时产生的涌现性集体行为或潜在的系统不稳定性。
  - 模型提供的参与者级别条件生成（Participant-level conditioning）机制，为初始化和微调上述多样化的智能体策略提供了天然且强大的支持。

# 生成式的自回归序列建模问题

- 这篇论文（TradeFM）将金融市场微观结构的建模，严格定义为一个生成式的自回归序列建模问题（Autoregressive Sequence Modeling Problem）。

- 为了让大语言模型（Transformer）能够理解并预测高度非平稳且充满噪音的金融交易流，论文采用了一系列精密的数学定义和特征变换方法。以下是详细的数学原理解析

# 核心问题的数学抽象

- 1. 交易事件序列与目标分布

论文将市场动态表示为一个随时间展开的离散交易事件序列 $E = (e_1, e_2, \ldots, e_T)$ 。模型的核心优化目标是学习条件概率分布 $P(e_t | e_{

- 2. 交易事件的多元元组表示

在真实市场中，每个时间步 $t$ 的单个交易事件 $e_t$ 不是一个单一数值，而被严谨地定义为一个包含五个核心特征的元组：$e_t = (\Delta t_t, \delta p_t, v_t, a_t, s_t)$ 。

# 核心问题的数学抽象

$\Delta t_t$：距离上一个事件的**到达时间间隔**（单位为秒）。

$\delta p_t$：交易的**价格深度**（通常以基点表示）。

$v_t$：交易的**数量/份额**。

$a_t$：**动作类型**（如添加订单或取消订单）。

$s_t$：发起订单的**买卖方向**（买入/Bid 或 卖出/Ask）。

# 核心数学处理方法

为了让基础模型能够实现跨越数千只不同股票（具有极大绝对价格和流动性差异）的泛化，论文采用了以下三种关键的数学变换方法：

- 1. 鲁棒的中间价估计 (EW-VWAP)

在模型所处的"部分可观察"设定下，模型无法直接看到真实的买卖中间价 $p_t^{mid}$，只能看到带有噪音的实际执行价格 $p_t^{exec}$ 。传统的成交量加权平均价 (VWAP) 公式为 $\hat{p}_t^{VWAP} = \frac{\sum_{i=0}^{W} v_{t-i} p_{t-i}^{exec}}{\sum_{i=0}^{W} v_{t-i}}$ ，但这在不同流动性的资产间缺乏可比性 。

# 核心数学处理方法

为了使其对近期大额交易更敏感，论文引入了**指数加权成交量加权平均价 (EW-VWAP)**，通过维护分子 $N_t$ 和分母 $D_t$ 的指数移动平均来实现：

- 加权价格分子：$N_t = \alpha \cdot (p_t^{exec} \cdot v_t) + (1 - \alpha) \cdot N_{t-1}$
- 成交量分母：$D_t = \alpha \cdot v_t + (1 - \alpha) \cdot D_{t-1}$
- 最终估计值：$\hat{p}_t^{EW-VWAP} = \frac{N_t}{D_t}$ 其中，平滑因子 $\alpha$ 由基于时间的半衰期决定，确保估计值在时间上保持一致的敏感度 。

# 核心数学处理方法

- 2. 尺度不变特征构建 (Scale-Invariant Features)

为了消除不同股票（如高价科技股与廉价仙股）之间的绝对数值差异，特征被严格映射为无量纲的相对值：

- 对数交易量：**为了压缩服从重尾幂律分布的订单数量范围，采用对数变换** $v_t = \log(1 + V_t)$，其中 $V_t$ 是原始交易份额 。
- 归一化价格深度：**计算订单价格相对于估计中间价的深度比率** $d_t = \frac{p_t^{order} - \hat{p}_t^{mid}}{\hat{p}_t^{mid}}$ ，从而替代绝对的最小变动价位（Tick）深度 。
- 相对价格水平：**为了捕捉日内价格变动，定义** $\Delta p_t = \frac{\hat{p}_t^{mid} - p_0}{p_0}$，其中 $p_0$ 是当天的开盘价 。

# 核心数学处理方法

- 3. 基于混合进制系统的分词映射 (Universal Tokenization via Mixed Base System)

Transformer 的解码器在每个时间步通常只能预测一个一维的 Token 。为了将上述包含连续和离散变量的多元特征元组 $(i_{\Delta t}, i_{\delta p}, i_v, i_a, i_s)$ 压缩成单一整数，论文首先通过分箱（Binning）将连续变量离散化为索引，然后巧妙地采用了一种**混合进制数系统 (Mixed Base Number System)** 。

假设每个特征的离散化分箱（Bin）数量分别为：动作 $n_a = 2$、方向 $n_s = 2$、深度 $n_{\delta p} = 16$、交易量 $n_v = 16$、到达时间 $n_{\Delta t} = 16$ 。

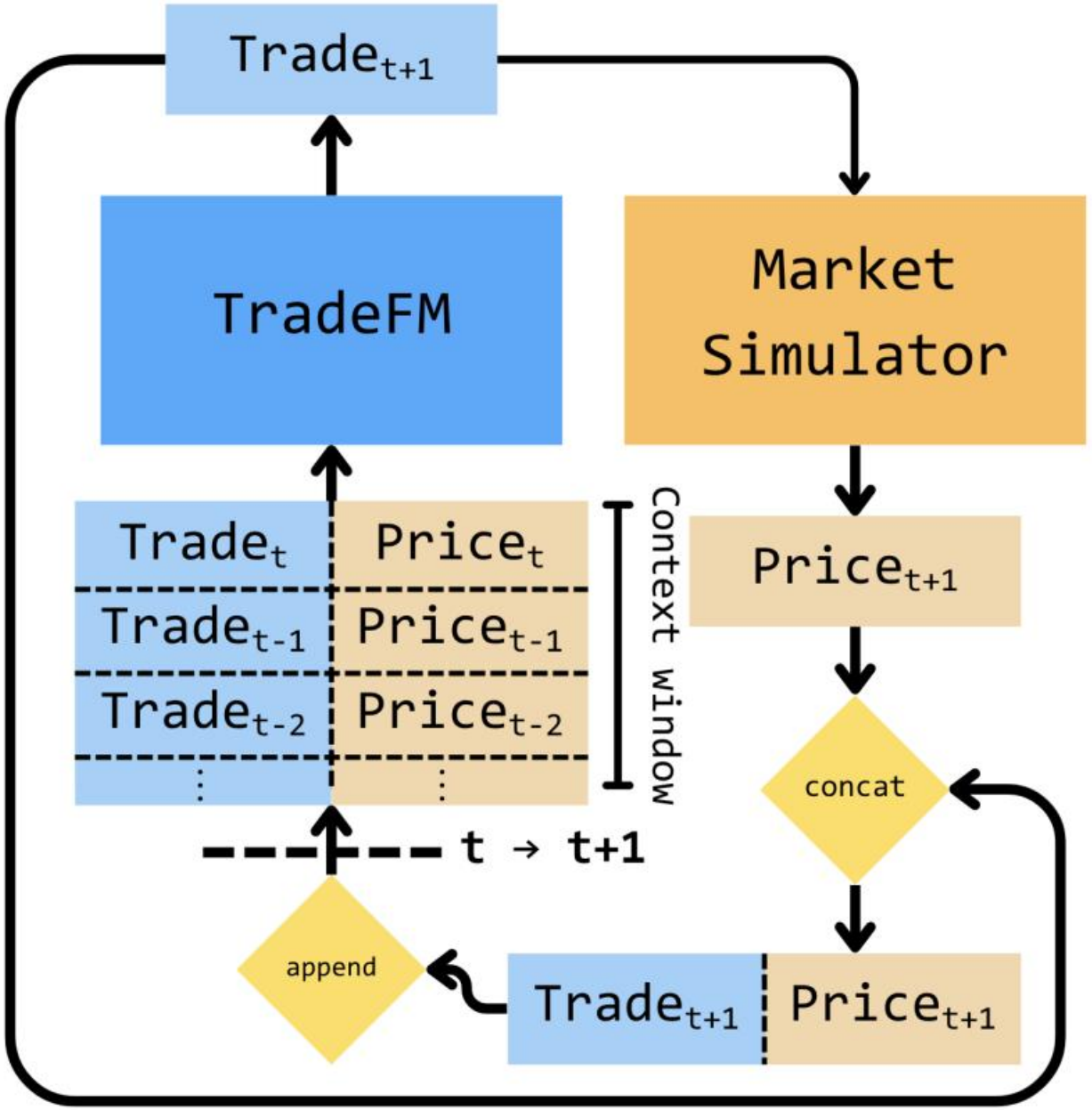
# 核心数学处理方法

系统将每个特征的箱索引视为混合进制系统中的一个“位（Digit）”，并将后续特征的可能取值数量作为该位对应的“基数（Base）”。复合贸易 Token $i_{trade}$ 的数学组合公式如下：

$$i_{trade} = (i_a \times n_s \times n_{\delta p} \times n_v \times n_{\Delta t}) + (i_s \times n_{\delta p} \times n_v \times n_{\Delta t}) + (i_{\delta p} \times n_v$$

通过这一数学映射机制，多模态、多维度的复杂交易信息被完美压缩，使得模型最终只需在一个固定大小为 16,384 的一维词汇表中进行标准的自回归多项式采样预测。

# Market Simulato



# 金融量化

- AlphaAgent: LLM-Driven Alpha Mining with Regularized Exploration to Counteract Alpha Decay
- QuantaAlpha: An Evolutionary Framework for LLM-Driven Alpha Mining
- TradingAgents: Multi-Agents LLM Financial Trading Framework
- TradeFM: A Generative Foundation Model for Trade-flow and Market Microstructure (J.P. Morgan AI Research)
- **FactorEngine: A Program-level Knowledge-Infused Factor Mining Framework for Quantitative Investment**

# FactorEngine 程序级知识注入因子挖掘框架

这篇论文提出了一种名为 **FactorEngine (FE)** 的程序级知识注入因子挖掘框架。与传统的依赖预定义符号空间（如遗传规划）或缺乏可解释性的黑盒神经网络不同，FE 将因子挖掘转化为**图灵完备的程序（Python代码）进化问题**。

该方法的核心创新在于实现了“宏观-微观协同进化”，并通过三个解耦（逻辑修改与参数优化解耦、LLM定向搜索与贝叶斯搜索解耦、LLM资源与本地计算解耦）来大幅提升搜索效率和因子质量。

具体而言，这篇论文的方法由三个核心模块构成：

# 核心模块

## 1. 引导/初始化模块 (Bootstrapping Module)

该模块负责将金融研究报告中的非结构化专家知识和专家设计的因子转化为可执行的代码，构建出高质量的初始因子池。

- **多智能体提取与转化：** 利用一个闭环的多智能体系统，通过“理解到生成”的工作流，将研报中的核心金融思想提取为结构化的 JSON 和 LaTeX 伪代码。
- **自动代码生成与纠错：** 随后将伪代码转化为可执行的 Python 代码。系统包含执行与纠错机制，能够自动识别并修复代码中的逻辑或语法漏洞，确保初始因子的质量和可执行性。

# 核心模块

## 2. 进化模块 (Evolution Module)

这是框架的核心模块，采用**宏观-微观协同进化 (Macro-Micro Co-evolution)**机制，在一个组织为树状结构的因子池中不断迭代。进化过程分为四个阶段：

- **程序选择 (Program Selection)：** 将因子池组织为蒙特卡洛树结构，采用树的上限置信区间（UCT）标准来平衡探索（Exploration）与利用（Exploitation），挑选出当前最具潜力的节点（代码程序）进行下一步变异。
- **宏观想法生成 (Idea Generation/Macro Mutation)：** 这是 LLM 发挥作用的阶段。系统会为 LLM 提供**经验链 (Chain of Experience, CoE)**，即包含成功经验与失败教训的历史进化轨迹。LLM 结合这些反馈与自身的金融先验知识，推理出程序结构的宏观修改建议（如改变计算逻辑、引入新特征）并确定参数的搜索范围。

# 核心模块

- **微观实施与参数优化 (Implementation/Micro Mutation)：** 将参数优化从 LLM 的推理中剥离，交由本地的贝叶斯搜索（如 TPE 算法）执行。这一阶段在本地通过多进程并行计算，自动微调代码中的超参数（如窗口大小、权重），极大降低了调用 LLM 的成本并避免了参数陷入局部最优。

- **反馈传播 (Feedback Propagation)：** 新程序经过历史市场数据回测后，会获得定量的绩效评估。LLM 会总结这次变异的核心逻辑及优劣，连同定量指标一起反向传播更新树节点的价值（Q值和访问次数N），指导未来的搜索方向。

- **多岛屿进化机制 (Multi-island Evolution)：** 为了避免 LLM 随机性导致的低效探索，系统并行运行多个独立的进化过程（岛屿），并定期交换它们排名前 3 的优秀因子，从而在保持种群多样性的同时加速优质基因的传播。

# 核心模块

## 3. 整合模块 (Integration Module)

该模块负责从进化产生的庞大因子池中筛选出精英节点，用于最终的多因子建模。

- 为了降低通过计算相关性来筛选因子的计算成本，论文采用了一种轻量级的阈值过滤机制。
- 系统设计了一个综合**适应度分数 (Fitness Score, FS)**，该分数同时考虑了线性预测能力（IC, ICIR）和基于排序的预测能力（Rank IC, Rank ICIR）的幅度和稳定性。
- 只有超过设定阈值的顶级因子及其最佳参数配置会被保留，最终融合输入到下游的机器学习模型（如 LightGBM）中生成投资组合交易信号。

# 量化因子挖掘问题的数学定义

## 1. 基础变量设定

- **股票池 (Universe)**：$S = \{s_1, s_2, \ldots, s_N\}$，包含 $N$ 只股票。
- **交易时间段**：$\mathcal{T} = \{t_1, t_2, \ldots, t_T\}$。
- **特征张量**：在回溯窗口长度为 $L$ 的情况下，输入到因子的原始市场特征被表示为三维张量 $X_{t-L+1:t} \in \mathbb{R}^{N \times L \times M}$，其中 $M$ 为特征的维度（如开高低收量 OHLCV）。
- **真实目标 (Ground-truth)**：在时间 $t$，股票池真实的未来收益向量为 $Y = \{y_{t,1}, y_{t,2}, \ldots, y_{t,N}\} \in \mathbb{R}^N$。

# 量化因子挖掘问题的数学定义

## 2. 因子与预测模型的数学映射

- **单因子映射**：一个因子被定义为一个映射函数 $f$，它将历史特征窗口转化为提前 $l$ 步的预测信号 $r_{t+l} \in \mathbb{R}^N$。公式表示为：

$$f(X_{t-L+1:t}) \rightarrow r_{t+l}$$

- **多因子聚合**：在实践中，会构建一个包含 $K$ 个因子的集合 $\{f_k\}_{k=1}^{K}$。它们的输出通过一个聚合函数 $g$（例如线性回归或 LightGBM 等神经网络）被融合成最终的复合预测信号 $z_t$：

$$z_t = g(f_1(X_{t-L+1:t}), \cdots, f_K(X_{t-L+1:t}))$$

# 量化因子挖掘问题的数学定义

### 3. 最终优化目标

设整个评估时间段内的复合预测信号集合为 $\mathcal{Z} = \{z_t\}_{t=L}^{T-l}$，对应的真实标签集合为 $\mathcal{Y} = \{Y_t\}_{t=L}^{T-l}$。

因子挖掘问题的终极数学目标是**搜索并构建一组最优的因子**，以最大化预先定义的性能评估指标 $\mathcal{R}(\mathcal{Z}, \mathcal{Y})$（例如信息系数 IC 或夏普比率等）。

# 核心方法的数学定义

## 1. 进化的强化学习抽象

整个代码程序进化框架被抽象为一个类似马尔可夫决策/强化学习的过程，定义为元组 $\mathcal{F} = (\mathcal{P}, \mathcal{E}, \phi)$：

- $\mathcal{P}$：树状结构的程序搜索空间。
- $\mathcal{E}$：代码的执行与验证环境。
- $\phi$：大语言模型（Agent）自身携带的参数化金融先验知识。

# 核心方法的数学定义

## 2. 宏观层：基于 UCT 的程序选择 (Program Selection)

生成的代码因子被组织成一棵蒙特卡洛搜索树。在每次进化迭代时，系统需要选择最有潜力的“程序节点”进行变异。为了平衡“探索未知”和“利用已知优秀结果”，论文采用了**树的上限置信区间 (UCT) 公式**对候选节点 $v$ 进行打分：

$$UCT(v) = Q(v) + c\sqrt{\frac{\ln N_{parent(v)}}{N_v}}$$

- $Q(v)$：以节点 $v$ 为根的子树中所有代码在回测中的经验平均奖励（性能得分）。
- $N_v$ 与 $N_{parent(v)}$：分别是节点 $v$ 及其父节点的被访问/评估次数。
- $c$：探索-利用权衡常数（论文中设定为 $\sqrt{2}$）。

# 核心方法的数学定义

## 3. 宏观层：经验链的路径评估 (Path Evaluation for Idea Generation)

为了给大模型（LLM）提供高质量的修改灵感，系统会从全局树中提取候选的经验路径 $\{p_i\}_{i=1}^{n}$ 提供给大模型。为了确保提取的历史路径既有高性能又不过度重复，系统通过以下公式综合评估并选择路径：

- **有效性分数 (Effectiveness Score)**，衡量路径中节点的平均历史性能：

$$S_{eff}(p_i) = \frac{1}{|p_i|} \sum_{m=1}^{|p_i|} Score(p_{i,m})$$

- **覆盖率分数/重合度惩罚 (Coverage Score)**，衡量候选路径 $p_i$ 与当前进化链 $C$ 的重叠程度（其中 $\Phi = p_i \cap C$）：

$$S_{cov}(p_i) = \alpha \frac{|\Phi|}{|p_i|} + \beta \frac{|\Phi|}{|C|}$$

$$S_{total}(p_i) = S_{eff}(p_i) - S_{cov}(p_i)$$

# 核心方法的数学定义

## 4. 微观层：贝叶斯参数优化 (Bayesian Hyperparameter Optimization)

这是该论文的一大亮点：**将具体的数值参数调优从大语言模型中剥离出来，交由纯数学的贝叶斯搜索在本地完成。**

对于 LLM 确定了代码逻辑结构（但参数未知）的程序 $P$，以及其参数空间 $\Theta$，这是一个标准的最优化问题：

$$\theta^* = \arg\max_{\theta \in \Theta} f(P, \theta)$$

系统采用 TPE（树状结构 Parzen 估计器）等贝叶斯算法，对目标函数进行概率建模，并通过最大化**期望改进 (Expected Improvement, EI)** 来高效逼近最优参数组（如均线的天数、归一化的权重等）：

$$EI(\theta) = \int_{-\infty}^{\infty} \max(y^* - y, 0) \cdot p(y|\theta) dy$$

# 核心方法的数学定义

## 5. 因子过滤与适应度分数 (Fitness Score)

在进化出成百上千的程序因子后，需要筛选出“精英节点”投入实盘或回归模型。为了避免计算全量相关性矩阵的高昂开销，论文定义了一个轻量级的加权**适应度分数 (FS)**，对因子的多维表现进行综合降维：

$$FS = \frac{1}{4}(IC \times 10 + ICIR + RIC \times 10 + RICIR)$$

- 该公式联合考虑了因子的**线性预测能力**（IC 与其时间稳定性 ICIR）以及**基于排名的非线性预测能力**（Rank IC 与 RICIR），并且为了量级对齐对 IC/RIC 乘以了 10。只有 FS 超过 0.4 的顶级因子及其最佳参数配置才会被截留。

本节内容

CONTENTS

一、金融量化

**二、物理AI**

# 说明

- 本节内容涉及非常先进、打破传统的算法，尚未发表，暂不在网站公布

# 问题和讨论