AUTOSCIENTIST-QUANT: SELF-EVOLVING CODING AGENTS FOR AUTOMATIC RESEARCH IN QUANTITATIVE INVESTMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model agents can discover alphas, yet current methods have three weaknesses. The search cannot adapt during the run, automation usually ends at alpha generation while library selection and model choice stay manual, and alpha discovery can read the test window through loop feedback or code problems. We present AutoScientist-Quant, a self evolving search process that regards quantitative research as one budgeted search problem. A single controller conditions every decision on the remaining budget, choosing at each round whether to improve, combine, pivot, or stop, which node to expand, how many alphas to generate, and how to retrieve past trajectories from the shared memory. The same core then selects from the library and tunes the model, closing the loop from hypothesis to deployable strategy. We also review the evaluation pipeline reused from prior work, fix two lookahead problems, and keep the feedback window disjoint from the held out test window, so every comparison tests true generalization. On CSI universes, the framework attains the best value of nearly every metric in every setting, and these conclusions hold across several backbones and markets.

1 INTRODUCTION

1.1 BACKGROUND & RELATED WORK

Alphas rank stocks by expected future return (Fama & French, 1992) and are the basic input of quantitative investment (Bosworth et al., 1975; Barro, 1990; Demirgüç-Kunt & Levine, 1996; Demirgüç-Kunt & Maksimovic, 1996; Teweles & Bradley, 1998). Because markets are noisy and nonstationary (Fama, 1965; Engle, 1982), individual alphas are not effective once they become crowded, so new alphas need to be discovered at scale and renewed over time. Hand engineered libraries such as the 101 formulaic alphas (Kakushadze, 2016) and ALPHA158 (Yang et al., 2020) encode expert knowledge but grow slowly, while symbolic search based on reinforcement learning (Zhang et al., 2020; Yu et al., 2023) expands the formula space cheaply yet optimizes backtest fit alone, producing complex expressions that overfit and do not last.

Large language models (Vaswani et al., 2017; Minaee et al., 2025; Wan et al., 2024) bring financial priors and reliable code generation to this problem (Wu et al., 2023; Fu, 2025; Li et al., 2026b;c). Agentic frameworks emulate the workflow of a human researcher by proposing a market hypothesis, translating it into an alpha expression and executable code, backtesting it, and revising from feedback (Yao et al., 2023b; Wang et al., 2025; Li et al., 2026a). Within this category, AlphaAgent (Tang et al., 2025) regularizes the loop with originality, hypothesis consistency, and complexity bounds to keep alphas effective longer, and QuantaAlpha (Han et al., 2026) regards every end to end run as a trajectory and evolves trajectories by revising weak parts and recombining strong ones, making refinement traceable and controllable.

These methods automate alpha generation but leave the surrounding research loop fixed. Exploration follows a preset schedule with a constant number of directions, rounds, and alphas, parallel searches evolve in isolation without sharing validated experience, and the resulting library is handed to the downstream model without portfolio level selection. The gap is visible in Table 1, where the agentic baselines reach strong information quality yet negative excess returns after costs.

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.

1.2 MOTIVATIONS

- **Static and incomplete automation in prior frameworks.** Existing designs (Tang et al., 2025; Han et al., 2026; Li et al., 2026a) fix every structural choice, predefining directions, rounds, alphas, and operations before the search starts, yet the question of which directions deserve depth is never answered. Automation also usually ends at alpha generation, handing the library to one fixed predictor with no selection or model search, leaving the model no freedom to correct earlier mistakes or explore new approaches.
- **Train test period overlap in prior agentic pipeline.** Existing agentic pipelines (Tang et al., 2025; Han et al., 2026; Li et al., 2026a) compute loop feedback on the same window used for final evaluation. The separation of training, validation, and test data holds at the model level but not at the search level, since every accepted alpha is selected on test performance, so reported results have a selection effect toward the evaluation period (Bailey et al., 2017; Harvey & Liu, 2015). Overlap free evaluation requires a feedback window disjoint from the held out test window, as in Section 3.1.
- **Test data multiple usage from prior code problem.** The evaluation code reused from prior work (Tang et al., 2025; Han et al., 2026; Li et al., 2026a) contained two lookahead problems. Information metrics were computed over the full sample rather than the declared test segment, so selection feedback incorporated the future evaluation period, and a failed segment filter scored models on their own training span. We fixed both problems, as detailed in Appendix A.6, and reevaluated all methods on the corrected pipeline, so the comparisons test true generalization.

1.3 CONTRIBUTIONS

This paper makes four contributions.

- **Resource aware dynamic search framework.** A single controller regards cost and structure as decisions rather than constants. Every choice conditions on the remaining global budget, and at each round the controller selects among IMPROVE, COMBINE, PIVOT, and STOP, branches from any stored node, decides how to retrieve past trajectories from the shared memory, and sets how many alphas to generate, so the search adapts to evidence while staying within a fixed spend.
- **Closed research loop.** AutoScientist-Quant regards quantitative research as a complete scientific workflow (Lu et al., 2024; Yamada et al., 2025; Lu et al., 2026) rather than a single generation step. Alpha discovery, alpha filtering, and model search share one decision core, every round runs the full cycle of review, ideation, experiment, analysis, and reflection, and each experiment is itself a complete development pass through coding and debugging, training, and backtesting, with model selection and parameter tuning in the model search stage.
- **Trustworthy evaluation infrastructure.** We reviewed the evaluation code reused from prior work, corrected two lookahead problems, and separated the feedback window from the held out test window at the design level, then reevaluated every method on the corrected pipeline. Thus, all reported comparisons rest on an overlap free foundation.
- **Validated alpha library and financial insights.** The discovered alphas earn positive excess returns after costs on the held out test window. Each alpha comes with a readable history, the hypothesis that produced it, the revisions that refined it, and the evidence that kept it in the library. The process yields transferable insights on how alphas are found, how they are improved, and what the selected ones reveal about the market.

2 METHODS

AutoScientist-Quant regards quantitative research as one budgeted search problem solved in three linked stages, overviewed in Figure 1. Alpha discovery generates a candidate alpha library, alpha filtering selects it, and model search tunes the predictor that consumes it. All three stages share one decision core. At every round a controller conditions on the remaining global budget and chooses the option among IMPROVE, COMBINE, PIVOT, and STOP, the parent node to expand, and the number of candidates to generate. The same process is reused on the alpha space, the library subset space, and the model configuration space.

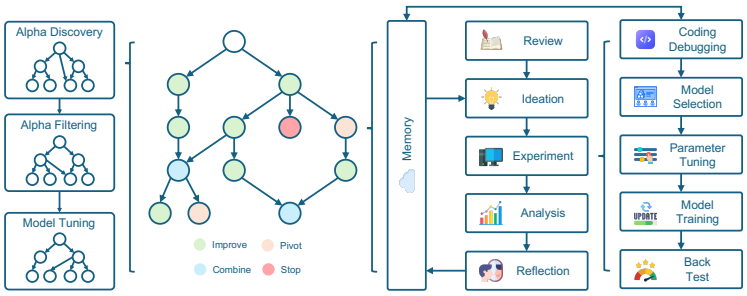

Figure 1: **Overview of AutoScientist-Quant.** Left: the three stages, each a budgeted tree search. Middle: the search, where every node is a round of a trajectory, colors mark the options chosen by the controller, and all trajectories share one memory. Right: the agent cycle inside each round and the pipeline inside each experiment.

2.1 Problem Formulation

Let $\mathbf{X}_{\le t} \in \mathbb{R}^{N\times t\times W}$ collect the history of N stocks up to day t, with W price and volume fields per stock and day. An alpha is a function f mapping this history to cross sectional scores $f(\mathbf{X}_{\le t}) \in \mathbb{R}^N$, a library $\mathcal{F} = \{f_1, \dots, f_K\}$ combines the scores into inputs, and a predictor $g \in \mathcal{G}$ maps inputs to forecasts $\hat{\mathbf{y}}_t = g(\mathcal{F}(\mathbf{X}_{\le t})) \in \mathbb{R}^N$ of the future return defined in Appendix A.3. A fixed portfolio rule turns forecasts into daily positions, and $U(\mathcal{F}, g; \mathcal{D})$ denotes the combined backtest utility on window $\mathcal{D}$, aggregating the metrics of Section 3.2, either through the controller's own reading of these metrics during alpha discovery or through the exact standardized score of Eq. 4 during filtering and model search. The framework solves

$$\max_{\mathcal{F}_B,\ \mathcal{F}\subseteq\mathcal{F}_B,\ g\in\mathcal{G}} U(\mathcal{F}\cup\mathcal{A},\ g;\ \mathcal{D}_{\text{fb}}) \qquad \text{s.t.} \qquad |\mathcal{F}_B| \le B, \tag{1}$$

where $\mathcal{F}_B$ ranges over libraries discoverable under the global budget B, $\mathcal{A}$ is an optional standard library, and $\mathcal{D}_{\text{fb}}$ is the feedback window of Section 3.1. The held out window enters exactly once, to produce the numbers in Section 4.

2.2 Alpha Discovery

Alpha discovery runs trajectories. A trajectory is a sequence $\tau = ((h_1, \mathcal{F}_1, e_1), \dots, (h_R, \mathcal{F}_R, e_R))$ whose round r proposes a market hypothesis h_r, builds and executes the alpha set $\mathcal{F}_r$ implementing it, and records backtest feedback e_r; its reward is $R(\tau) = \max_r V(\mathcal{F}_r)$ with V the feedback window utility. All trajectories from all directions live in a shared pool $\mathcal{P}$, the working memory, visible to every branch. At round r the controller observes the pool and the remaining budget b_r and outputs one decision per active branch d,

$$(o_{d,r}, \sigma_{d,r}, n_{d,r}) = \pi_{\text{LLM}}(\mathcal{P}_r,\ b_r)\,, \qquad o_{d,r} \in \{\textsc{Improve}, \textsc{Combine}, \textsc{Pivot}, \textsc{Stop}\}, \tag{2}$$

where the position $\sigma_{d,r}$ selects parent nodes in $\mathcal{P}_r$ and the cardinality $n_{d,r} \in [n_{\min}, n_{\max}]$ caps how many alphas the round may generate. Improve performs a targeted revision, localizing a step of the parent trajectory by self reflection and rewriting only that part, and Combine recombines complementary parts of stored parents, where the strategy for retrieving parents from the pool is itself decided by the controller at run time; the two options follow Han et al. (2026). Pivot opens a fresh direction from a new market hypothesis when existing directions plateau, and Stop closes a branch whose expected gain no longer justifies its cost. The budget evolves as

$$b_{r+1} = b_r - \sum\nolimits_d n_{d,r}, \qquad \pi_{\text{LLM}}(\cdot \mid b = 0) = \textsc{Stop}, \tag{3}$$

so the controller allocates search capacity where marginal evidence is strongest and stops once it is not obvious.

2.3 Alpha Filtering

Discovery returns a library whose members are individually validated but jointly redundant. The library seeds a subset search guided by the same controller. The state is a split of the library into a

selected pool S and a removed pool; the edits are remove, add, and swap; and IMPROVE, COMBINE, and PIVOT extend, recombine, or restart edit sequences from any stored node. Each candidate subset is scored against the historic pool of nodes,

$$Z(S) = \sum_k \frac{m_k(S) - \mu_k}{\varsigma_k}, \tag{4}$$

where $m_k(S)$ is the value of metric k for node S, and μ_k, ς_k standardize metric k against its historic distribution across previously evaluated nodes. The sum covers the eight metrics of Section 3.2, and all nodes are rescored against the current pool at each comparison, so every ranking uses one set of statistics. Both this stage and the model search stage reuse the budget rule of Eq. 3, with $n_{d,r}$ counting evaluated candidates, subset edits here and model configurations in Section 2.4.

2.4 MODEL SEARCH

The final stage reuses the controller on the model configuration space $\Omega = \bigcup_{m \in \mathcal{M}} \{m\} \times \Theta_m$, where $\mathcal{M}$ contains Linear, LightGBM (Ke et al., 2017), XGBoost (Chen & Guestrin, 2016), CatBoost (Prokhorenkova et al., 2018), and other models and Θ_m its hyperparameter space, so a candidate is a complete algorithm and hyperparameter choice. Candidates branch from any earlier configuration through IMPROVE, COMBINE, and PIVOT and are scored by the standardized utility of Eq. 4 with S replaced by (m, θ_m). The winning pair of library and configuration is trained once on the training and validation windows of Section 3.1, with early stopping on validation, and evaluated once on the held out window, which no stage of the search ever observes.

3 EXPERIMENTAL DESIGN

3.1 DATASETS

We use daily open, high, low, close, and volume data of six universes, **CSI 300**, **CSI 500**, **CSI 800**, **CSI 1000**, **S&P 500**, and **NASDAQ 100**, from January 2015 to May 2026, accessed through Qlib (Yang et al., 2020) with index members taken as of each historical date. The main experiments in Table 1 use CSI 300, the market robustness analysis in Table 4 covers the other three CSI universes, and Table 5 covers the two US universes. The data are split into four continuous windows, training from January 2015 to January 2020, validation from January 2020 to June 2021, a feedback window from June 2021 to June 2023 that supplies backtest feedback to the agents, and a test window from June 2023 to May 2026 reserved for final evaluation. The test window never overlaps any window the agents observe, so information from the evaluation period cannot reach the discovery loop, and Appendix A.3 gives full details. For a stricter evaluation, Table 7 in Appendix A.2 re-evaluates the agentic methods on the portion of the test window that postdates the release of both backbones.

3.2 EVALUATION METRICS

We report four information metrics and four portfolio metrics, with each result averaged over three independent runs, and Table 6 in Appendix A.1 reports the run to run variability and statistical significance. The information coefficient **IC** is the Pearson correlation between predicted scores and realized labels within each daily cross section, **ICIR** is the mean daily IC over its standard deviation, and **RIC** (Rank IC) and **RICIR** (Rank ICIR) are the Spearman counterparts. Portfolio metrics come from a cost adjusted daily strategy that holds the 50 stocks with the highest scores, replaces at most 5 of them per day, and is benchmarked against the index of each universe. **ARR** is the annualized excess return over the benchmark index net of transaction costs, **IR** is the information ratio of daily excess returns, **MDD** is the maximum decrease of the cumulative excess return, and **CR** is the Calmar ratio equal to ARR divided by the absolute value of MDD. On the US universes these metrics are computed on absolute returns without subtracting the index return, as noted in Table 5. Appendix A.4 gives full details.

3.3 BASELINES

Baselines span three categories over three standard Qlib alpha libraries, ALPHA158, ALPHA360, and ALPHA20 (Yang et al., 2020). Machine learning baselines include **Linear**, **LightGBM** (Ke

Method	Alpha	IC	ICIR	RIC	RICIR	ARR	IR	MDD	CR
				Machine Learning Methods					
LightGBM	ALPHA20	-0.003	-0.019	0.014	0.079	-1.0%	-0.100	-24.9%	-0.039
	ALPHA158	0.010	0.063	0.029	0.176	-0.1%	-0.007	-13.2%	-0.004
	ALPHA360	0.006	0.036	0.025	0.147	-0.1%	-0.007	-17.3%	-0.003
Linear	ALPHA20	-0.006	-0.033	0.010	0.051	-7.1%	-0.686	-31.6%	-0.224
	ALPHA158	0.005	0.025	0.020	0.102	-5.1%	-0.562	-20.1%	-0.252
	ALPHA360	-0.002	-0.013	0.014	0.072	-6.5%	-0.665	-23.9%	-0.272
CatBoost	ALPHA20	-0.003	-0.013	0.012	0.062	-2.7%	-0.259	-26.3%	-0.102
	ALPHA158	0.009	0.050	0.027	0.157	-2.1%	-0.275	-13.7%	-0.153
	ALPHA360	0.008	0.043	0.025	0.141	-0.0%	-0.004	-16.1%	-0.002
				Deep Learning Methods					
LSTM	ALPHA20	-0.009	-0.042	0.008	0.035	-9.9%	-0.916	-32.2%	-0.307
	ALPHA158	0.006	0.043	0.017	0.151	0.5%	0.092	-7.5%	0.072
	ALPHA360	-0.000	-0.001	0.021	0.110	-4.5%	-0.552	-25.3%	-0.177
TCN	ALPHA20	-0.006	-0.033	0.007	0.035	-3.7%	-0.347	-30.8%	-0.121
	ALPHA158	0.007	0.039	0.026	0.152	-1.5%	-0.190	-12.9%	-0.114
	ALPHA360	-0.000	-0.002	0.013	0.068	-0.3%	-0.027	-20.4%	-0.013
KRNN	ALPHA20	0.001	0.008	0.014	0.097	-2.5%	-0.280	-20.3%	-0.123
	ALPHA158	0.003	0.025	0.018	0.150	-4.2%	-0.723	-14.0%	-0.296
	ALPHA360	0.003	0.017	0.025	0.152	-4.8%	-0.673	-21.6%	-0.221
				Agentic Methods					
AlphaAgent	CUSTOM	0.024	0.147	0.022	0.135	-2.3%	-0.321	-15.1%	-0.153
	+ALPHA20	0.023	0.135	0.020	0.121	-1.0%	-0.130	-17.0%	-0.055
	+ALPHA158	0.030	0.182	0.027	0.168	-1.0%	-0.151	-12.5%	-0.087
	+ALPHA360	0.027	0.164	0.026	0.155	-1.5%	-0.202	-15.5%	-0.098
QuantaAlpha	CUSTOM	0.024	0.144	0.022	0.130	-1.6%	-0.207	-14.5%	-0.096
	+ALPHA20	0.023	0.139	0.021	0.125	-0.5%	-0.060	-15.6%	-0.033
	+ALPHA158	0.031	0.190	0.029	0.177	-0.2%	-0.018	-12.5%	-0.013
	+ALPHA360	0.029	0.173	0.027	0.160	-1.5%	-0.189	-13.3%	-0.098
AutoScientist -Quant	CUSTOM	**0.028**	**0.185**	**0.026**	**0.172**	**1.8%**	**0.259**	**-12.7%**	**0.143**
	+ALPHA20	**0.028**	**0.181**	**0.026**	**0.165**	**2.4%**	**0.330**	**-12.5%**	**0.209**
	+ALPHA158	**0.034**	**0.219**	**0.032**	**0.206**	**3.5%**	**0.500**	**-9.5%**	**0.368**
	+ALPHA360	**0.033**	**0.212**	**0.031**	**0.198**	**3.0%**	**0.401**	**-12.0%**	**0.263**

Table 1: **Main results on CSI 300.** Machine learning, deep learning, and agentic methods across four alpha settings, each library used alone and merged with a standard library. Agentic methods use the GPT backbone. ARR is the annualized excess return over the CSI 300 index, net of transaction costs. AutoScientist-Quant reports the full three stage framework, and the ablation study in Table 2 shows that alpha discovery alone, without alpha filtering or model search, still outperforms every baseline on almost every metric.

et al., 2017), and **CatBoost** (Prokhorenkova et al., 2018), and deep learning baselines include **LSTM** (Hochreiter & Schmidhuber, 1997), **TCN** (Bai et al., 2018), and **KRNN** (Chattha et al., 2025). The agentic baselines **AlphaAgent** (Tang et al., 2025) and **QuantaAlpha** (Han et al., 2026) of Section 1.1 share our backbone, search budget, and backtest procedure. R&D-Agent-Quant (Li et al., 2026a) is excluded since its loop selects alphas on the window it reports, the overlap of Section 1.2. Every resulting library is backtested with a fixed LightGBM predictor before the model search stage, evaluated alone as CUSTOM, and merged with each standard library as the +ALPHA settings, and Appendix A.5 gives full details.

3.4 Models

All agentic methods use the same large language model backbone within each comparison. We run two backbones, **GPT-OSS-120B** (OpenAI et al., 2025) and **GLM-5** (GLM-5-Team, 2026), referred to as GPT and GLM in the tables. The main results and ablations in Tables 1 and 2 use GPT, the model robustness analysis in Table 3 repeats the comparison under GLM, and the market robustness analysis in Table 4 runs both backbones on three further universes, so every conclusion is checked against a change of backbone. The framework itself is model agnostic and also supports other backbones.

4 Results

4.1 Main Results

Table 1 compares the three categories on CSI 300 under the GPT backbone.

Alpha	IC	ICIR	RIC	RICIR	ARR	IR	MDD	CR
AutoScientist-Quant (w/o Model Search) = Alpha Discovery + Alpha Filtering								
CUSTOM	0.026 $^{+0.002}_{-0.002}$	0.155 $^{+0.008}_{-0.030}$	0.025 $^{+0.003}_{-0.002}$	0.148 $^{+0.014}_{-0.024}$	0.9% $^{+2.5\%}_{-0.9\%}$	0.130 $^{+0.337}_{-0.129}$	-13.3% $^{+1.2\%}_{-0.6\%}$	0.066 $^{+0.162}_{-0.077}$
+ALPHA20	0.026 $^{+0.003}_{-0.002}$	0.156 $^{+0.017}_{-0.025}$	0.025 $^{+0.004}_{-0.001}$	0.148 $^{+0.024}_{-0.016}$	2.1% $^{+2.6\%}_{-0.3\%}$	0.295 $^{+0.354}_{-0.035}$	-13.6% $^{+2.0\%}_{-1.2\%}$	0.194 $^{+0.226}_{-0.016}$
+ALPHA158	0.033 $^{+0.002}_{-0.001}$	0.198 $^{+0.008}_{-0.021}$	0.031 $^{+0.002}_{-0.001}$	0.189 $^{+0.012}_{-0.017}$	1.5% $^{+1.7\%}_{-2.0\%}$	0.221 $^{+0.239}_{-0.279}$	-9.9% $^{+2.6\%}_{-0.4\%}$	0.167 $^{+0.179}_{-0.201}$
+ALPHA360	0.032 $^{+0.003}_{-0.001}$	0.187 $^{+0.014}_{-0.025}$	0.030 $^{+0.004}_{-0.001}$	0.179 $^{+0.020}_{-0.019}$	0.8% $^{+2.3\%}_{-2.2\%}$	0.125 $^{+0.315}_{-0.276}$	-12.9% $^{+0.5\%}_{-0.9\%}$	0.087 $^{+0.185}_{-0.176}$
AutoScientist-Quant (w/o Alpha Filtering or Model Search) = Alpha Discovery								
CUSTOM	0.026 $^{+0.001}_{-0.003}$	0.153 $^{+0.005}_{-0.033}$	0.024 $^{+0.003}_{-0.002}$	0.147 $^{+0.012}_{-0.025}$	-0.3% $^{+1.3\%}_{-2.1\%}$	-0.022 $^{+0.184}_{-0.281}$	-15.1% $^{-0.6\%}_{-2.4\%}$	0.000 $^{+0.096}_{-0.143}$
+ALPHA20	0.026 $^{+0.002}_{-0.003}$	0.152 $^{+0.013}_{-0.029}$	0.024 $^{+0.003}_{-0.002}$	0.145 $^{+0.020}_{-0.020}$	1.2% $^{+1.7\%}_{-1.2\%}$	0.174 $^{+0.233}_{-0.156}$	-14.2% $^{+1.4\%}_{-1.8\%}$	0.127 $^{+0.160}_{-0.082}$
+ALPHA158	0.033 $^{+0.001}_{-0.002}$	0.195 $^{+0.005}_{-0.025}$	0.031 $^{+0.002}_{-0.001}$	0.185 $^{+0.008}_{-0.021}$	0.6% $^{+0.7\%}_{-3.0\%}$	0.079 $^{+0.097}_{-0.422}$	-10.0% $^{+2.5\%}_{-0.5\%}$	0.064 $^{+0.077}_{-0.304}$
+ALPHA360	0.031 $^{+0.002}_{-0.002}$	0.182 $^{+0.009}_{-0.030}$	0.029 $^{+0.003}_{-0.002}$	0.174 $^{+0.014}_{-0.024}$	-0.1% $^{+1.3\%}_{-3.1\%}$	0.005 $^{+0.194}_{-0.397}$	-14.2% $^{-0.9\%}_{-2.2\%}$	0.040 $^{+0.138}_{-0.223}$
Alpha Discovery (w/o Searching Strategy)								
CUSTOM	0.025 $_{-0.003}$	0.145 $_{-0.040}$	0.023 $_{-0.004}$	0.133 $_{-0.040}$	-2.4% $_{-4.1\%}$	-0.291 $_{-0.551}$	-16.5% $_{-3.8\%}$	-0.126 $_{-0.269}$
+ALPHA20	0.024 $_{-0.005}$	0.138 $_{-0.043}$	0.021 $_{-0.005}$	0.125 $_{-0.039}$	-0.7% $_{-3.1\%}$	-0.086 $_{-0.416}$	-15.0% $_{-2.6\%}$	-0.034 $_{-0.244}$
+ALPHA158	0.031 $_{-0.003}$	0.189 $_{-0.030}$	0.029 $_{-0.003}$	0.178 $_{-0.028}$	-0.1% $_{-3.7\%}$	-0.022 $_{-0.522}$	-12.0% $_{-2.5\%}$	-0.008 $_{-0.375}$
+ALPHA360	0.030 $_{-0.003}$	0.182 $_{-0.030}$	0.028 $_{-0.003}$	0.173 $_{-0.026}$	-0.4% $_{-3.4\%}$	-0.051 $_{-0.452}$	-13.5% $_{-1.6\%}$	-0.030 $_{-0.293}$
Alpha Discovery (w/o Dynamics)								
CUSTOM	0.024 $_{-0.004}$	0.142 $_{-0.043}$	0.022 $_{-0.004}$	0.132 $_{-0.040}$	-2.9% $_{-4.6\%}$	-0.389 $_{-0.648}$	-16.9% $_{-4.2\%}$	-0.152 $_{-0.295}$
+ALPHA20	0.025 $_{-0.003}$	0.153 $_{-0.028}$	0.023 $_{-0.002}$	0.143 $_{-0.021}$	-2.6% $_{-5.0\%}$	-0.363 $_{-0.693}$	-14.4% $_{-1.9\%}$	-0.168 $_{-0.378}$
+ALPHA158	0.032 $_{-0.002}$	0.189 $_{-0.030}$	0.029 $_{-0.002}$	0.177 $_{-0.030}$	0.5% $_{-3.0\%}$	0.078 $_{-0.422}$	-11.2% $_{-1.7\%}$	0.045 $_{-0.323}$
+ALPHA360	0.028 $_{-0.005}$	0.170 $_{-0.042}$	0.026 $_{-0.005}$	0.159 $_{-0.039}$	-1.2% $_{-4.2\%}$	-0.167 $_{-0.569}$	-13.2% $_{-1.3\%}$	-0.092 $_{-0.355}$

Table 2: **Ablation of AutoScientist-Quant on CSI 300 under GPT.** The center value in each cell is the absolute metric, the top right corner, shown for the first two parts, is the gain over the best agentic value per metric, and the bottom right corner is the gap to our full method, both computed from unrounded values. Green marks positive values and red negative.

Agentic generation creates alpha beyond fixed input sets. The best machine learning or deep learning result reaches an IC of 0.010, obtained by LightGBM on ALPHA158, and most of these strategies lose to the index. All three agentic methods lift IC to between 0.023 and 0.034, roughly three times the best static result, so hypothesis guided discovery adds information beyond predefined inputs.

AutoScientist-Quant is the strongest agentic method in every setting. It attains the best value of all eight metrics in all four alpha settings among agentic methods. With ALPHA158 it reaches an IC of 0.034 and an ICIR of 0.219, against 0.031 and 0.190 for QuantaAlpha and 0.030 and 0.182 for AlphaAgent, and the same ordering holds for the rank based metrics.

AutoScientist-Quant turns alpha into strong portfolio value. Its excess return is positive in all four settings, rising from 1.8% with discovered alphas alone to 3.5% with ALPHA158, where the information ratio reaches 0.500 and the Calmar ratio 0.368. QuantaAlpha with ALPHA158 pairs an IC of 0.031 with an ARR of -0.2%, and among agentic methods AutoScientist-Quant also keeps the smallest MDD in every setting. This is the gap anticipated in Section 1.2. Information metrics alone do not price turnover, costs, and tail behavior.

Generated and predefined alphas are complements. For every agentic method the merged settings outperform its own library alone, with ALPHA158 the strongest partner for all three methods, and AutoScientist-Quant rises from an IC of 0.028 and an ARR of 1.8% with CUSTOM to 0.034 and 3.5% with ALPHA158. The complement runs both ways, since even our CUSTOM setting, which uses no predefined alphas, earns more than any baseline setting that includes them.

4.2 Ablation Studies

Table 2 removes the stages of AutoScientist-Quant one at a time on CSI 300 under GPT. The first part keeps alpha discovery and alpha filtering and removes model search, the second part keeps alpha discovery alone, and the last two parts remove single components inside alpha discovery. The searching strategy denotes the Improve, Combine, Pivot, and Stop options of Section 2.2, and the dynamics denotes the per round decisions on alpha count, early stopping, and direction creation. A removed component follows the setting of the agentic baselines.

The downstream stages turn alpha into returns. Removing model search leaves IC nearly unchanged yet costs up to 2.2 percentage points of excess return, and removing alpha filtering as well pushes the CUSTOM setting from an ARR of 0.9% to -0.3% while IC stays within 0.003 of the full

Alpha	IC	ICIR	RIC	RICIR	ARR	IR	MDD	CR
AlphaAgent								
CUSTOM	0.026 -0.007	0.148 -0.052	0.024 -0.007	0.139 -0.053	-3.1% -3.7%	-0.463 -0.554	-15.3% -6.7%	-0.191 -0.262
+ALPHA20	0.026 -0.007	0.156 -0.054	0.024 -0.007	0.147 -0.055	-1.1% -2.2%	-0.159 -0.319	-14.5% -4.6%	-0.034 -0.175
+ALPHA158	0.032 -0.004	0.199 -0.025	0.030 -0.004	0.187 -0.030	0.9% -1.6%	0.141 -0.243	-11.0% -2.4%	0.116 -0.173
+ALPHA360	0.028 -0.007	0.168 -0.045	0.026 -0.007	0.156 -0.049	-3.3% -4.4%	-0.452 -0.622	-16.2% -6.6%	-0.190 -0.311
QuantaAlpha								
CUSTOM	0.028 -0.005	0.174 -0.026	0.026 -0.005	0.165 -0.027	-3.6% -4.2%	-0.566 -0.656	-15.1% -6.5%	-0.222 -0.294
+ALPHA20	0.026 -0.007	0.159 -0.050	0.024 -0.007	0.150 -0.052	-2.0% -3.0%	-0.307 -0.467	-15.6% -5.7%	-0.061 -0.203
+ALPHA158	0.032 -0.004	0.198 -0.026	0.030 -0.005	0.186 -0.030	-1.2% -3.8%	-0.178 -0.562	-11.7% -3.1%	-0.095 -0.385
+ALPHA360	0.029 -0.006	0.175 -0.037	0.027 -0.006	0.164 -0.041	-2.2% -3.3%	-0.297 -0.467	-13.9% -4.2%	-0.155 -0.276
AutoScientist-Quant (w/o Alpha Filtering or Model Search) = Alpha Discovery								
CUSTOM	0.031 +0.002 -0.002	0.184 +0.010 -0.016	0.029 +0.003 -0.002	0.177 +0.013 -0.015	-0.4% +2.7% -1.0%	-0.064 +0.399 -0.155	-10.1% +5.0% -1.5%	-0.038 +0.153 -0.109
+ALPHA20	0.032 +0.005 -0.002	0.193 +0.033 -0.017	0.030 +0.006 -0.001	0.185 +0.035 -0.017	0.1% +1.2% -1.0%	0.006 +0.166 -0.154	-10.3% +4.2% -0.4%	0.064 +0.098 -0.077
+ALPHA158	0.034 +0.002 -0.002	0.206 +0.008 -0.018	0.032 +0.002 -0.002	0.196 +0.009 -0.021	1.4% +0.5% -1.2%	0.202 +0.061 -0.182	-9.5% +1.6% -0.8%	0.140 +0.024 -0.149
+ALPHA360	0.033 +0.004 -0.002	0.196 +0.021 -0.016	0.031 +0.004 -0.002	0.188 +0.023 -0.018	0.3% +2.4% -0.8%	0.044 +0.341 -0.126	-10.1% +3.8% -0.5%	0.032 +0.187 -0.089
AutoScientist-Quant = Alpha Discovery + Alpha Filtering + Model Search								
CUSTOM	0.033 +0.005	0.200 +0.026	0.031 +0.005	0.192 +0.027	0.6% +3.7%	0.091 +0.554	-8.6% +6.5%	0.071 +0.262
+ALPHA20	0.033 +0.007	0.209 +0.050	0.031 +0.007	0.202 +0.052	1.1% +2.2%	0.160 +0.319	-9.9% +4.6%	0.142 +0.175
+ALPHA158	0.036 +0.004	0.224 +0.025	0.035 +0.004	0.216 +0.030	2.5% +1.6%	0.384 +0.243	-8.6% +2.4%	0.289 +0.173
+ALPHA360	0.035 +0.006	0.212 +0.037	0.033 +0.006	0.205 +0.041	1.1% +3.3%	0.170 +0.467	-9.6% +4.2%	0.121 +0.276

Table 3: **Model robustness on CSI 300 under GLM.** The center value in each cell is the absolute metric, the top right corner is the gain over the best baseline value per metric, and the bottom right corner is the gap to our full method, both computed from unrounded values. Green marks positive values and red negative.

method. The two stages barely touch the information metrics and act almost entirely on portfolio outcomes.

The discovery components create the alpha itself. Removing the searching strategy lowers IC by up to 0.002 relative to the discovery only variant and decreases the CUSTOM excess return from -0.3% to -2.4%, and removing the dynamics costs up to 3.8 percentage points of excess return against the same control, the largest single decrease in the study. Unlike the downstream stages, these components influence information and portfolio metrics at once, so exploration quality cannot be repaired later in the pipeline.

No single component explains the advantage. The two partial methods in the first two parts still improve on the best agentic baseline in nearly every cell, with excess returns up to 2.6 percentage points higher, so the lead of AutoScientist-Quant is the sum of stage level and component level contributions rather than the effect of one source.

4.3 Model Robustness Analysis

Table 3 repeats the comparison on CSI 300 with the GLM backbone, covering the two agentic baselines, the discovery only variant, and the full method.

The ranking holds up across the backbone change. The full method again delivers the best information quality and the only consistent profits. IC reaches 0.036 with ALPHA158, above 0.032 for both baselines, ARR stays positive in all four settings, and seven of the eight baseline settings lose money after costs.

The stage contributions are backbone independent. Under GLM the full method outperforms the discovery only variant by 0.8 to 1.2 percentage points of excess return with a shallower MDD in every setting, matching the ablation trend found under GPT. GLM also lifts the absolute IC of nearly every method and setting, so the backbone sets the level of information quality while the framework sets the ordering.

4.4 Market Robustness Analysis

Table 4 evaluates the three agentic methods on CSI 500, CSI 800, and CSI 1000 under both backbones, and Table 5 extends the comparison to S&P 500 and NASDAQ 100 under the GLM backbone. Alphas are discovered within each universe rather than being transferred, so the analysis tests whether the framework itself, not one particular alpha set, generalizes.

Backbone	Method	IC	ICIR	RIC	RICIR	ARR	IR	MDD	CR
				CSI 500					
GPT	AlphaAgent	0.047	0.312	0.044	0.290	-0.7%	-0.076	-19.4%	-0.035
	QuantaAlpha	0.047	0.322	0.043	0.299	0.2%	0.022	-17.7%	0.009
	AutoScientist-Quant	**0.049**	**0.328**	**0.045**	**0.303**	**0.5%**	**0.062**	**-17.2%**	**0.032**
GLM	AlphaAgent	0.047	0.328	0.043	0.302	-0.1%	-0.016	-17.3%	-0.008
	QuantaAlpha	**0.048**	0.331	0.044	0.305	-1.3%	-0.143	-16.7%	-0.077
	AutoScientist-Quant	0.048	**0.341**	**0.044**	**0.315**	**-0.1%**	**-0.014**	**-15.2%**	**-0.007**
				CSI 800					
GPT	AlphaAgent	0.043	0.297	0.039	0.275	0.8%	0.088	-19.3%	0.063
	QuantaAlpha	0.043	0.313	0.040	0.291	0.2%	0.023	-19.1%	0.012
	AutoScientist-Quant	**0.045**	**0.318**	**0.041**	**0.298**	**2.4%**	**0.261**	**-17.9%**	**0.135**
GLM	AlphaAgent	0.044	0.309	0.040	0.289	-0.2%	-0.018	-19.8%	-0.007
	QuantaAlpha	0.044	0.307	0.041	0.285	0.9%	0.094	-20.2%	0.055
	AutoScientist-Quant	**0.046**	**0.317**	**0.042**	**0.292**	**1.3%**	**0.133**	**-18.5%**	**0.074**
				CSI 1000					
GPT	AlphaAgent	0.065	0.469	0.058	0.417	6.4%	0.654	-15.2%	0.429
	QuantaAlpha	**0.067**	**0.485**	0.060	0.431	4.8%	0.489	-13.0%	0.371
	AutoScientist-Quant	0.066	0.481	**0.060**	**0.437**	**7.1%**	**0.735**	**-12.9%**	**0.591**
GLM	AlphaAgent	0.065	0.486	0.058	0.437	6.5%	0.687	-13.3%	0.533
	QuantaAlpha	0.066	0.485	0.058	0.427	6.4%	0.671	-13.5%	0.471
	AutoScientist-Quant	**0.068**	**0.496**	**0.061**	**0.444**	**6.9%**	**0.711**	**-11.6%**	**0.600**

Table 4: **Market robustness on CSI 500, CSI 800, and CSI 1000.** The three agentic methods run under both the GPT and the GLM backbone, with alphas discovered within each universe. **Bold** marks the best method in each setting, based on unrounded original values, so equal displayed values can differ in bolding.

Method	Alpha	IC	ICIR	RIC	RICIR	ARR	IR	MDD	CR
				S&P 500					
AlphaAgent	CUSTOM	0.006	0.043	0.005	0.040	10.4%	0.687	-21.1%	0.476
	+ALPHA20	0.005	0.035	0.005	0.036	9.8%	0.632	-24.2%	0.399
	+ALPHA158	0.004	0.028	0.002	0.019	12.4%	0.860	-20.5%	0.606
	+ALPHA360	0.005	0.035	0.004	0.030	10.3%	0.674	-26.5%	0.386
QuantaAlpha	CUSTOM	0.007	0.051	0.007	0.053	12.3%	0.816	-22.5%	0.538
	+ALPHA20	0.006	0.043	0.006	0.046	13.0%	0.857	**-21.4%**	0.615
	+ALPHA158	0.005	0.037	0.004	0.033	10.7%	0.758	-21.7%	0.494
	+ALPHA360	0.006	0.043	0.006	0.041	10.7%	0.714	-26.4%	0.405
AutoScientist-Quant	CUSTOM	**0.009**	**0.073**	**0.009**	**0.071**	**15.4%**	**1.021**	**-19.8%**	**0.795**
	+ALPHA20	**0.011**	**0.080**	**0.010**	**0.073**	**15.2%**	**0.982**	-22.3%	**0.681**
	+ALPHA158	**0.007**	**0.053**	**0.006**	**0.048**	**12.8%**	**0.905**	**-19.8%**	**0.648**
	+ALPHA360	**0.008**	**0.059**	**0.007**	**0.051**	**11.5%**	**0.759**	**-25.3%**	**0.453**
				NASDAQ 100					
AlphaAgent	CUSTOM	0.009	0.049	0.010	0.051	20.1%	1.083	**-20.9%**	0.970
	+ALPHA20	0.006	0.032	0.006	0.036	13.7%	0.781	-20.8%	0.695
	+ALPHA158	0.003	0.014	0.002	0.012	12.1%	0.749	-19.6%	0.609
	+ALPHA360	0.003	0.017	0.004	0.022	13.1%	0.799	-19.7%	0.662
QuantaAlpha	CUSTOM	0.010	0.053	0.010	0.052	22.3%	1.041	-25.0%	0.877
	+ALPHA20	0.001	0.005	0.002	0.012	17.8%	1.028	-18.6%	0.953
	+ALPHA158	0.001	0.004	0.000	0.002	13.3%	0.883	**-17.9%**	0.749
	+ALPHA360	0.000	0.001	0.000	0.002	12.6%	0.818	-19.2%	0.648
AutoScientist-Quant	CUSTOM	**0.018**	**0.091**	**0.018**	**0.091**	**28.9%**	**1.297**	-22.5%	**1.289**
	+ALPHA20	**0.011**	**0.059**	**0.011**	**0.060**	**24.6%**	**1.280**	**-18.5%**	**1.323**
	+ALPHA158	**0.007**	**0.034**	**0.007**	**0.035**	**18.3%**	**1.053**	-18.7%	**1.010**
	+ALPHA360	**0.006**	**0.032**	**0.007**	**0.035**	**16.3%**	**0.949**	**-19.0%**	**0.887**

Table 5: **Market robustness on S&P 500 and NASDAQ 100 under GLM.** Each cell is the absolute metric computed without subtracting the index return, since US equity strategies are commonly evaluated this way (Sharpe et al., 1998) rather than by the index excess used on the CSI universes. **Bold** marks the best method in each setting, as in Table 4.

The advantage holds in every market and under both backbones. AutoScientist-Quant delivers the best or equal best excess return in all six settings and the best value on nearly all other metrics. The exceptions are individual information metrics in two settings, such as the IC of QuantaAlpha on CSI 1000 under GPT at 0.067 against our 0.066, where our portfolio metrics remain the strongest.

Gains concentrate where pricing is least efficient. For all methods IC rises from between 0.043 and 0.049 on CSI 500 and CSI 800 to between 0.065 and 0.068 on CSI 1000, and our excess return climbs from 0.5% on CSI 500 to 7.1% on CSI 1000 under GPT and from -0.1% to 6.9% under GLM. This matches the common view that small caps have more remaining alpha.

The framework holds up in hard markets and pushes in easy ones. On the efficiently priced large and mid cap universes the baselines often lose after costs, such as AlphaAgent at -0.7% on

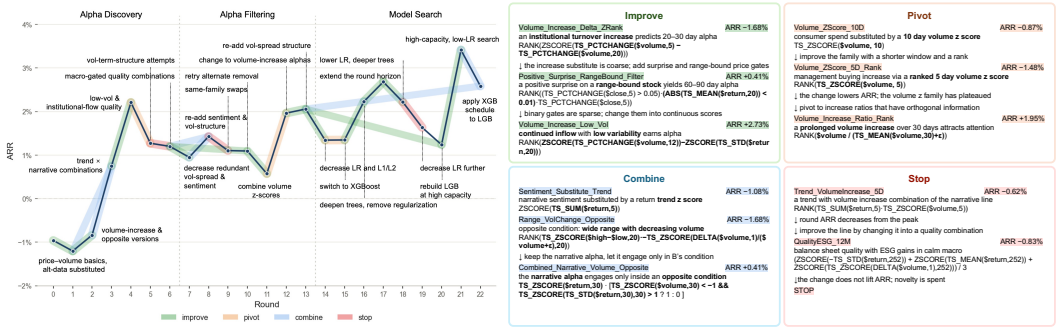

Figure 2: **Example of one search.** Left: test window ARR of the +ALPHA20 deployment after each round of the CSI 300 search under GPT; the search itself sees only the feedback window. Colors mark the option of each move. Right: one example per option with its alpha names, ARR, and the reasoning and reflection of the model.

CSI 500 under GPT, while AutoScientist-Quant stays ahead with the shallowest MDD. On CSI 1000 it extends the best baseline result from 6.4% to 7.1% under GPT and reaches a CR of 0.600 under GLM.

4.5 Case Study

Figure 2 traces the searches behind Table 1; four behaviors stand out.

Test performance recovers from dips the search cannot see. Filtering touches its stage low one move before the swap that lifts the subset to its stage best, and model search leaves an XGBoost branch, restarts from the root, loses further ground, and reaches the stage best on the next move. The pool of stored parents lets the search return to a promising branch after a period of losses. The curve reports the test window while every move is chosen on the feedback window alone, so these dips and recoveries are invisible to the search and appear only in hindsight.

A few redundant members move ARR more than library size. The selected subset keeps 89 of 91 alphas yet single swaps move test ARR by more than a percentage point. Adding two same family volume z score alphas brings the subset to its stage low, while replacing two relative volume members with two volume increase members lifts it to the stage best. The margin lies in redundancy control rather than raw library size.

Revision accumulates along histories. Improve rewrites a turnover increase substitute into an earnings surprise gate on flat priced stocks, then changes the binary gates into continuous scores, two revisions of one history rather than fresh samples; the rounds that add its three generations backtest at -1.7%, $+0.4\%$, and $+2.7\%$. Combine keeps the trend term of a sentiment substitute parent and then uses it only inside the opposite condition of a range and volume parent; the sentiment parent never enters the final library although its children do.

Stopping is evidential, not scheduled. Revising a trend and volume combination at -0.6% into a one year quality combination lowers ARR to -0.8%, and the stage stops with two alphas produced in its final round; filtering spends fourteen rounds confirming that no edit outperforms the selected subset, then rolls back to it. The budget rule of Eq. 3 makes this policy. Capacity follows marginal evidence and stops once it is not obvious.

5 Conclusions

AutoScientist-Quant runs alpha discovery, alpha filtering, and model search under one controller that regards option choice and spend as decisions made during the run. On CSI universes it leads on nearly every metric in every setting, under two backbones and across four universes, and the ablations and case study explain the reasons. Exploration creates the information, downstream stages turn it into returns, and the search wins by recovering from dips it cannot see, controlling redundancy, and stopping on evidence.

In the future, the same search framework can be used in other domains such as algorithm development (Li et al., 2025d;b; 2026d; Romera-Paredes et al., 2024), reasoning (Yao et al., 2023a; Shinn et al., 2023; Li et al., 2025a;c), and evaluation (Zhou et al., 2026).

AI USE STATEMENT

In this work, we used tools mainly to polish the grammar and improve the readability of the manuscript. We did not use tools for other tasks. We have reviewed the texts and codes.

ETHICS STATEMENT

There are no ethics concerns in this work.

REPRODUCIBILITY STATEMENT

The main text and the appendix provide the details needed to reproduce our method.

REFERENCES

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018. URL https://arxiv.org/abs/1803.01271.

David Bailey, Jonathan Borwein, Marcos Lopez de Prado, and Qiji Jim Zhu. The probability of backtest overfitting. *The Journal of Computational Finance*, 20(4):39–69, 2017. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2326253.

Robert J Barro. The stock market and investment. *The review of financial studies*, 3(1):115–131, 1990. URL https://academic.oup.com/rfs/article/3/1/115/1576947.

Barry Bosworth, Saul Hymans, and Franco Modigliani. The stock market and the economy. *Brookings Papers on Economic Activity*, 1975(2):257–300, 1975. URL https://www.jstor.org/stable/2534104.

Muhammad Ali Chattha, Muhammad Imran Malik, Andreas Dengel, and Sheraz Ahmed. Krnn: A hybrid data and knowledge oriented time series forecasting approach for health care applications. *Expert Systems with Applications*, 268:126148, 2025. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2024.126148. URL https://www.sciencedirect.com/science/article/pii/S095741742403015X.

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*, KDD '16, pp. 785–794, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

Ash Demirgüç-Kunt and Ross Levine. Stock markets, corporate finance, and economic growth: an overview. *The World Bank Economic Review*, 10(2):223–239, 1996. URL https://academic.oup.com/wber/article/10/2/223/1684737.

Asli Demirgüç-Kunt and Vojislav Maksimovic. Stock market development and corporate finance decisions. *Finance and Development-English Edition*, 33(2):47–49, 1996.

Robert F Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the econometric society*, pp. 987–1007, 1982. URL https://www.jstor.org/stable/1912773.

Eugene F Fama. The behavior of stock-market prices. *The journal of Business*, 38(1):34–105, 1965. URL https://www.jstor.org/stable/2350752.

Eugene F Fama and Kenneth R French. The cross-section of expected stock returns. *the Journal of Finance*, 47(2):427–465, 1992. URL https://onlinelibrary.wiley.com/doi/full/10.1111/j.1540-6261.1992.tb04398.x.

Weilong Fu. The new quant: A survey of large language models in financial prediction and trading, 2025. URL https://arxiv.org/abs/2510.05533.

GLM-5-Team. Glm-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

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, 2026. URL https://arxiv.org/abs/2602.07085.

Campbell R Harvey and Yan Liu. Backtesting. *Available at SSRN 2345489*, 2015. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2345489.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 11 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Zura Kakushadze. 101 formulaic alphas. *Wilmott*, 2016(84):72–81, 2016. URL https://onlinelibrary.wiley.com/doi/abs/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 I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026a. URL https://openreview.net/forum?id=9VxTXAUH7G.

Zongqian Li, Ehsan Shareghi, and Nigel Collier. ReasonGraph: Visualization of reasoning methods and extended inference paths. In Pushkar Mishra, Smaranda Muresan, and Tao Yu (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 140–147, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-253-4. doi: 10.18653/v1/2025.acl-demo.14. URL https://aclanthology.org/2025.acl-demo.14/.

Zongqian Li, Yixuan Su, and Nigel Collier. Pt-moe: An efficient finetuning framework for integrating mixture-of-experts into prompt tuning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 83240–83261. Curran Associates, Inc., 2025b. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/77f33e8bd80345de4aea8554bbe5a4da-Paper-Conference.pdf.

Zongqian Li, Yixuan Su, and Nigel Collier. 500xCompressor: Generalized prompt compression for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25081–25091, Vienna, Austria, July 2025c. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1219. URL https://aclanthology.org/2025.acl-long.1219/.

Zongqian Li, Yixuan Su, and Nigel Collier. A survey on prompt tuning. In *ES-FoMo III: 3rd Workshop on Efficient Systems for Foundation Models*, 2025d. URL https://openreview.net/forum?id=JEMGDajQ1G.

Zongqian Li, Shaohan Huang, Zewen Chi, Yixuan Su, Lexin Zhou, Li Dong, Nigel Collier, and Furu Wei. Breaking training bottlenecks: Effective and stable reinforcement learning for coding models, 2026b. URL https://arxiv.org/abs/2603.07777.

Zongqian Li, Tengchao Lv, Shaohan Huang, Yixuan Su, Qinzheng Sun, Qiufeng Yin, Ying Xin, Scarlett Li, Lei Cui, Nigel Collier, and Furu Wei. Scaling data difficulty: Improving coding models via reinforcement learning on fresh and challenging problems, 2026c. URL https://arxiv.org/abs/2603.07779.

Zongqian Li, Yixuan Su, Han Zhou, Zihao Fu, and Nigel Collier. Flexi-lora with input-adaptive ranks: Efficient finetuning for speech and reasoning tasks. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5016–5020, 2026d. doi: 10.1109/ICASSP55912.2026.11462836.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024. URL https://arxiv.org/abs/2408.06292.

Chris Lu, Cong Lu, Robert Tjarko Lange, Yutaro Yamada, Shengran Hu, Jakob Foerster, David Ha, and Jeff Clune. Towards end-to-end automation of ai research. *Nature*, 651(8107):914, 2026. URL https://www.nature.com/articles/s41586-026-10265-5.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey, 2025. URL https://arxiv.org/abs/2402.06196.

OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily, Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b & gpt-oss-20b Model Card, 2025. URL https://arxiv.org/abs/2508.10925.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 6639–6649, Red Hook, NY, USA, 2018. Curran Associates Inc. URL https://papers.nips.cc/paper_files/paper/2018/hash/14491b756b3a51daac41c24863285549-Abstract.html.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024. URL https://www.nature.com/articles/s41586-023-06924-6.

William F Sharpe et al. The sharpe ratio. *Streetwise–the Best of the Journal of Portfolio Management*, 3(3):169–85, 1998.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=vAElhFcKW6.

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*, KDD '25, pp. 2813–2822, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3736838. URL https://doi.org/10.1145/3711896.3736838.

Richard J Teweles and Edward S Bradley. *The stock market*. John Wiley & Sons, 1998. URL https://www.wiley.com/en-ae/shop/general-finance-investments/the-stock-market-7th-edition-p-9780471191346.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, Mosharaf Chowdhury, and Mi Zhang. Efficient large language models: A survey. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=bsCCJHbO8A. Survey Certification.

Saizhuo Wang, Hang Yuan, Leon Zhou, Lionel Ni, Heung-Yeung Shum, and Jian Guo. Alpha-GPT: Human-AI interactive alpha mining for quantitative investment. In Ivan Habernal, Peter Schulam, and Jörg Tiedemann (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 196–206, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-334-0. doi: 10.18653/v1/2025.emnlp-demos.14. URL https://aclanthology.org/2025.emnlp-demos.14/.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance, 2023. URL https://arxiv.org/abs/2303.17564.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025. URL https://arxiv.org/abs/2504.08066.

Xiao Yang, Weiqing Liu, Dong Zhou, Jiang Bian, and Tie-Yan Liu. Qlib: An ai-oriented quantitative investment platform, 2020. URL https://arxiv.org/abs/2009.11189.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=5Xc1ecxO1h.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=WE_vluYUL-X.

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*, KDD '23, pp. 5476–5486, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030. 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, 2020. URL https://arxiv.org/abs/2002.08245.

Lexin Zhou, Lorenzo Pacchiardi, Fernando Martínez-Plumed, Katherine M Collins, Yael Moros-Daval, Seraphina Zhang, Qinlin Zhao, Yitian Huang, Luning Sun, Jonathan E Prunty, et al. General scales unlock ai evaluation with explanatory and predictive power. *Nature*, 652(8108):58–67, 2026. URL https://www.nature.com/articles/s41586-026-10303-2.

Method	Alpha	ARR	IR	MDD	CR
		GPT Backbone			
AlphaAgent	CUSTOM	-2.3% ± 1.0%	-0.321 ± 0.141	-15.1% ± 0.2%	-0.153 ± 0.062
	+ALPHA360	-1.5% ± 1.9%	-0.202 ± 0.254	-15.5% ± 1.9%	-0.098 ± 0.118
QuantaAlpha	CUSTOM	-1.6% ± 2.1%	-0.207 ± 0.262	-14.5% ± 2.9%	-0.096 ± 0.113
	+ALPHA360	-1.5% ± 1.8%	-0.189 ± 0.243	-13.3% ± 1.9%	-0.098 ± 0.135
AutoScientist-Quant	CUSTOM	**1.8% ± 1.3%***	**0.259 ± 0.200***	**-12.7% ± 0.8%**†	**0.143 ± 0.107***
	+ALPHA360	**3.0% ± 2.4%***	**0.401 ± 0.310***	**-12.0% ± 1.6%**†	**0.263 ± 0.211***
		GLM Backbone			
AlphaAgent	CUSTOM	-3.1% ± 2.1%	-0.463 ± 0.312	-15.3% ± 5.9%	-0.191 ± 0.055
	+ALPHA360	-3.3% ± 2.2%	-0.452 ± 0.291	-16.2% ± 3.4%	-0.190 ± 0.103
QuantaAlpha	CUSTOM	-3.6% ± 2.1%	-0.566 ± 0.345	-15.1% ± 4.2%	-0.222 ± 0.096
	+ALPHA360	-2.2% ± 1.5%	-0.297 ± 0.206	-13.9% ± 0.2%	-0.155 ± 0.107
AutoScientist-Quant	CUSTOM	**0.6% ± 0.8%***	**0.091 ± 0.108***	**-8.6% ± 0.5%**	**0.071 ± 0.087***
	+ALPHA360	**1.1% ± 1.2%***	**0.170 ± 0.184***	**-9.6% ± 1.0%***	**0.121 ± 0.138***

Table 6: **Variability and significance of portfolio metrics on CSI 300.** Mean ± standard deviation over the independent runs behind Tables 1 to 3. **Bold** marks the best method in each setting; * and † mark AutoScientist-Quant significantly better than both baselines and one baseline, respectively, under a one-sided Welch's t-test at $p < 0.05$.

Backbone	Alpha	Method	IC	ICIR	RIC	RICIR	ARR	IR	MDD	CR
			CSI 800							
GPT	+ALPHA20	AlphaAgent	0.033	0.215	0.029	0.194	-1.6%	-0.170	-6.2%	-0.205
		QuantaAlpha	0.034	0.227	0.031	0.210	-3.1%	-0.312	-5.6%	-0.399
		AutoScientist-Quant	**0.039**	**0.261**	**0.037**	**0.246**	**0.2%**	**0.002**	**-4.7%**	**0.292**
GLM	+ALPHA360	AlphaAgent	0.035	0.220	0.032	0.202	-1.9%	-0.196	-5.9%	-0.253
		QuantaAlpha	0.039	0.266	0.036	0.250	-2.2%	-0.255	-5.0%	-0.457
		AutoScientist-Quant	**0.042**	**0.275**	**0.038**	**0.253**	**2.1%**	**0.216**	**-4.8%**	**0.510**
			CSI 1000							
GPT	+ALPHA360	AlphaAgent	0.058	0.431	0.052	0.371	0.7%	0.064	-8.2%	0.093
		QuantaAlpha	0.059	0.458	0.054	0.397	-4.3%	-0.394	-11.2%	-0.375
		AutoScientist-Quant	**0.060**	**0.480**	**0.054**	**0.424**	**1.0%**	**0.090**	**-7.7%**	**0.123**
GLM	+ALPHA158	AlphaAgent	0.057	0.494	0.051	0.432	2.7%	0.253	-7.7%	0.352
		QuantaAlpha	0.058	0.494	0.053	0.439	-1.5%	-0.150	-7.7%	-0.189
		AutoScientist-Quant	**0.059**	**0.533**	**0.053**	**0.480**	**4.7%**	**0.504**	**-5.3%**	**0.888**
GLM	+ALPHA360	AlphaAgent	0.057	0.456	0.050	0.388	0.8%	0.070	-9.1%	0.085
		QuantaAlpha	0.059	0.469	0.052	0.406	3.1%	0.263	-7.8%	0.425
		AutoScientist-Quant	**0.059**	**0.523**	**0.053**	**0.468**	**3.9%**	**0.431**	**-6.6%**	**0.589**

Table 7: **Stricter post-release evaluation.** The three agentic methods are re-evaluated on the portion of the test window that postdates the release of both backbones, in the settings shown, with each value averaged over independent runs. **Bold** marks the best method in each setting, based on unrounded original values, so equal displayed values can differ in bolding.

A Appendix

A.1 Variability and Significance

Table 6 reports the run to run variability and statistical significance of the portfolio metrics in Tables 1 to 3.

A.2 Stricter Post-Release Evaluation

Table 7 re-evaluates the three agentic methods on the portion of the test window that postdates the release of both backbones.

A.3 Datasets

We use daily open, high, low, close, and volume data of the CSI and US universes from January 2015 to May 2026, accessed through Qlib (Yang et al., 2020), and take index members as of each historical date. Experiments cover six universes. **CSI 300** comprises the 300 largest and most liquid stocks and represents large caps, **CSI 500** comprises the next 500 largest and represents mid caps, **CSI 800** is the union of the two, and **CSI 1000** comprises 1,000 liquid small caps outside CSI 800.

S&P 500 comprises 500 US large caps, and **NASDAQ 100** comprises the 100 largest non-financial stocks listed on the Nasdaq. The main experiments in Table 1 use CSI 300, the market robustness analysis in Table 4 covers the other three CSI universes, and Table 5 covers the two US universes.

The data are split into four continuous windows. Training covers January 2015 to January 2020, validation covers January 2020 to June 2021 and guides early stopping, a feedback window from June 2021 to June 2023 supplies backtest feedback to the agents during alpha discovery, and a test window from June 2023 to May 2026 is reserved for final evaluation. The test window never overlaps any window the agents observe, so information from the evaluation period cannot reach the discovery loop. Scores are formed on day t and executed on day $t{+}1$, and the prediction target is the return from the close of day $t{+}1$ to the close of day $t{+}2$, namely $y_{i,t} = c_{i,t+2}/c_{i,t+1} - 1$, where $c_{i,t}$ is the close price of stock i on day t. This setup rules out the use of future information. Inputs and labels are rank normalized within each daily cross section.

A.4 Evaluation Metrics

We report four information metrics and four portfolio metrics. The information coefficient **IC** is the Pearson correlation between predicted scores and realized labels within each daily cross section, averaged over the test period, and **ICIR** divides the mean daily IC by its standard deviation to measure stability. **RIC** (Rank IC) and **RICIR** (Rank ICIR) are the Spearman counterparts and are robust to outliers.

Portfolio metrics come from a daily strategy that holds the 50 stocks with the highest scores and replaces at most 5 of them per day. Trades execute at the open price, pay 0.05% on purchase and 0.15% on sale, and are stopped for stocks at the 9.5% price limit; each universe is benchmarked against its own index, following the same backtest procedure as prior agentic baselines. **ARR** is the annualized excess return over the benchmark index, net of transaction costs, **IR** is the information ratio of daily excess returns, **MDD** is the maximum decrease of the cumulative excess return, and **CR** is the Calmar ratio equal to ARR divided by the absolute value of MDD, computed to eight decimal places and rounded to three for display. Each reported primary result is averaged over three independent runs, with each metric computed per run before averaging.

A.5 Baselines

Three standard alpha libraries from Qlib serve as reference input sets (Yang et al., 2020). ALPHA158 contains 158 alphas built from rolling price and volume functions, ALPHA360 combines the normalized price and volume values of the past 60 days into 360 inputs, and ALPHA20 is a small set of 20 representative alphas derived from ALPHA158 for the small input budget setting.

Baselines span three categories. Machine learning baselines fit **Linear**, **LightGBM** (Ke et al., 2017), and **CatBoost** (Prokhorenkova et al., 2018) models on each library, and deep learning baselines fit **LSTM** (Hochreiter & Schmidhuber, 1997), **TCN** (Bai et al., 2018), and **KRNN** (Chattha et al., 2025) models following the Qlib benchmark implementations. Agentic baselines generate new alphas with LLMs. **AlphaAgent** (Tang et al., 2025) regularizes alpha generation with originality, hypothesis consistency, and complexity bounds to keep alphas effective longer, and **QuantaAlpha** (Han et al., 2026) regards each run as a trajectory and improves alphas by revising and recombining trajectory parts. Both share our backbone, a search budget of 30 nodes and at most 108 generated alphas, and backtest procedure, and our method actually generates fewer alphas than the 108 of the baselines due to its flexible search and early stopping. R&D-Agent-Quant (Li et al., 2026a) is not included as a baseline, since its loop keeps each alpha by its gain on the same window it reports, the overlap of Section 1.2, and all previous agentic baselines report better results in published comparisons (Han et al., 2026). Our method follows the baselines in backtesting every library with a fixed LightGBM predictor before the model search stage. Every resulting library is evaluated alone as CUSTOM and merged with each standard library as the +ALPHA settings, which measures the added value of discovered alphas over predefined ones.

A.6 LOOKAHEAD PROBLEMS IN THE REUSED EVALUATION CODE

This section shows the two lookahead problems of Section 1.2, both in `backtest/runner.py` of the evaluation code reused from prior agentic pipelines (Tang et al., 2025; Han et al., 2026; Li et al., 2026a), with original lines marked by − and corrected lines by +.

The first problem is in the averaging of the information metrics.

```
  ic_series  = recorder.load_object("sig_analysis/ic.pkl")
  ric_series = recorder.load_object("sig_analysis/ric.pkl")
+ test_start, test_end = segments['test']
+ ic_series  = ic_series[(ic_series.index >= test_start) & (ic_series.index < test_end)]
+ ric_series = ric_series[(ric_series.index >= test_start) & (ric_series.index < test_end)]
  metrics['IC']   = float(ic_series.mean())
  metrics['ICIR'] = float(ic_series.mean() / ic_series.std())
```

The stored IC and Rank IC series cover the full prediction range rather than the declared test segment, so the original averaging mixes the training, validation, and evaluation periods, and the correction clamps both series to the test segment.

The second problem is in the segment filter of the data handler.

```
  def fetch(self, selector=None, ...):
      ...
-     if isinstance(selector, tuple) and len(selector) == 2:
+     if isinstance(selector, (tuple, list)) and len(selector) == 2:
          start, end = selector
          mask = (dates >= pd.Timestamp(start)) & (dates <= pd.Timestamp(end))
          result = result.loc[mask]
```

Segment boundaries parsed from the YAML configuration arrive as lists while the filter accepts only tuples, so the date mask is never applied and the handler silently returns the full history including the training span, which the correction fixes by accepting both types.