GAMMA: MARKET-GROUNDED HYPOTHESIS GENERATION AND GROUPWISE FEEDBACK FOR LLM-BASED ALPHA MINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Alpha factors are scoring rules that rank assets for portfolio construction. Recent LLM agents automate factor discovery through three stages: (1) economic hypothesis generation, (2) factor implementation, and (3) evaluation and evolution. However, existing methods face two limitations. First, hypothesis generation often relies on the LLM's general financial knowledge without grounding hypotheses in the target-market's training data. Second, evaluating a hypothesis through a single factor often leads to prematurely discarding a valid hypothesis, mistaking a poor implementation for a flawed economic rationale. We introduce **GAMMA**, **G**roupwise **A**lpha **M**ining via **M**arket-grounded **A**gents, to address these limitations. During hypothesis generation, GAMMA analyzes the target market's feature distributions, cross-sectional relationships, and temporal dependencies, grounding each hypothesis in statistically verified evidence. During factor implementation, GAMMA constructs a group of diverse factor implementations for each hypothesis. During evaluation and evolution, GAMMA clusters these implementations by stock-ranking similarity, summarizes feedback at the cluster level, and expands successful clusters into refined sub-hypotheses. On CSI 300, GAMMA achieves the highest net annualized excess return of 13.29% and an information ratio of 1.471 among 17 compared methods. CSI 300 factor expressions also retain predictive value when transferred to CSI 500. Additional experiments show that GAMMA remains effective on both CSI 500 and S&P 500, demonstrating its generalizability across Chinese and U.S. equity markets.

1 INTRODUCTION

Quantitative portfolios rely on alpha factors—scoring rules that rank assets from historical market data and guide portfolio construction. Because predictive relationships change over time, factor libraries require continuous renewal. Traditionally, researchers designed alpha factors by translating economic intuitions into mathematical expressions (Kakushadze, 2016; Yu et al., 2023). This process preserves an interpretable connection between an economic rationale and its measurement, but manual factor design is difficult to scale. Automated alpha mining has therefore explored evolutionary search (Cui et al., 2021; Zhang et al., 2020), symbolic regression (Petersen et al., 2021; Shi et al., 2025), and reinforcement learning (Yu et al., 2023). However, these approaches primarily rely on statistical objectives rather than explicit economic reasoning to search a vast space of mathematical expressions, often evaluating many candidates to identify only a small number of useful factors.

Recent large language model (LLM) agents bring explicit economic reasoning into factor discovery. They formulate economic hypotheses to guide factor construction, translate these hypotheses into executable factors, and iteratively refine their search from empirical outcomes. A typical LLM-based mining loop therefore spans three stages: *economic hypothesis generation*, *factor implementation*, and *evaluation and evolution*.

Despite recent progress, current paradigms still face two limitations. First, existing agents exhibit ungrounded hallucinations. Because raw market series cannot be directly ingested into language model context windows, agents mainly rely on static schemas or qualitative reports. Without verifying whether these premises hold in target numerical data, agents hallucinate non-existent regularities,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

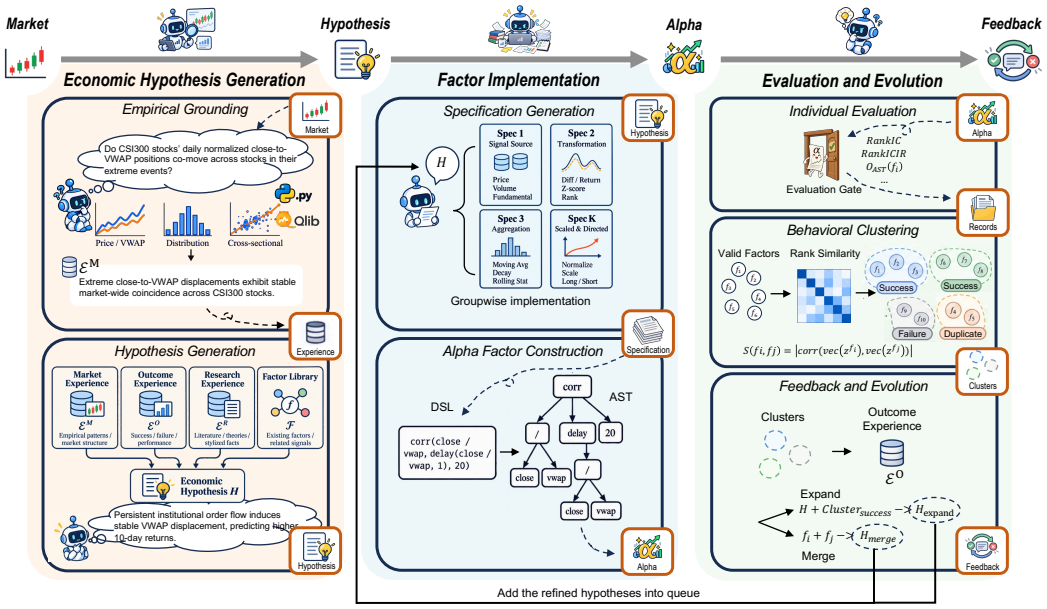

Figure 1: Overview of GAMMA's three-stage search loop. Stage I probes the target market's data and combines the findings to propose an economic hypothesis H. Stage II develops K distinct specifications for H and converts each into an executable factor. Stage III evaluates implementations individually, clusters factors with ranking exposures, updates outcome experience and evolves at the cluster level.

causing factor search to degenerate into blind trial-and-error. Second, current systems suffer from misaligned credit assignment. While the conceptual search unit is an economic hypothesis, feedback evaluates a single formulaic implementation. A sound hypothesis may fail simply due to an unsuitable proxy, normalization, or lookback scale, causing premature rejection. Furthermore, reflecting on isolated expressions overlooks functional redundancy among syntactically distinct formulas.

To address these limitations, we propose **GAMMA**, **G**roupwise **A**lpha **M**ining via **M**arket-grounded **A**gents (shown in Figure 1). During hypothesis generation, GAMMA executes sandboxed queries on target training features to distill verified market experience without label leakage. During factor implementation, each hypothesis is expanded into multiple measurement specifications that preserve the same parent economic rationale while varying proxies, operators, normalizations, and temporal structures. During evaluation and evolution, GAMMA clusters implementations by cross-sectional ranking similarity, summarizes feedback at the cluster level, and expands successful clusters into specialized sub-hypotheses while periodically merging low-correlation factors.

We evaluate GAMMA against 16 baselines spanning machine learning, deep neural networks, symbolic and evolutionary alpha mining, and LLM-based agents. On CSI 300 over the 2022–2025 out-of-sample period, GAMMA achieves an annualized excess return of 13.29% and an information ratio of 1.471 after transaction costs. Controlled ablations further separate the roles of the two mechanisms: market grounding improves the empirical basis of hypothesis generation, whereas groupwise feedback improves search efficiency and the attribution of outcomes across implementations. The resulting factor library also retains predictive value when transferred from CSI 300 to CSI 500, and independent mining runs on CSI 500 and S&P 500 show that the framework remains applicable across distinct market environments.

Our contributions are:

- We identify the credit assignment failure caused by single-formula evaluation alongside the lack of empirical market grounding in agentic alpha mining. Thus we propose **GAMMA**, the novel framework that anchors hypothesis generation in target market data while shifting the unit of reflection from isolated code expressions to behavioral implementation clusters.

- We introduce two core mechanisms: *Market Grounding* statistically verifies empirical regularities via code before hypothesis generation, while *Group-and-Cluster Exploration* groups factor implementations by ranking similarity into reflection units. In GAMMA, hypotheses are the unit of reasoning, factors are the unit of market evaluation, and behavioral clusters become the unit of feedback.
- Extensive experiments show that GAMMA achieves strong downstream portfolio performance on CSI 300 while reducing redundant search. Ablations demonstrate complementary roles for market grounding and groupwise feedback, and additional transfer and independent-mining evaluations show that the discovered factors and search framework remain useful beyond the original CSI 300 mining setting.

2 Related Work

Automated Formulaic Alpha Mining Automated approaches search over mathematical expressions or programs (Chen et al., 2026) to discover predictive factors. Representative methods in this category apply genetic programming to mutate formulaic primitives and policy gradient reinforcement learning (Petersen et al., 2021) to navigate discrete symbolic search spaces, while AlphaGen (Yu et al., 2023) uses reinforcement learning to discover factors that improve a combination model and AlphaForge (Shi et al., 2025) combines factor generation with dynamic weighting of the resulting factors. These approaches efficiently explore large combinatorial spaces, but do not explicitly organize exploration around economic hypotheses and their alternative measurements. GAMMA makes this relationship explicit, connecting each hypothesis to multiple implementations and using their outcomes to guide subsequent research.

LLM-Based Alpha Mining To mitigate the opacity and overfitting of symbolic search, recent frameworks (Shi et al., 2026a;b) use LLMs to guide factor discovery with financial knowledge and economic reasoning. FAMA (Li et al., 2024) reuses diverse examples and successful mining experience through Cross-Sample Selection and Chain-of-Experience. FactorMAD (Duan et al., 2025) introduces multi-agent debate for interpretable factor mining. AlphaAgent (Tang et al., 2025) constrains complexity and redundancy while checking alignment between hypotheses and factors, and RD-Agent(Q) (Li et al., 2025) coordinates agents to jointly optimize factors and prediction models. AlphaBench (Luo et al., 2026) evaluates LLMs across factor generation, evaluation, and search. However, general financial theory and literature cannot verify whether an assumed pattern actually exists in the target market data. GAMMA addresses this need by using agents to actively query and screen empirical market regularities in a code sandbox, then using the supported observations to inform hypothesis generation.

Evolution and Research Memory Recent work also studies how iterative search (Wang et al., 2026a) and accumulated experience (Wang et al., 2026b; Yu et al., 2026) can improve factor discovery. Representative efforts include code-level refinement and recombination (Liu et al., 2026b), genetic trajectory evolution via localized mutations and crossover (Han et al., 2026), multi-turn search policy learning via agentic reinforcement learning (Tang et al., 2026), and continuous research knowledge accumulation across discovery cycles (Liu et al., 2026a). GAMMA focuses on feedback organization within each hypothesis. It generates multiple implementations and clusters them by cross-sectional ranking behavior, reducing reliance on an isolated formula when determining how a hypothesis should evolve. Each cluster provides a reflection unit, and each successful cluster generates a refined sub-hypothesis. Behavioral similarity therefore guides both experience distillation and evolutionary branching, extending its role beyond factor deduplication.

3 Methodology

3.1 Problem Setup and Framework Overview

Formulaic alpha mining searches a discrete space of executable expressions for predictive factors. Let $X \in \mathbb{R}^{N \times T \times D}$ contain market observations for N assets over T trading dates, where D is the number of observed variables, such as prices and trading volume. An alpha factor f uses the

Table 1: Operational summary of the GAMMA research lifecycle.

Operation	Inputs	Outputs
Stage I: Economic hypothesis generation		
Empirical grounding	Target-market features X_{train}	Market experience $\mathcal{E}^M$
Hypothesis generation	Experiences $\mathcal{E}^R$, $\mathcal{E}^M$, $\mathcal{E}^O$; Factor library $\mathcal{F}$	Economic Hypothesis H
Stage II: Groupwise factor implementation		
Specification generation	Economic Hypothesis H	Specifications $\{s_k\}$
Alpha Factor construction	Specifications $\{s_k\}$	Factors $\{f_k\}$
Stage III: Evaluation and evolution		
Individual evaluation	Factors $\{f_k\}$; Factor library $\mathcal{F}$	Evaluation records; Admitted factors
Behavioral clustering	Ranking exposures $\{z_t^f\}$	Clusters $\{\mathcal{C}_c\}$
Feedback and evolution	Cluster records; Parent hypotheses H; Factor Library $\mathcal{F}$	Updated $\mathcal{E}^O$; Expanded H_{expand}; Merged H_{merge}

observations available by date t, denoted by $X_{\leq t}$, to compute asset scores,

$$a_t^f = f(X_{\leq t}) \in \mathbb{R}^N, \tag{1}$$

where $a_{t,i}^f$ is the score assigned to asset i on date t. These scores are used to rank assets for portfolio construction. The alpha mining goal is to build a nonredundant factor library $\mathcal{F} = \{f_j\}_{j=1}^J$, where J is the target library size.

GAMMA organizes this search into three stages, as shown in Figure 1 and summarized in Table 1. Stage I examines the target market's data and combines the findings with prior experience to propose an economic hypothesis. Stage II constructs several factors that measure the same hypothesized mechanism in different ways. Stage III evaluates factors individually for library admission, groups implementations by ranking similarity, and summarizes their results at the cluster level. Successful clusters produce refined hypotheses, while periodic merging combines the economic rationales of library factors to propose composite hypotheses. These refined hypotheses return to Stage II, guiding subsequent alpha factor generation.

3.2 Stage I: Economic hypothesis generation

Stage I obtains empirical evidence from the target market and uses it to guide economic hypothesis generation.

Active Empirical Grounding. Active Empirical Grounding (AEG) probes X_{train} for statistically supported market regularities and distills them into the market experience store $\mathcal{E}^M$. The goal is not to predict returns directly, but to establish whether descriptive premises that may support later economic reasoning are present in the target data.

AEG examines six structural dimensions: univariate distributions, cross-sectional dispersions, temporal persistence, asset co-movements, inter-feature covariances, and regime transitions. For a selected dimension, the system constructs a research query q and compiles it into a structured measurement specification $d(q)$.

A deterministic runner evaluates $d(q)$ on X_{train} in a masked sandbox that excludes target returns y_{t+h}, preventing direct label exposure during grounding. To reduce sensitivity to noisy observations, each measurement is screened through three validation gates: cross-sectional asset coverage, temporal consistency across contiguous time blocks, and bootstrap uncertainty bounds. Measurements satisfying the gates are summarized into compact market observations and stored in $\mathcal{E}^M$.

Hypothesis Generation. The hypothesis agent combines four complementary information sources: research experience $\mathcal{E}^R$ containing economic principles, market experience $\mathcal{E}^M$ containing empirical target-market observations, outcome experience $\mathcal{E}^O$ containing success and failure modes from previous search rounds, and the factor library $\mathcal{F}$ indicating explored signal families.

Conditioned on this context, the agent generates a structured hypothesis H describing (i) a proposed economic mechanism, (ii) an observable signal through which the mechanism may be measured, and (iii) its expected direction. The hypothesis is then passed to Stage II, where the proposed mechanism is instantiated through multiple alternative measurements.

3.3 Stage II: Groupwise factor implementation

Stage II converts the hypothesis from Stage I into a group of executable alpha factors. These factors use different measurements while preserving the hypothesis's economic rationale.

Group Specification Generation. Given an economic hypothesis H, the system generates an ensemble of K candidate measurement specifications $\{s_k\}_{k=1}^K$. Each specification defines the target variables, measured statistical property, temporal aggregation, cross-sectional normalization, and qualitative horizon required to operationalize the hypothesis. The specifications systematically vary proxies, operators, and temporal choices while preserving the mechanism stated in H. Candidates are filtered for attribute completeness and pairwise functional divergence so that the resulting group represents meaningfully different implementations rather than superficial syntactic variations.

Alpha Factor construction. The external interface of this operation maps each semantic specification s_k to an executable factor f_k:

$$H \xrightarrow{\text{diversify}} \{s_k\}_{k=1}^K \xrightarrow{\text{DSL}} \{e_k\}_{k=1}^K \xrightarrow{\mathcal{G}} \{T_k\}_{k=1}^K \xrightarrow{\text{Compile}} \{f_k(X)\}_{k=1}^K, \tag{2}$$

Each verified specification s_k is independently instantiated into a domain-specific language (DSL) expression e_k. The DSL restricts primitive operators to registered market variables and causal, backward-looking time-series transformations over individual asset histories. The expression e_k is parsed via a formal grammar $\mathcal{G}$ into an Abstract Syntax Tree (AST) T_k, which is statically verified and compiled into an executable alpha factor f_k. Appendix D.2 documents the internal representation and operator rules.

The resulting factors enter Stage III, where evaluation and behavioral clustering determine whether distinct measurement designs yield predictive and behaviorally distinct signals.

3.4 Stage III: Evaluation and evolution

Stage III separates two decisions: *which individual factors should enter the factor library* and *which implementations provide distinct evidence for evolving*. The former is based on factor-level predictive quality and library redundancy; the latter is based on realized ranking behavior.

Individual Evaluation. Each candidate factor $f \in \mathcal{F}_H$ is evaluated on training and validation splits for execution coverage and predictive efficacy (e.g., RankIC and RankICIR), followed by redundancy checking against the active library $\mathcal{F}$. Candidate factors failing statistical significance thresholds are rejected. Detailed metric definitions and admission protocols are provided in Appendix D.2.

Structural similarity $O_{\text{AST}}(f_k, f_j)$ between candidate f_k and library factor $f_j \in \mathcal{F}$ measures the normalized multiset overlap of canonical AST subtrees:

$$O_{\text{AST}}(f_k, f_j) = \frac{\sum_u \min\{c_{f_k}(u), c_{f_j}(u)\}}{\max\{\sum_u c_{f_k}(u), \sum_u c_{f_j}(u)\}}, \tag{3}$$

where $c_f(u)$ denotes the multiplicity of canonical subtree u in factor f, preserving node types, operators, constants, and ordered child structures.

Qualified factors will enter the factor library. All candidates retain their evaluation records, and no search decision uses test data. Appendix D.2 specifies the admission criteria.

Behavioral Clustering. GAMMA clusters implementations of the same hypothesis by their validation period ranking exposures. Let z_t^f denote the cross-sectional percentile ranks of factor f on date t. After aligning observations by stock and date, pairwise similarity is

$$S(f_i, f_j) = \left|\mathrm{corr}\left(\mathrm{vec}(z^{f_i}), \mathrm{vec}(z^{f_j})\right)\right|. \tag{4}$$

Absolute correlation treats sign-reversed rankings as the same behavioral axis, while evaluation records preserve their orientations. Complete-linkage clustering merges clusters A and B only when

$$\min_{f_i \in A,\, f_j \in B} S(f_i, f_j) \geq \tau_{\mathrm{cluster}}. \tag{5}$$

This prevents chains of partially similar implementations from forming one feedback unit.

All evaluated candidates with usable exposures participate, including those that fail predictive screening. A cluster receives a **Success** verdict if at least one member enters the factor library $\mathcal{F}$, **Duplicate** if none enters but at least one qualifies and is redundant, and **Failure** otherwise.

Feedback and Evolution. For each cluster $\mathcal{C}_c$, a reflection agent summarizes its specifications, evaluation outcomes, factor orientations, and redundancy status into an experience record e_c. The resulting feedback identifies which measurements received empirical support, which produced redundant behavior, and which implementation choices failed, without treating any single implementation outcome as a direct verdict on the complete parent hypothesis. The record is stored in outcome experience $\mathcal{E}^O$ and informs subsequent search. GAMMA then updates the hypothesis search space through two operators: *expansion* and *merging*.

Expand Each successful cluster generates a refined sub-hypothesis and enqueues it into the hypothesis queue for further exploration. The reflection agent takes the parent hypothesis H, the admitted factors $\mathcal{F}_c$ in that cluster, and the active library summary $\Pi(\mathcal{F})$ as inputs to generate:

$$H_{expand} = \mathrm{Expand}(H, \mathcal{F}_c, \Pi(\mathcal{F})). \tag{6}$$

By introducing targeted perturbations to the successful parent hypothesis, H_{expand} formulates a more specific research question by refining measurement proxies, temporal structures, or applicable market conditions, enabling the parent hypothesis to branch into multiple specialized sub-directions.

Merge To combine complementary economic mechanisms across different research branches, any successful search round with an odd index n additionally triggers cross-hypothesis merging. The system selects the unmerged factor pair in the library $\mathcal{F}$ exhibiting the lowest absolute validation correlation, subject to $S(f_i, f_j) \leq \tau_{\mathrm{merge}}$. Using their underlying expressions and economic mechanisms, the merging agent generates:

$$H_{\mathrm{merge}} = \mathrm{Merge}\left(\{(f_j, H_j)\}_{j \in J}, \Pi(\mathcal{F})\right), \tag{7}$$

where J indexes the selected factor pair.

The merging agent formulates an economic rationale explaining why combining these low-correlation factors yields complementary predictive power. Merge thus operates at the conceptual level of economic rationales rather than directly aggregating factor scores. A successful round can enqueue both expanded and merged hypotheses. Both return to Stage II for diverse implementations and subsequent evaluation, while updated outcome experience informs future hypothesis generation, ensuring effective, diverse, and sustained multi-branch exploration across the search space.

4 Experiments

Our experiments examine four questions: (1) whether GAMMA discovers useful factor libraries under a common out-of-sample evaluation; (2) how market grounding and groupwise feedback separately contribute to search quality; (3) whether behavioral clustering is necessary for efficient and reliable feedback assignment; and (4) whether the resulting factors and search procedure remain effective beyond the original CSI 300 setting.

4.1 Experiment Setup

Dataset. We use point-in-time constituents of the CSI 300 index and daily OHLCV+VWAP market data retrieved from Qlib (Yang et al., 2020). Training spans 2010/01–2019/11, validation

2020/01–2021/11, and testing 2022/01/01–2025/12/31, with one-month embargoes between splits to prevent temporal data leakage. The target is the 10-day open-to-open return from $t+1$ to $t+11$.

Metrics. We report RankIC, Median RankIC, RankICIR, factor-level sign-flip rate, net annualized return (AR), annualized excess return (AER), information ratio (IR), and maximum drawdown (MDD); Appendix A defines them.

Baselines. We compare GAMMA with 16 baselines under unified data splits and backtesting protocols. Machine- and deep-learning predictors (Chen & Guestrin, 2016; Cho et al., 2014; Ke et al., 2017; Prokhorenkova et al., 2018; Vaswani et al., 2017) use Alpha360 features; program-search and LLM-based mining methods each produce 50 valid factors; and Alpha101 and Alpha158 use predefined libraries. GAMMA and all LLM baselines use `deepseekv4-flash`. Appendix A provides implementation details, score construction, and evaluation settings.

4.2 Main Results

Table 2 compares predictive and downstream portfolio performance on the 2022–2025 CSI 300 test period. GAMMA attains the highest net AER of 13.29% and the highest portfolio IR of 1.471 among the compared methods. Its advantage is more pronounced in downstream portfolio performance than in average factor-level predictive metrics: DSO obtains a higher mean RankIC (0.0716 versus 0.0637), and GRU obtains a higher RankICIR (0.4705 versus 0.3340), while GAMMA achieves substantially higher net AER than both methods. Figure 2 shows the corresponding wealth curves.

Table 2: Performance comparison on CSI 300 over the 2022–2025 test period. Higher values are better except for sign flip and MDD. Within each column, ranks among available values are shown by progressively lighter blue: **Best (1)**, Second Best (2), Good (3–5), and Average (6–10); lower ranks are unshaded. Table 2 reports the primary run; multi-seed stability is detailed in Section 4.3.

Models		Factor Predictive Metrics				Strategy Performance			
Category	Model	RankIC ↑	Median RankIC ↑	RankICIR ↑	Sign flip (%) ↓	AR (%) ↑	AER (%) ↑	IR ↑	MDD (%) ↓
Machine Learning	CatBoost	0.0547	0.0548	0.3296	—	3.14	3.21	0.411	29.83
	XGBoost	0.0337	0.0364	0.2938	—	4.14	4.22	0.494	33.73
	LightGBM	0.0524	0.0508	0.3261	—	5.76	5.84	0.719	28.96
Deep Learning	MLP	0.0118	0.0146	0.1143	—	−0.28	−0.20	−0.032	33.48
	GRU	0.0671	0.0666	**0.4705 (1)**	—	8.22	8.30	1.286	26.19
	Transformer	0.0670	0.0680	0.4093	—	7.12	7.20	0.902	19.76 (2)
Alpha Library	Alpha101	0.0428	0.0436	0.3053	64.63	6.27	6.35	1.014	28.81
	Alpha158	0.0590	0.0613	0.3229	47.47	3.86	3.94	0.541	23.06
Automated Alpha Mining	GP	0.0331	0.0366	0.1759	**0.00 (1)**	1.25	1.33	0.160	32.48
	DSO	**0.0716 (1)**	**0.0768 (1)**	0.3232	58.00	5.67	5.74	0.673	22.08
	AlphaGen	0.0431	0.0471	0.2673	70.00	8.02	8.10	1.095	24.71
	AlphaForge	0.0608	0.0714	0.3032	40.00	0.88	0.96	0.119	22.87
LLM-based Alpha Mining	Vanilla LLM	0.0279	0.0291	0.1634	38.00	2.97	3.04	0.364	28.46
	AlphaBench-EA	0.0528	0.0577	0.3032	68.00	9.46	9.54	0.963	30.36
	RD-Agent(Q)	0.0594	0.0617	0.3417	72.00	3.60	3.68	0.438	**19.46 (1)**
	AlphaAgent	0.0516	0.0582	0.3027	28.00 (2)	1.29	1.37	0.165	33.82
	GAMMA (ours)	0.0637 (4)	0.0761 (2)	0.3340 (4)	38.00 (3)	**13.21 (1)**	**13.29 (1)**	**1.471 (1)**	22.12 (4)

RankIC evaluates full-universe rankings, whereas portfolio returns also depend on directional stability, upper-tail selection, factor redundancy, turnover, and execution costs. GAMMA's median RankIC is close to DSO's (0.0761 versus 0.0768), with a lower sign-flip rate (38% versus 58%). Among its 50 factors, the 1,225 validation-period pairs have mean and median absolute rank-exposure correlations of 0.157 and 0.117, with effective rank 27.3 (Appendix Figure 7). These diagnostics characterize the directional stability and behavioral diversity of the discovered library.

Together, these results indicate that GAMMA constructs a factor library that is both predictive and behaviorally diverse. This is consistent with the framework's objective: rather than optimizing one

formula in isolation, GAMMA searches across alternative implementations and organizes feedback according to the distinct market behavior they realize.

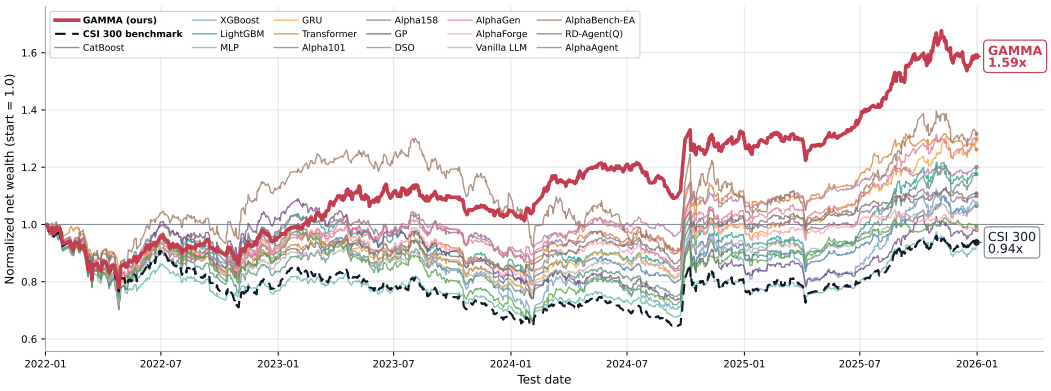

Figure 2: Net wealth on CSI 300 from 2022-2025 under the shared portfolio evaluator. Curves include transaction costs and begin at 1.0.

4.3 Ablation Study

We conduct a controlled 2×2 ablation of market grounding (AEG) and groupwise feedback, with Table 3 reporting means over three independent mining runs. AEG alone raises net AER from 2.63% to 5.02% and net IR from 0.143 to 0.610, but consumes 30.30M tokens. Conversely, groupwise feedback alone improves net AER to 5.39% while halving token usage to 7.65M. Full GAMMA combines both to achieve the highest net AER (12.31%) and net IR (1.332) with a positive 2×2 AER interaction of 4.53 points.

To isolate behavioral clustering, we retain AEG and groupwise generation but omit clustering. Net AER falls from 12.31% to 4.66%, while token usage nearly triples to 24.35M across 725 evaluated candidates. These results support behavioral clustering's role in reducing redundant reflections and improving feedback assignment.

Table 3: Ablation of AEG and groupwise feedback. Values are means over three independent mining runs, with standard deviations in parentheses.

Variant	Candidates	Admit (%)	Tokens (M)	RankICIR	Net AER (%)	Net IR
Base	901 (82)	5.58 (0.51)	16.50 (1.89)	0.2803 (0.0042)	2.63 (0.22)	0.143 (0.059)
AEG only	844 (125)	6.05 (0.93)	30.30 (3.43)	0.2769 (0.0058)	5.02 (0.36)	0.610 (0.041)
Group only	710 (69)	7.09 (0.69)	**7.65 (2.45)**	0.2834 (0.0064)	5.39 (0.43)	0.591 (0.045)
GAMMA	**653 (51)**	**7.69 (0.63)**	8.55 (1.07)	**0.3220 (0.0135)**	**12.31 (1.03)**	**1.332 (0.131)**

4.4 Stability and Cross-Universe Evaluation

Temporal Persistence and Horizon Sensitivity. We examine whether GAMMA's selected factors retain predictive structure across periods and target horizons. The 50-factor library achieves mean $|\mathrm{RankIC}|$ values of 0.0313, 0.0223, and 0.0218 across training, validation, and test splits, with 43 factors preserving their mean $|\mathrm{RankIC}|$ sign across all three splits (Figure 3a). Across 10-, 20-, and 30-day cumulative return targets, aggregate test RankIC increases from 0.0637 to 0.0782 (Figure 3b), indicating predictive value across horizons alongside directional consistency over time.

Cross-Universe Factor Library Transfer. To evaluate cross-universe generalizability, we apply the 50 frozen factor expressions mined from CSI 300 directly to point-in-time CSI 500 constituents without formulaic modification or re-selection. On the CSI 500 test set, the transferred library

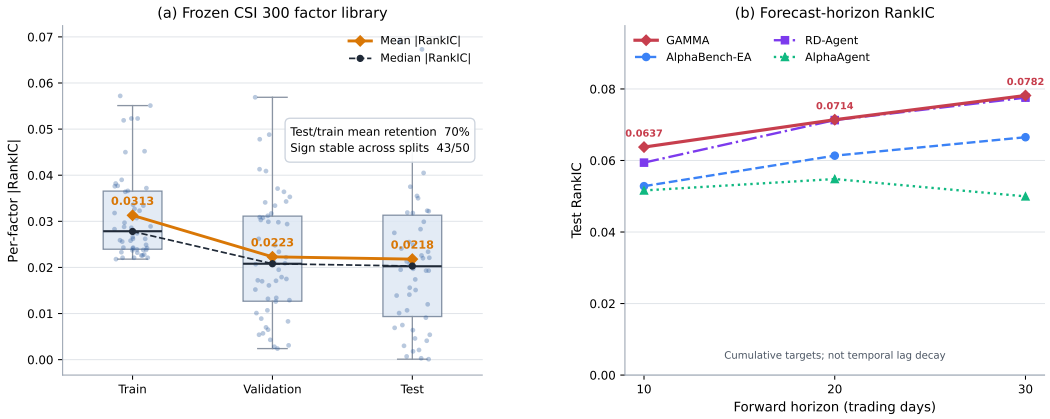

Figure 3: Stability diagnostics on CSI 300. Panel (a) shows $|\mathrm{RankIC}|$ for 50 selected factors across training, validation, and test. Panel (b) compares test RankIC under 10-, 20-, and 30-day targets.

achieves a RankIC of 0.0655, a Net AR of 9.74%, and a Net AER of 6.99% over the benchmark. These strong out-of-sample results demonstrate that the discovered factor mechanics capture intrinsic pricing logic rather than market cap noise, enabling seamless transfer across market universes.

Cross Market Mining. To evaluate whether the framework discovers effective pricing signals in distinct market environments, we conduct independent alpha mining on CSI 500 and S&P 500 constituents. Independent mining on CSI 500 yields a RankIC of 0.0568 and Net AR of 11.08% (Sharpe 0.707); mining on S&P 500 yields Net AR of 9.34% (Sharpe 0.498). Table 4 and Appendix Figure 4 report the results and wealth curves, demonstrating the framework's applicability across Chinese and U.S. equity markets.

Table 4: Results for cross-universe transfer and two separately mined settings. CSI 500 returns include China-equity transaction costs; S&P 500 results exclude costs. MDD is reported as positive.

Setting	RankIC	Annual Return (%)	Sharpe	MDD (%)
CSI 300 transfer to CSI 500	0.0655	9.74	0.582	29.64
CSI 500 mining	0.0568	11.08	0.707	24.95
S&P 500 mining	0.0306	9.34	0.498	21.98

5 Limitations

We note two limitations of this work. First, our evaluation relies primarily on daily OHLCV and VWAP data, without incorporating alternative data sources such as fundamentals or news. Second, trading simulations adopt standard fixed cost assumptions, which do not fully capture dynamic market impact during live execution. These limitations motivate evaluating the framework with richer data sources and more realistic execution models.

6 Conclusion

Evaluating an alpha hypothesis through a single formula risks misdiagnosing implementation flaws as conceptual failures. GAMMA addresses this by unifying empirical market grounding with group-wise hypothesis evolution, effectively decoupling abstract economic mechanisms from concrete formula artifacts. Rather than relying on brute-force candidate generation, the framework achieves high search efficiency by evaluating the fewest total candidates while eliminating redundant feedback across behaviorally similar implementations. Empirical results on CSI 300 demonstrate competitive RankICIR and Net AER, while cross-market evaluations on CSI 500 and S&P 500 confirm robust generalization across distinct microstructures.

REPRODUCIBILITY STATEMENT

We provide the complete source code, execution scripts, and environment setups in the supplementary material to ensure full reproducibility. Complete experimental details—including dataset splits, AST grammar, admission thresholds, and checkpoint logs—are thoroughly documented in Appendices A–D.

AI USE STATEMENT

In this work, we used generative AI tools to support research ideation and execution, retrieve and discover potentially relevant literature, draft portions of the paper, and edit or polish the manuscript's writing. All AI-assisted outputs were reviewed by the authors. We manually checked relevant literature and citations against source materials, independently evaluated AI-suggested ideas and research decisions, and verified the manuscript, implementation, and reported results. We take full responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with AI assistance.

REFERENCES

Binqi Chen, Hongjun Ding, Ning Shen, Taian Guo, Jinsheng Huang, Luchen Liu, and Ming Zhang. AlphaSAGE: Structure-aware alpha mining via GFlowNets for robust exploration. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/06115011841a8d500050261c0ac4b1e4-Abstract-Conference.html.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. Association for Computing Machinery, 2016. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179/.

Can Cui, Wei Wang, Meihui Zhang, Gang Chen, Zhaojing Luo, and Beng Chin Ooi. AlphaEvolve: A learning framework to discover novel alphas in quantitative investment. In *Proceedings of the 2021 International Conference on Management of Data*, pp. 2208–2216. Association for Computing Machinery, 2021. doi: 10.1145/3448016.3457324. URL https://doi.org/10.1145/3448016.3457324.

Yitong Duan, Chuheng Zhang, and Jian Li. FactorMAD: A multi-agent debate framework based on large language models for interpretable stock alpha factor mining. In *Proceedings of the 6th ACM International Conference on AI in Finance*, pp. 605–613, 2025. doi: 10.1145/3768292.3770377. URL https://doi.org/10.1145/3768292.3770377.

Jun Han, Shuo Zhang, Wei Li, Yifan Dong, Tu Hu, Yumo Zhu, Xiaomin Yu, Xin Guo, Zhaowei Liu, Kunyi Wang, Jingping Liu, Tianyi Jiang, Ruichuan An, Sen Hu, Zhi Yang, Ronghao Che, and Huacan Wang. QuantaAlpha: An evolutionary framework for LLM-driven alpha mining. *arXiv preprint arXiv:2602.07085*, 2026. doi: 10.48550/arXiv.2602.07085. URL https://arxiv.org/abs/2602.07085v3.

Zura Kakushadze. 101 formulaic alphas. *Wilmott*, 2016(84):72–81, 2016. doi: 10.1002/wilm.10525. URL https://doi.org/10.1002/wilm.10525.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, pp. 3146–3154. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

Yuante Li, Xu Yang, Xiao Yang, Xisen Wang, Weiqing Liu, and Jiang Bian. R&D-Agent-Quant: A multi-agent framework for data-centric factors and model joint optimization. In *Advances in Neural Information Processing Systems*, volume 38, pp. 131781–131830, 2025. doi: 10.52202/085713-3967. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/ac5c2b6e423883cbcacbcccf88491b78-Abstract-Datasets_and_Benchmarks_Track.html.

Zhiwei Li, Ran Song, Caihong Sun, Wei Xu, Zhengtao Yu, and Ji-Rong Wen. Can large language models mine interpretable financial factors more effectively? a neural-symbolic factor mining agent model. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3891–3902. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.233. URL https://aclanthology.org/2024.findings-acl.233/.

Fengyuan Liu, Yuchen Fu, Yuqi Wang, and Qi Liu. XALPHA: A memory-driven AI quant researcher for hypothesis-to-code alpha discovery. *arXiv preprint arXiv:2607.08332*, 2026a. doi: 10.48550/arXiv.2607.08332. URL https://arxiv.org/abs/2607.08332v2.

Fengyuan Liu, Yi Huang, Sichun Luo, Yuqi Wang, Yazheng Yang, Xinye Li, Zefa Hu, Junlan Feng, and Qi Liu. Cognitive alpha mining via LLM-driven code-based evolution. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11715–11749. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.acl-long.538. URL https://aclanthology.org/2026.acl-long.538/.

Haochen Luo, Ho Tin Ko, Jiandong Chen, David Sun, Yuan Zhang, and Chen Liu. AlphaBench: Benchmarking large language models in formulaic alpha factor mining. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/4f3820576130a8f796ddbf204c841487-Abstract-Conference.html.

Brenden K. Petersen, Mikel Landajuela Larma, Terrell N. Mundhenk, Claudio Prata Santiago, Soo Kyung Kim, and Joanne Taery Kim. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=m5Qsh0kBQG.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. CatBoost: Unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, volume 31, pp. 6638–6648. Curran Associates, Inc., 2018. URL https://papers.neurips.cc/paper_files/paper/2018/hash/14491b756b3a51daac41c24863285549-Abstract.html.

Hao Shi, Weili Song, Xinting Zhang, Jiahe Shi, Cuicui Luo, Xiang Ao, Hamid Arian, and Luis Angel Seco. AlphaForge: A framework to mine and dynamically combine formulaic alpha factors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12524–12532, 2025. doi: 10.1609/aaai.v39i12.33365. URL https://ojs.aaai.org/index.php/AAAI/article/view/33365.

Runze Shi, Shengyu Yan, Yuecheng Cai, and Chengxi Lv. Hubble: An LLM-driven agentic framework for safe, diverse, and reproducible alpha factor discovery. *arXiv preprint arXiv:2604.09601*, 2026a. doi: 10.48550/arXiv.2604.09601. URL https://arxiv.org/abs/2604.09601v2.

Yu Shi, Yitong Duan, and Jian Li. Navigating the Alpha Jungle: An LLM-powered MCTS framework for formulaic alpha factor mining. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 997–1005, 2026b. doi: 10.1609/aaai.v40i2.37069. URL https://ojs.aaai.org/index.php/AAAI/article/view/37069.

Ziyi Tang, Zechuan Chen, Jiarui Yang, Jiayao Mai, Yongsen Zheng, Keze Wang, Jinrui Chen, and Liang Lin. AlphaAgent: LLM-driven alpha mining with regularized exploration to counteract alpha decay. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 2813–2822, 2025. doi: 10.1145/3711896.3736838. URL https://doi.org/10.1145/3711896.3736838.

Ziyi Tang, Xuexiong Yin, Weixing Chen, Zechuan Chen, Yongsen Zheng, Wenxuan Ye, Keze Wang, and Liang Lin. AlphaAgentEvo: Evolution-oriented alpha mining via self-evolving agentic reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/741aab8b41a2987867acc9939ad50383-Abstract-Conference.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Haozengran Wang, Shuo Yin, Rong Fu, Mengting Zhang, Jiayi Zeng, and Simon James Fong. AlphaMaster: Dual-chain feedback for scalable and diverse alpha factor discovery. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 4882–4893, 2026a. doi: 10.1145/3770855.3818120. URL https://doi.org/10.1145/3770855.3818120.

Yanlong Wang, Jian Xu, Hongkang Zhang, Shao-Lun Huang, Danny Dongning Sun, and Xiao-Ping Zhang. FactorMiner: A self-evolving agent with skills and experience memory for financial alpha discovery. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 12327–12338, 2026b. doi: 10.1145/3770855.3818978. URL https://doi.org/10.1145/3770855.3818978.

Xiao Yang, Weiqing Liu, Dong Zhou, Jiang Bian, and Tie-Yan Liu. Qlib: An AI-oriented quantitative investment platform. *arXiv preprint arXiv:2009.11189*, 2020. doi: 10.48550/arXiv.2009.11189. URL https://arxiv.org/abs/2009.11189v1.

Hang Yu, Zifan Zheng, Jeff Z. Pan, Tongliang Liu, Zhiyong Wang, and Fengxiang He. AlphaMemo: Structured search-process memory for self-evolving alpha mining agents. *arXiv preprint arXiv:2606.20625*, 2026. doi: 10.48550/arXiv.2606.20625. URL https://arxiv.org/abs/2606.20625v1.

Shuo Yu, Hongyan Xue, Xiang Ao, Feiyang Pan, Jia He, Dandan Tu, and Qing He. Generating synergistic formulaic alpha collections via reinforcement learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5476–5486, 2023. doi: 10.1145/3580305.3599831. URL https://doi.org/10.1145/3580305.3599831.

Tianping Zhang, Yuanqi Li, Yifei Jin, and Jian Li. AutoAlpha: an efficient hierarchical evolutionary algorithm for mining alpha factors in quantitative investment. *arXiv preprint arXiv:2002.08245*, 2020. doi: 10.48550/arXiv.2002.08245. URL https://arxiv.org/abs/2002.08245v2.

A EXPERIMENTAL DETAILS

This appendix supplements the workflow in Figure 1 and Table 1. Appendix A defines the evaluation protocol and metrics used in Table 2; Appendix B provides additional curves and ablation diagnostics; Appendix C presents a complete logged search trace and a frozen-library factor record. Appendix D expands the three stages in Section 3.

A.1 DATA SPLITS AND SCORE CONSTRUCTION

The market environment is partitioned chronologically into three non-overlapping periods. The training period spans January 2010 through November 2019, the validation period covers January 2020 through November 2021, and the out-of-sample testing period runs from January 2022 through December 2025.

The prediction target $y_{it}^{(10)}$ represents the cumulative asset return across a ten-day forward horizon, computed from the market open price on trading day $t+1$ to the open price on day $t+11$. Measuring returns from day $t+1$ open ensures that trade decisions generated at the close of day t cannot access contemporaneous execution prices. Furthermore, because multi-day forward targets on adjacent dates share overlapping evaluation windows, back-to-back splits would introduce serial correlation leakage. We eliminate this autoregressive overlap by inserting a mandatory one-month buffer, or embargo window, between contiguous splits. Under this protocol, the empirical grounding module queries unlabeled training features exclusively, factor screening and behavioral clustering operate on training and validation splits, and the final factor library is evaluated on the test split without any retraining.

To maintain capacity equity across all discovery frameworks, every automated mining method and LLM agent discovers exactly 50 valid factors. These 50 factors serve as input features for a downstream linear meta-learner, fitted via Ridge regression with an L_2 penalty of $\alpha = 10$ and no intercept, to produce the aggregate cross-sectional ranking score z_{it}.

For predefined formula libraries, the available daily price and volume fields support 82 expressions from the WorldQuant Alpha101 library. We exclude the 19 formulas requiring external market capitalization or industry metadata, ensuring an identical observation space across all baselines. Alpha101 then selects the 50 expressions with the highest training absolute RankIC from these 82 formulas to match the 50-factor capacity. Alpha158 evaluates its standard set of 158 technical features. Subsetting Alpha158 to its top 50 features yields comparable portfolio returns, showing that feature count does not drive the benchmark difference. Supervised learning baselines, including gradient boosted trees and recurrent neural networks, do not produce symbolic formulas and ingest the dense Alpha360 tensor of 60-day historical price and volume observations to output ranking scores directly.

A.2 PORTFOLIO CONSTRUCTION AND TRADING FRICTIONS

The execution environment converts the model ranking scores z_{it} into daily asset holdings under a standardized portfolio policy across CSI 300, CSI 500, and S&P 500 universes. On each trading date, the policy selects the 50 assets with the highest predicted scores. To prevent excessive portfolio turnover and reflect realistic execution limits, the simulator restricts rebalancing to replacing at most five positions per day. All orders execute at official market open prices on day $t + 1$.

For the Chinese equity benchmarks (CSI 300 and CSI 500), the simulator deducts execution frictions of 5 basis points (0.05%) on buy orders and 15 basis points (0.15%) on sell orders. The higher fee on sales accounts for Chinese market regulatory stamp duties alongside broker commissions. For the S&P 500 benchmark, following standard empirical conventions in U.S. equity literature, the portfolio is evaluated without transaction fees against the official S&P 500 Price Index.

To ensure fair benchmark comparisons, every strategy reported in Table 2 must be evaluated through this identical portfolio simulator. This requirement governs our baseline eligibility. We omit CogAlpha and QuantaAlpha because neither method provided an immutable codebase or a tagged factor release at the time of our evaluation freeze. Transcribing unreplicated headline numbers from prior papers would violate controlled benchmarking, because reported metrics frequently reflect different execution assumptions, slippage models, or transaction fees. Our evaluation admits only methods

whose discovered factor expressions or native prediction scores can be deterministically replayed under this shared execution protocol.

A.3 Signal Metrics and Directional Stability

To evaluate prediction quality independently of portfolio trading rules, we analyze the cross-sectional correlation between model scores and realized returns. This evaluation operates on two distinct levels: first on the final integrated prediction scores, and second on the individual factors within each discovered library.

Model-Level Prediction Metrics. Let z_{it} denote the final integrated score assigned to asset i on trading day t, and let $y_{it}^{(h)}$ denote its h-day forward realized return. For the set of active assets $\mathcal{U}_t$ on date t, the daily evaluation statistics are defined as

$$\mathrm{IC}_t = \mathrm{Corr}_{i\in\mathcal{U}_t}\left(z_{it}, y_{it}^{(h)}\right), \tag{8}$$

$$\rho_t = \mathrm{Corr}_{i\in\mathcal{U}_t}\left(\mathrm{rank}(z_{it}), \mathrm{rank}(y_{it}^{(h)})\right). \tag{9}$$

Here, IC_t represents the linear Pearson correlation, while ρ_t represents the Spearman rank correlation (RankIC). Because asset returns contain substantial market noise, RankIC serves as the primary metric for evaluating the monotonic accuracy of the cross-sectional asset ranking.

Across all T trading dates in the evaluation window, we report three aggregate statistics in Table 2. The sample mean RankIC is computed as

$$\text{RankIC} = \frac{1}{T}\sum_{t=1}^{T}\rho_t. \tag{10}$$

To evaluate typical performance without sensitivity to extreme single-day market movements, the Median RankIC is computed as

$$\text{Median RankIC} = \mathrm{median}\left(\{\rho_1, \rho_2, \ldots, \rho_T\}\right). \tag{11}$$

To quantify the stability of ranking accuracy across time, the RankIC Information Ratio measures the empirical signal-to-noise ratio:

$$\text{RankICIR} = \frac{\overline{\rho}}{s(\rho)} = \frac{\frac{1}{T}\sum_{t=1}^{T}\rho_t}{\sqrt{\frac{1}{T-1}\sum_{t=1}^{T}(\rho_t-\overline{\rho})^2}}, \tag{12}$$

where $s(\rho)$ represents the sample standard deviation across dates. Following standard quantitative conventions, RankICIR is computed on daily observations without annualization. The IC information ratio (ICIR) is computed analogously by replacing ρ_t with IC_t. In Table 2, these metrics evaluate the final composite score z_{it} for all models, including end-to-end neural networks and tree ensembles.

Factor-Level Directional Stability. While the ensemble score z_{it} evaluates combined ranking capacity, the factor sign-flip rate inspects the individual constituent factors within each library to detect distribution shift and historical overfitting. In automated factor discovery, expressions often capture sample-specific patterns that invert their correlation direction when evaluated out of sample.

Let J denote the number of factors in a library, and let $f_j(X_{\le t})_i$ denote the score assigned to asset i by factor j on date t. For dataset partition $q \in \{\mathrm{train}, \mathrm{valid}, \mathrm{test}\}$ containing trading dates $\mathcal{T}_q$ with count T_q, the mean daily correlation of factor j within that partition is

$$\overline{c}_j^{(q)} = \frac{1}{T_q}\sum_{t\in\mathcal{T}_q}\mathrm{Corr}_{i\in\mathcal{U}_t}\left(f_j(X_{\le t})_i, y_{it}^{(h)}\right). \tag{13}$$

The library sign-flip rate measures the percentage of constituent factors that fail to preserve their directional orientation across all three partitions:

$$\mathrm{FlipRate} = \frac{100}{J}\sum_{j=1}^{J}\mathbf{1}\left[\neg\left(\mathrm{sign}(\overline{c}_j^{(\mathrm{train})}) = \mathrm{sign}(\overline{c}_j^{(\mathrm{valid})}) = \mathrm{sign}(\overline{c}_j^{(\mathrm{test})}) \in \{-1, +1\}\right)\right]. \tag{14}$$

Here, $\mathbf{1}[\cdot]$ is the indicator function. Factors with an exact empirical mean of zero in any split lack a definite sign and are categorized as unstable. Lower values reflect superior generalization across market cycles.

In Table 2, the factor count is $J = 50$ for all automated mining algorithms, evaluating their complete frozen libraries without subset selection. End-to-end predictive models (such as LightGBM, GRU, and Transformer) map market features directly to a single prediction score without generating an underlying library of discrete factors, which makes Equation 14 inapplicable to their architectures.

The sign-stability count in Figure 3a uses mean RankIC, whereas the sign-flip rate in Table 2 uses mean Pearson IC.

A.4 Portfolio Metrics

To evaluate downstream investment performance under realistic market conditions, the portfolio simulator defined in Appendix A.2 tracks daily portfolio holdings and transaction costs across the test period.

Return and Wealth Trajectories. Let r_t denote the net daily return of the portfolio on trading day t after deducting all transaction frictions. Let b_t denote the daily return of the benchmark index. The active daily excess return e_t is defined as

$$e_t = r_t - b_t. \tag{15}$$

To track compounding performance over time, the cumulative wealth curve W_t starting from initial capital $W_0 = 1$ is computed as

$$W_t = \prod_{u=1}^{t} (1 + r_u). \tag{16}$$

The running historical peak wealth achieved through date t is defined as

$$P_t = \max\left(\{1, W_1, W_2, \ldots, W_t\}\right). \tag{17}$$

Strategy Performance Statistics. Following standard quantitative conventions, annualized metrics assume 252 trading days per calendar year. Across T evaluation dates, the sample mean daily portfolio return is $\overline{r} = \frac{1}{T}\sum_{t=1}^{T} r_t$, the sample mean daily excess return is $\overline{e} = \frac{1}{T}\sum_{t=1}^{T} e_t$, and the sample standard deviation of daily excess return is

$$s(e) = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}(e_t - \overline{e})^2}. \tag{18}$$

In Table 2, the strategy performance columns report four standard summary metrics:

$$\text{AR} = 252 \cdot \overline{r}, \tag{19}$$

$$\text{AER} = 252 \cdot \overline{e}, \tag{20}$$

$$\text{IR} = \sqrt{252} \cdot \frac{\overline{e}}{s(e)}, \tag{21}$$

$$\text{MDD} = \max_{t \in \{1,\ldots,T\}} \left(1 - \frac{W_t}{P_t}\right). \tag{22}$$

Here, the Annualized Return (AR) and Annualized Excess Return (AER) quantify absolute and benchmark-relative capital gains, respectively. The Information Ratio (IR) measures risk-adjusted excess returns by dividing active return by tracking error volatility, scaled by $\sqrt{252}$ to reflect annual excess Sharpe performance. The Maximum Drawdown (MDD) quantifies the largest peak-to-trough decline in portfolio wealth and is reported as a positive percentage.

All out-of-sample portfolio trajectories are evaluated strictly after the factor library, selection thresholds, and ensemble weights are finalized and frozen, preventing lookahead bias or post-hoc parameter adjustments. To ensure fair algorithmic comparisons across all agent baselines, all LLM-driven discovery systems operate under the identical foundation model backbone (`deepseekv4-flash`) and standardized prompt contexts.

B Additional Experimental Results

This appendix provides additional experimental curves, ablation diagnostics, and structural properties of the discovered factor library.

B.1 Cross-Universe Curves and Evaluation Scope

Figure 4(a) supplements Table 4 by illustrating policy generalizability across the CSI 500 universe. It compares the market index against two portfolios, where one directly transfers the 50 factor formulas mined on CSI 300 without modification or retraining, and the other utilizes factors mined independently on CSI 500. Both curves incorporate the full China-equity transaction fees defined in Appendix A.2, showing that factors discovered on large-cap stocks transfer robustly to smaller-cap assets.

Figure 4(b) evaluates cross-market applicability on the U.S. equity market using factors mined independently on S&P 500 constituents. Following standard conventions, this portfolio is evaluated without transaction fees against the official S&P 500 Price Index, achieving an annualized return of 9.34% over the 2022 through 2025 test period. All wealth trajectories are normalized to 1.0 at the beginning of the test window.

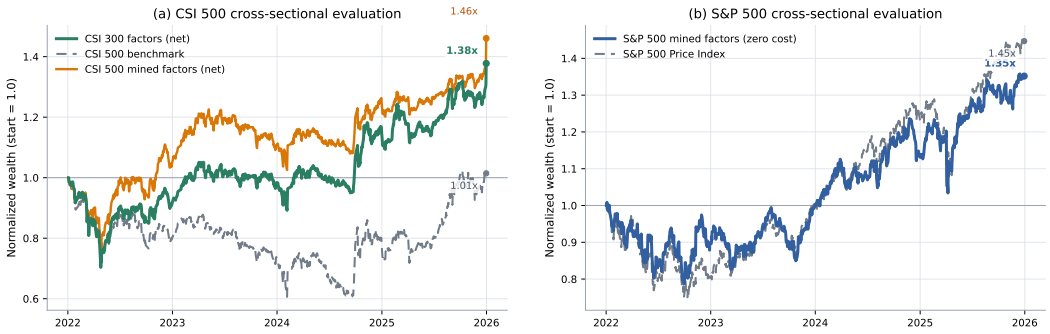

Figure 4: Normalized out-of-sample wealth curves across equity universes. Panel (a) compares the transferred CSI 300 factor library and the independently mined CSI 500 library against the CSI 500 market index under transaction fees. Panel (b) compares the separately mined S&P 500 cross-sectional portfolio against the official S&P 500 Price Index.

B.2 Component Ablation Diagnostics

Figure 5 visualizes the search efficiency and downstream performance across the four ablation variants reported in Table 3. While Active Empirical Grounding (AEG) in isolation improves predictive quality, it increases computational cost due to exploratory data queries. Conversely, groupwise reflection halves token consumption by clustering behaviorally equivalent implementations and eliminating redundant reflection prompts. When combined in GAMMA, the two mechanisms operate synergistically, achieving the highest factor admission rate of 7.69% and evaluating the fewest candidates (653). This combination yields the strongest downstream portfolio performance, delivering 12.31% in net annualized excess return and an Information Ratio of 1.332.

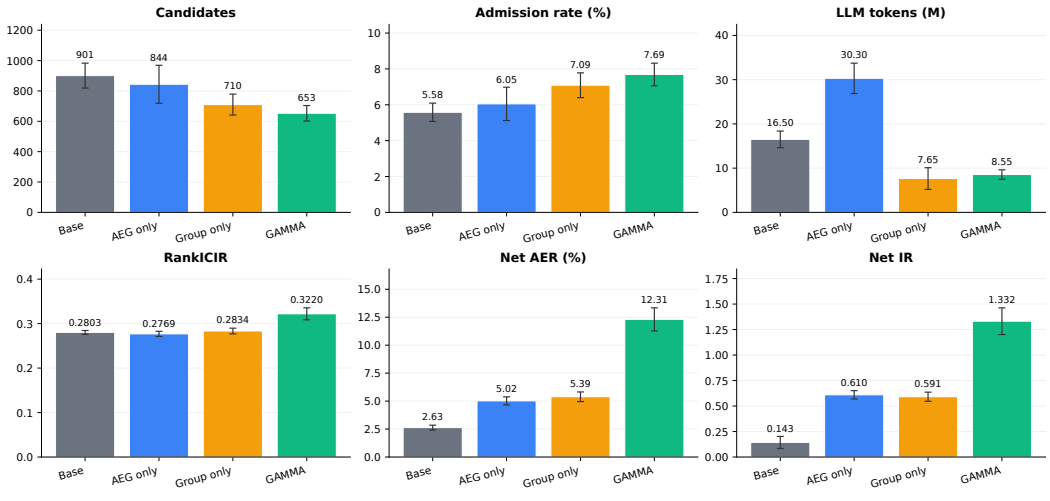

Figure 5: Search efficiency and policy performance across the ablation variants. Lower values are better for candidate counts and token usage. Higher values are better for admission rate, RankICIR, net annualized excess return, and net Information Ratio. Values represent sample means across three independent runs with identical search budgets.

B.3 CSI 300 Annual Performance and Regime Robustness

To evaluate whether the discovered trading policy relies on specific market trends or maintains consistent outperformance across changing environments, Figure 6 provides a calendar-year breakdown of net returns alongside the CSI 300 benchmark. The 2022 through 2025 out-of-sample period covers distinct macroeconomic regimes, shifting from a multi-year bear market to structural market rebounds.

GAMMA achieves positive active excess returns over the benchmark in all four test years after incorporating all transaction costs. During the 2022 and 2023 market downturns, the CSI 300 index suffered steep declines of -21.6% and -11.4%, respectively. In contrast, GAMMA demonstrated downside capital preservation, recording -4.6% in 2022 (a net excess return of $+17.0\%$) and turning positive in 2023 with a net return of $+10.1\%$ (a net excess return of $+21.5\%$). When the market stabilized and rebounded in 2024 and 2025, with benchmark gains of $+14.7\%$ and $+17.7\%$, the policy continued to outpace the index by delivering net returns of $+24.4\%$ and $+21.5\%$, respectively.

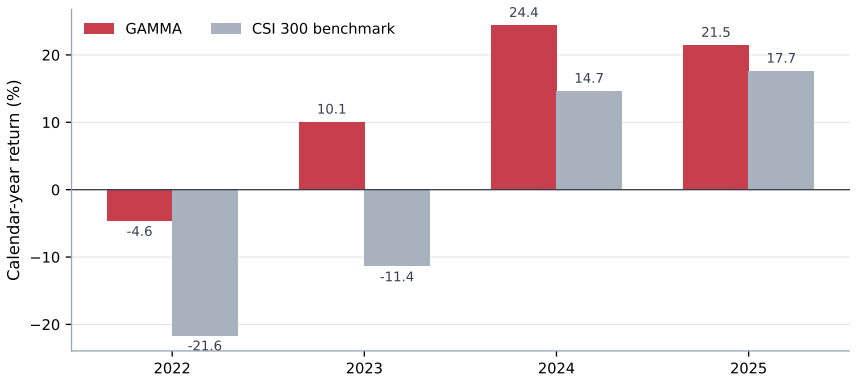

Figure 6: Calendar-year net returns for the main GAMMA CSI 300 strategy and the benchmark index across the 2022 through 2025 test window. All strategy returns include the full China-equity transaction fee and stamp duty schedule.

B.4 Frozen Factor-Library Correlation

Figure 7 illustrates the pairwise Pearson correlation matrix among the 50 factors in the frozen GAMMA library, computed from cross-sectional percentile ranks across validation dates. Across all 1,225 unique off-diagonal factor pairs, the mean and median absolute correlations are 0.157 and 0.117, respectively. The correlation matrix achieves an entropy-based effective rank of 27.3 out of 50. These low correlations indicate limited pairwise redundancy in the frozen factor library. Rather than accumulating repetitive variations of the same pattern, the multi-agent framework constructs a library of mutually complementary features that support stable downstream ensemble learning.

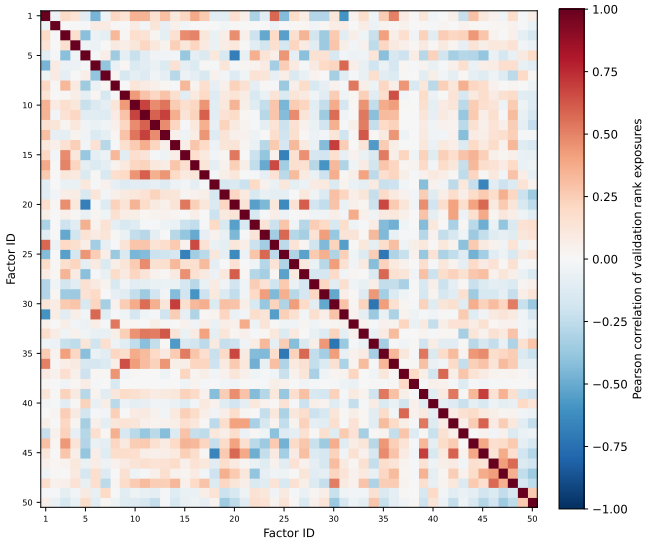

Figure 7: Pairwise correlation matrix for the frozen 50-factor GAMMA library. Each cell displays the Pearson correlation between two factors across validation dates, indexed by registry order. Red and blue denote positive and negative correlation, respectively.

C Case Study

This appendix provides an end-to-end trace of a complete multi-agent discovery cycle from the CSI 300 execution log, followed by an in-depth analysis of a representative discovered factor.

C.1 End-to-End CSI 300 Search Trace

This subsection follows one complete path from the GAMMA CSI 300 search log.

1. Market data to research query. The AEG operation receives only X_{train}. Its internally generated query q asks whether the normalized low-to-VWAP position

$$x_{i,t} = \text{SignedLog1p}\left(\frac{\text{low}_{i,t} - \text{vwap}_{i,t}}{\text{close}_{i,t}}\right) \tag{23}$$

persists across training dates, measured by the daily cross-sectional Pearson correlation between $x_{i,t}$ and $x_{i,t-\ell}$ at positive lags ℓ. The resulting DiscoverySpec is

$$\begin{aligned} d(q) = \{\ & x = \text{SignedLog1p}((\text{low} - \text{vwap})/\text{close}), \\ & \text{kind} = \texttt{temporal}, \quad \text{metric} = \texttt{linear_persistence}, \\ & \text{lags} = \{1, 2, 3, 5, 10\}\ \}. \end{aligned}$$

The deterministic Runner returns the following lag-1 statistics:

$$\begin{aligned} &\text{mean} = 0.36255, && \text{median} = 0.36779, && \text{block range} = [0.32203, 0.39045], \\ &\text{CI}_{95\%} = [0.35342, 0.37238], && n_{\text{dates}} = 2408, && \widetilde{N} = 290. \end{aligned}$$

The reproduction gate passes. GAMMA distills the market experience that normalized low-to-VWAP positions exhibit positive temporal persistence and adds it to the market experience store.

2. Experience and library context to hypothesis. The hypothesis agent sees sampled research and market experiences together with summaries of the incumbent factors. The generated hypothesis explicitly cites research experience and market experience. Compactly, the logged hypothesis is

> *Persistent VWAP-relative anchoring combined with low dispersion of low-to-VWAP positions reflects sustained one-sided order flow and is expected to predict continuation over the subsequent 10 trading days.*

Its structured content is: mechanism—persistent one-sided liquidity absorption; observable signal—the joint level and dispersion of low relative to VWAP; expected direction—continuation in the direction of the anchoring state. This is an economic conjecture inspired by market experience and research experience. The factor library summary is used here to avoid already represented formulas and families.

3. Hypothesis to specifications and DSL expressions. Let $u = \mathrm{low} - \mathrm{vwap}$. The specification agent proposes four measurements, and the factor agent realizes three valid DSL expressions in Table 5. A specification records the intended signal, measurement, normalization, aggregation, and horizon; it is not yet executable. The DSL is the constrained textual realization that the parser accepts.

Table 5: Three implementations in the GAMMA CSI 300 mining trace. Metrics are RankIC/RankICIR. The shorthand u is defined in the text; each displayed formula is otherwise identical to the logged DSL.

Factor	Specification	DSL expression	Metrics and decision
f_1	20-day mean of u normalized by its 20-day standard deviation	$\mathrm{Mean}(u, 20) / \mathrm{Std}(u, 20)$	Train $-.0271/-.2042$; pooled $-.0245/-.1854$; admitted as new
f_2	20-day median of u normalized by its 20-day interquartile range	$\frac{\mathrm{Med}(u, 20)}{\mathrm{Quantile}(u, 20, 0.75) - \mathrm{Quantile}(u, 20, 0.25)}$	Train $-.0167/-.1685$; pooled $-.0140/-.1403$; Gate 2 failure
f_3	20-day mean of u normalized by the 20-day mean absolute deviation from its rolling mean	$\frac{\mathrm{Mean}(u, 20)}{\mathrm{Mean}(\lvert u - \mathrm{Mean}(u, 20)\rvert, 20)}$	Train $-.0270/-.2167$; pooled $-.0266/-.2092$; admitted as new

4. DSL to AST, compilation, and factor values. For f_1, the DSL parser produces the following AST (the parser's wrapper node is shown explicitly):

```
Expression
`-- Divide
    |-- Mean [window=20]
    |   `-- Subtract
    |       |-- Field(low)
    |       `-- Field(vwap)
    `-- Std [window=20]
        `-- Subtract
            |-- Field(low)
            `-- Field(vwap)
```

The tree makes fields, arity, nesting, and windows explicit for static validation. The compiler then emits

```
(Mean(($low - $vwap), 20) / Std(($low - $vwap), 20))
```

and the backend evaluates this expression on X to obtain the stock–date factor panel $f_1(X)$. Thus the actual chain is $H \to s_1 \to e_1^{\mathrm{DSL}} \to T_1 \to$ compiled expression $\to f_1(X)$; the AST is parsed program structure, not another free-form LLM output.

5. Admission, clustering, feedback, and evolution. Individual screening occurs before clustering. Factors f_1 and f_3 enter the factor library; f_2 fails Gate 2. Behavioral clustering then uses all three validation ranking exposures: $\mathcal{C}_0 = \{f_1, f_3\}$ is `Success`, and $\mathcal{C}_1 = \{f_2\}$ is `Failure`. Their reflections are stored as outcome experiences. The successful cluster queues one expanded hypothesis, which is evaluated in a subsequent round. This closes one full research iteration while preserving the distinction between library admission and post-hoc behavioral clustering.

6. Expanded hypothesis and subsequent evaluation. The expanded hypothesis shifts from the level and dispersion of one-sided VWAP displacement to the relative magnitude and timing of two-sided price excursions:

> *Persistent asymmetry between high-to-VWAP and VWAP-to-low spreads, together with a recent extreme excursion on the dominant side, reflects sustained liquidity demand and predicts subsequent price continuation.*

Two implementations measure the spread asymmetry. Let $a = \text{high} - \text{vwap}$ and $b = \text{vwap} - \text{low}$. The resulting expressions are

$$g_1 = \frac{\text{Mean}(a - b, 20)}{\text{Mean}(\text{high} - \text{low}, 20)},$$
$$g_2 = \text{Mean}\left(\text{Log}\left(\frac{\text{Mean}(a, 20)}{\text{Mean}(b, 20)}\right), 20\right).$$

Their pooled RankIC/RankICIR values are $-0.0003/-0.0020$ and $0.0115/0.0795$, respectively. Both fail Gate 2 and form a single failure cluster. Its reflection is stored as outcome experience, and no further expansion is triggered.

C.2 CSI 300 Factor Example

Factor 22 has the largest training-period absolute RankIC in the frozen 50-factor CSI 300 library and is selected for exposition on that basis alone. No validation or test statistic enters this choice. The factor belongs to the `trend_quality` family, with a pooled train–validation RankIC of 0.0535 and RankICIR of 0.3838 over 2,872 trading dates. Table 6 reports the split-specific results.

Expression.

```
(-1 * Corr((Delta($close, 1) / Ref($close, 1)),
(Delta($volume, 1) / Ref($volume, 1)), 60))
```

Connection to the Three-Stage Workflow. The expression measures the negative 60-day correlation between daily relative changes in price and volume. It illustrates how Stage II can instantiate a price–volume hypothesis with an explicit proxy and time scale. Under Stage III, its validation-period ranking exposures determine its behavioral cluster, while predictive screening and nonredundancy checks determine individual admission. A cluster may support further exploration without establishing the parent mechanism: the expression and its predictive statistics alone do not demonstrate institutional liquidity withdrawal or reconstruct the sequence of generated hypotheses.

Implementation Rationale. Dividing each first difference by its lagged level renders both price and volume approximately scale-free and cross-sectionally comparable. The 60-day window captures medium-horizon trend structure, while the leading negative coefficient assigns higher long preferences to stocks with weak or negative volume comovement.

Table 6: CSI 300 Factor 22 record. The factor is selected for exposition by training $|\mathrm{RankIC}|$ only. IC and RankIC are daily cross-sectional correlations; IR columns divide their time-series means by their standard deviations.

Metric	Train	Validation	Test
IC	0.0387	0.0224	0.0197
ICIR	0.3046	0.1393	0.1311
RankIC	0.0572	0.0341	0.0355
RankICIR	0.4322	0.2005	0.2214

D Implementation Details and Reproducibility

D.1 Active Empirical Grounding Protocol and Evidence Gate

Stage I Active Empirical Grounding (AEG) systematically queries the training panel without exposing return labels y_{t+h}. Its sole external input is X_{train}; the research query q and DiscoverySpec are intermediate artifacts created inside the operation. The program rotates across six structure categories: `distribution`, `cross_sectional`, `temporal`, `comovement`, `relational`, and `regime`. For the assigned category, the Researcher formulates q and encodes it as a DiscoverySpec containing the question, its rationale, one to three named backward-looking Qlib expressions, and one schema-constrained measurement. The Researcher does not emit executable Python.

Deterministic Runner and Reproduction Gate. The Runner is a program executor, not an additional LLM agent. It rejects unknown fields, unsupported syntax, forward references, and dimensionally invalid cross-stock price measurements, then loads only the one to three inputs declared by the `DiscoverySpec`. Program-owned routines convert these inputs into daily effect series $\{d_t^{(m)}\}_{m\in\mathcal{M}(q)}$. For series m, let n_m be its finite-date count, $\bar{d}_m$ its full-period mean, b_{mk} its mean in contiguous block k, $c(q)$ the median daily cross-section, and $[L_m, U_m]$ its 95% moving-block-bootstrap confidence interval. Reproduction is defined by

$$\begin{aligned} G_m(q) &= \mathbf{1}[n_m \geq 500]\,\mathbf{1}[c(q) \geq 50] \\ &\qquad \cdot \mathbf{1}\left[\sum_{k=1}^{5} \mathbf{1}\big[\operatorname{sign}(b_{mk}) = \operatorname{sign}(\bar{d}_m)\big] \geq 4\right] \mathbf{1}[L_m > 0 \ \vee \ U_m < 0], \\ V(q) &= \mathbf{1}[\text{execution succeeds}]\,\mathbf{1}\left[\max_{m\in\mathcal{M}(q)} G_m(q) = 1\right]. \end{aligned} \tag{24}$$

The circular bootstrap uses 20-day blocks, 2,000 resamples, and seed zero. Infinite values or a series with no finite observations fail execution; remaining missing dates are excluded before the coverage check. A multi-series specification is reproduced when any declared effect series passes every condition. Only after deterministic reproduction does the Researcher judge the observation's meaning and semantic novelty. A kept finding is written to the active market-experience pool $\mathcal{E}^M$ together with its structured specification and evidence; rejection does not alter the numerical verdict. This division makes the Runner reproducible while retaining the Researcher for question formulation and interpretation.

Interpretation Scope. AEG measurements describe structure in unlabeled training features. Passing the reproduction gate establishes that a specified descriptive pattern is stable under the fixed protocol; it does not establish future-return predictability, prescribe a factor formula or trading action, or authorize the hypothesis agent to copy a measurement as an alpha.

D.2 Factor Construction, DSL Constraints, and Admission Gates

The groupwise protocol requests $K = 4$ diverse specifications per hypothesis. Each specification is required to adopt a distinct design dimension (e.g., different normalization methods, operator structures, or aggregation time scales).

DSL and AST Construction. For each semantic specification s_k, the factor agent emits one expression e_k^{DSL} in the restricted DSL. The DSL is the contract between generative reasoning and deterministic execution: it is compact enough for inspection, but permits only registered fields, constants, arithmetic, and typed operators. A deterministic parser converts the string into $T_k = \mathrm{Parse}(e_k^{\mathrm{DSL}})$; it does not ask the LLM to provide an AST. The compiler traverses T_k, maps its fields and operators to Qlib syntax, and evaluates the compiled expression to obtain $f_k(X)$.

The DSL provides point-in-time daily OHLCV and VWAP series alongside arithmetic, rolling-window, correlation, covariance, decay, and differential operators. Appendix D.4 specifies their semantics and distinguishes time-series ranks from the cross-sectional ranks used for evaluation. Operator arity and argument types are strictly enforced on the AST. Every operator reference is backward-looking; the maximum lookback window is capped at 120 trading days. Direct references to unadjusted absolute price levels are disallowed unless normalized into scale-invariant relative measures (e.g., returns or ratio bands). The AST size is constrained to $C_{\max} = 70$ nodes (with agents prompted to maintain ≤ 35 nodes) to prevent overparameterized, brittle expressions. After these static checks, a semantic checker verifies that e_k^{DSL} implements s_k and that s_k remains faithful to H.

Statistical Screening and Library Admission. Each compiled factor candidate undergoes three sequential verification layers:

1. **Data Quality Gate**: Requires $\geq 50\%$ finite value cross-sectional coverage and a minimum median cross-section of 20 valid assets.
2. **Predictive Significance Gate**: For all benchmarks, candidates must meet the standardized absolute $|\mathrm{RankIC}| \geq 0.02$ and $|RankICIR| \geq 0.10$ both on the training split and on the pooled train–validation split. Sensitivity analyses indicate that factor admission rates and subsequent portfolio metrics remain highly stable across threshold intervals of $|\mathrm{RankIC}| \in [0.02, 0.035]$.
3. **Redundancy Screening**: A statistically qualified factor is compared against the existing library $\mathcal{F}$. It is rejected as redundant if the normalized AST-subtree multiset overlap in Equation 3 reaches 0.90 or its absolute validation cross-sectional rank-exposure correlation with any existing library member reaches 0.70.

D.3 Behavioral Clustering and Multi-Branch Evolution

Candidate factors implementing the same hypothesis H are evaluated on the validation split. To group implementations with similar ranking exposures, GAMMA computes the absolute Pearson correlation matrix of their cross-sectional percentile ranks:

$$S(f_i, f_j) = \left|\mathrm{Corr}\left(\mathrm{vec}(z^{f_i}), \mathrm{vec}(z^{f_j})\right)\right|. \tag{25}$$

The complete-linkage criterion with $\tau_{\mathrm{cluster}} = 0.60$ ensures that every factor pair within a merged cluster exhibits pairwise behavioral similarity at or above the threshold.

Cluster Verdicts and Evolutionary Branching. Each cluster is assigned a unified evaluation verdict:

- `Success`: At least one member factor passes predictive screening and nonredundancy checks to enter $\mathcal{F}$.
- `Duplicate`: No member is admitted, but at least one passes statistical screening and is redundant with existing factors in $\mathcal{F}$.
- `Failure`: No member is admitted or qualifies as a duplicate. Candidates without usable exposures remain singleton failure units.

A reflection agent summarizes the cluster's collective outcome into $\mathcal{E}^O$. Each `Success` cluster produces exactly one refined sub-hypothesis H'_c, refining the measurements, time scales, or conditions suggested by the admitted implementations. This allows a single promising parent hypothesis to branch into multiple complementary evolutionary trajectories while retaining duplicate and failure outcomes as experience without expansion. These outcomes concern the tested implementations and do not reject the parent economic mechanism.

Executable Cluster–Expand–Merge Schedule. For each zero-based search round n, the implementation applies the following deterministic schedule after evaluating the round's candidates:

1. **Cluster**: Candidates with usable validation exposures are grouped by complete linkage using $S(f_i, f_j)$ above and $\tau_{\text{cluster}} = 0.60$; a candidate without a usable exposure remains a singleton failure unit. The program then assigns each cluster a `Success`, `Duplicate`, or `Failure` verdict using the rules above.
2. **Expand**: Every `Success` cluster produces one queued sub-hypothesis from the parent hypothesis, that cluster's admitted factors only, and the current library summary. Its reflection is written separately to outcome experience and can enter later sampled hypothesis contexts. Duplicate and failure clusters update outcome experience but do not expand.
3. **Merge**: If the round contains at least one `Success` cluster and n is odd, the program considers all as-yet-unmerged expression pairs in the full admitted library. Let $p_{ij} = \{\text{expr}(f_i), \text{expr}(f_j)\}$ and define the eligible set $\mathcal{U}$ by the correlation ceiling $\tau_{\text{merge}} = 0.40$. When $\mathcal{U}$ is nonempty, the program selects

$$\begin{aligned} \mathcal{U} &= \{(i,j) : i < j,\ p_{ij} \notin \mathcal{P}_{\text{merged}},\ S(f_i, f_j) \le \tau_{\text{merge}}\}, \\ (i^\star, j^\star) &= \arg\min_{(i,j)\in\mathcal{U}} S(f_i, f_j). \end{aligned} \tag{26}$$

   Here $\mathcal{P}_{\text{merged}}$ stores previously attempted expression pairs, so it remains stable if library IDs are reassigned. If no eligible pair exists, no merge is enqueued. Otherwise, the merger receives the two factor expressions, their associated mechanisms, and the current library summary, and queues at most one composite hypothesis. Thus, every successful cluster can expand, while at most one additional Merge can occur on the same eligible round; the operators are not strictly alternating.

D.4 Operator Reference and Ranking Conventions

The operator registry defines the allowed function names, argument types, and arities. Table 7 summarizes the available families. Here x and y are per-stock time series, N is a positive integer lookback, and $q \in [0, 1]$ is a quantile level. Expressions are composed from declared fields, numeric constants, binary arithmetic, and typed function calls. The compiler maps field names to their Qlib representations and preserves the AST's parentheses.

Table 7: DSL operator families. Rolling functions operate on each stock's historical observations.

Operators	Meaning
`Ref(x,N)`, `Delta(x,N)`	Lagged value and change from that value.
`Mean`, `Sum`, `Std`, `Var`	Rolling mean, sum, standard deviation, and variance of (x, N).
`Max`, `Min`, `Med`, `Mad`	Rolling extrema, median, and mean absolute deviation of (x, N).
`Skew`, `Kurt`	Rolling skewness and kurtosis of (x, N).
`IdxMax`, `IdxMin`	Days since the rolling maximum or minimum of (x, N).
`TsRank(x,N)`, `TsZScore(x,N)`	Per-stock rolling percentile rank and z-score.
`WMA(x,N)`, `EMA(x,N)`	Linearly weighted and exponential moving averages.
`Slope`, `Rsquare`, `Resi`	Slope, R^2, and residual of a rolling time regression on (x, N).
`Corr(x,y,N)`, `Cov(x,y,N)`	Rolling correlation and covariance.
`Quantile(x,N,q)`	Rolling q-quantile.
`Abs`, `Log`, `Sign`, `Sqrt`, `Tanh`, `SignedLog1p`	Unary transformations; signed log is $\text{sign}(x)\log(1+\lvert x\rvert)$.
`Greater(x,y)`, `Less(x,y)`	Elementwise maximum and minimum.
`Power(x,a)`, `Clip(x,l,u)`	Constant power and clipping to constant bounds $l < u$.
$+$, $-$, $\star$, $/$	Binary arithmetic.

`TsRank` ranks observations over time for a single stock; the compiler maps it to Qlib's `Rank`. In contrast, z_t^f in Equation 4 ranks stocks on a common date and is constructed during evaluation. These two ranking operations are not interchangeable. Invalid numeric outputs remain subject to the coverage gate; for `TsZScore`, rolling standard deviations at or below 10^{-6} produce zero.