ALPHAGEAR: A SELF-EVOLVING AGENT FOR ALPHA MINING VIA LIBRARY-GUIDED EXPLORATION AND AUDITABLE REFINEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM agents can easily search alpha factors and explore program spaces at a scale difficult to achieve manually. The resulting candidates must then be selected and combined into a final predictor. Yet generating individually promising or diverse factors does not ensure that they contribute useful information to the final predictor, while repeated evaluation can amplify selection and feedback risks. We introduce AlphaGEAR, a self-evolving agent for constructing refined factor libraries. AlphaGEAR aligns search with deployment by guiding factor evolution with library-relative residual information and retaining complementary factors through stream-aware admission and marginal-contribution refinement for an equal-weight predictor. A discrete, query-limited certification interface makes additional held-out feedback bounded and auditable. Under a common equal-weight evaluation on CSI 300, AlphaGEAR achieves an average development RankIC of 0.0517 across three random seeds, compared with 0.0043 for the strongest competing agent, while retaining only 4–8 members against 23–44 in the competing libraries; the advantage persists after style orthogonalization. Frozen libraries pooled across search seeds attain forward RankICs of 0.0473, 0.0503, and 0.0589 on CSI 300, CSI 500, and CSI 1000, respectively, retaining 81%, 95%, and 103% of their corresponding development RankICs; every completed certified individual export is also positive out of sample. These results show that aligning LLM-driven exploration with library-level construction yields compact, auditable factor libraries that retain predictive value across windows and market universes.

1 INTRODUCTION

Alpha factors transform historical market observations into signals for ranking stocks by expected returns, providing inputs to systematic investment and portfolio construction. Discovering useful factors is difficult because financial observations are noisy, market relationships change, and predictive signals can decay as trading behavior adapts. Automated factor discovery enables large-scale hypothesis generation and evaluation (Gu et al., 2020; Chen & Dim, 2023), but its practical value depends on constructing signals that remain jointly useful beyond the data used for selection.

Automated approaches have progressed from searching symbolic expressions to reinforcement learning and LLM-based research agents. AlphaGen already rewards improvements to a downstream combination model, establishing joint predictive usefulness as a search objective (Yu et al., 2023); AlphaForge further couples factor generation with dynamic selection and weighting (Shi et al., 2025). Alpha-GPT introduced human–LLM interaction for factor mining, while AlphaJungle uses LLM-guided tree search to explore formulaic factors (Wang et al., 2023; Shi et al., 2026). More recent agents use language models to formulate financial hypotheses, write executable factors, and iterate on evaluation feedback, as illustrated by AlphaAgent, R&D-Agent(Q), and QuantaAlpha (Tang et al., 2025; Li et al., 2025; Han et al., 2026). These advances broaden the candidate space and automate the research loop.

A central challenge is ensuring that the evidence used to select a factor reflects its value in the final predictor. A high standalone score or low pairwise correlation does not establish that a candidate improves the deployed combination. Fitting combination weights may compensate for this mismatch,

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.

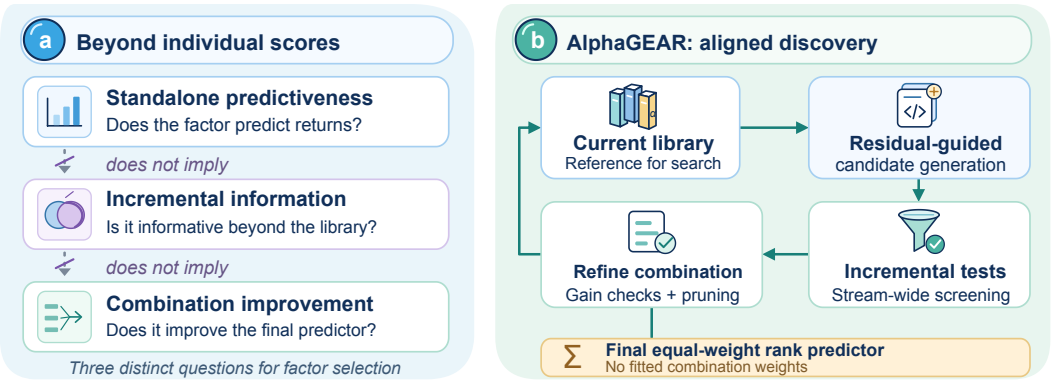


Figure 1: **AlphaGEAR aligns factor discovery with the final predictor. (a)** Standalone predictiveness does not establish incremental information beyond the current library; incremental information does not guarantee a better combination. **(b)** AlphaGEAR is a self-evolving agentic framework that guides LLM factor generation with library-relative residual information, screens incremental evidence across the candidate stream, and refines an equal-weight rank predictor through marginal-contribution checks and pruning on the discovery window. No combination weights are fitted.

but it introduces another estimation problem and can make apparent gains fragile to noise and regime shifts (Timmermann, 2006). Selection across many evaluated candidates creates a second risk. If rejected trials are omitted from the testing family, chance discoveries appear more convincing, while repeated development evaluation further favors patterns specific to that window (Harvey et al., 2016; White, 2000; Bryzgalova et al., 2022; Feng et al., 2020; Harvey & Liu, 2020; Hansen, 2005; Romano & Wolf, 2005). Numerical feedback from a held-out window creates a third risk. An adaptive agent can learn from successive evaluations without observing the underlying data, gradually turning the holdout into part of the search process (Dwork et al., 2015b; Blum & Hardt, 2015). Together, these effects allow a pipeline to report strong candidate-level statistics while producing a redundant, unstable, or evaluation-dependent library. Automated discovery must therefore align candidate selection with the final combination while limiting how evaluation feedback influences subsequent search.

We introduce **AlphaGEAR**, a self-evolving agentic framework that aligns factor discovery with the construction of its final predictor (Figure 1). A state-dependent controller schedules revision, recombination, simplification, and exploration, using library-relative residual information to direct the search toward complementary candidates. Statistical admission accounts for the recorded candidate stream, while refinement retains factors that improve an equal-weight predictor. A separate certification interface restricts held-out feedback to discrete responses under a fixed query budget and records each query. Together, our central empirical test is whether library-aware construction improves the final equal-weight predictor; residual-guided evolution shapes the candidate stream, while certification limits the held-out feedback available during search.

Building on this design, we make three contributions:

1. **Library-Guided Evolutionary Factor Search.** We design evolutionary factor search with residual predictive information relative to the current library, using a state-dependent controller to revise, recombine, simplify, and explore factor programs (§3.2).

2. **Stream-Aware Admission and Refinement.** We develop a factor-library admission and refinement mechanism that screens incremental evidence over the candidate stream and selects complementary factors for an equal-weight predictor (§3.3).

3. **Held-Out Certification under an Information Budget.** A discrete certification interface with single-use query tokens limits the information exposed by repeated held-out evaluation, while a frozen ledger records the resulting budget (§3.4).

2 Related Work

Automated alpha discovery. Formulaic alpha mining has moved beyond evaluating factors in isolation. AlphaGen rewards generation using the performance of a downstream combination model, while AlphaForge mines factors before dynamically selecting and weighting them (Yu et al., 2023; Shi et al., 2025). AlphaSAGE uses structure-aware GFlowNet exploration to encourage predictive and diverse formulas (Chen et al., 2026). These methods establish the importance of joint usefulness and diversity. For a fixed equal-weight rank predictor, however, incremental information and improvement to the final combination are distinct criteria: a candidate can add information beyond existing members yet lower the combined RankIC.

Asset pricing and the factor library. Machine-learning asset pricing expands the range of return predictors, while high-throughput research makes the scale of candidate evaluation explicit (Gu et al., 2020; Chen & Dim, 2023). Factor-library studies address model uncertainty and test whether proposed factors add to existing explanations (Bryzgalova et al., 2022; Feng et al., 2020). Work on false discoveries and replication further examines how selection affects reported evidence (Harvey & Liu, 2020; Jensen et al., 2023). For an agent that proposes executable programs sequentially, these concerns become concrete questions about which trials enter the evidence base and whether the chosen library works after selection ends.

Self-evolving agents for quantitative research. General program-search systems such as FunSearch and AlphaEvolve improve LLM-generated code through executable evaluation (Romera-Paredes et al., 2024; Novikov et al., 2025). In alpha mining, Alpha-GPT and AlphaJungle explore interaction and tree search (Wang et al., 2023; Shi et al., 2026), while AlphaAgent regularizes originality, hypothesis alignment, and complexity (Tang et al., 2025). R&D-Agent(Q) coordinates factor and model development (Li et al., 2025). More closely related to iterative research, QuantaAlpha evolves search trajectories, and AutoScientist-Quant uses a budget-conditioned controller across alpha discovery, library selection, and model search (Han et al., 2026; Li et al., 2026).

Statistical selection and adaptive feedback. Financial multiple-testing and data-snooping research shows why evidence from a search depends on the alternatives evaluated (Harvey et al., 2016; White, 2000; Hansen, 2005; Romano & Wolf, 2005). Benjamini–Hochberg provides a multiple-testing screen, while online procedures such as SAFFRON address sequential hypotheses under their own assumptions (Benjamini & Hochberg, 1995; Ramdas et al., 2018). Our screening family includes rejected candidates, but adaptive reuse of the development window does not establish formal online FDR control. Repeated access to held-out data poses a separate feedback problem studied by reusable-holdout and leaderboard methods (Dwork et al., 2015b; Blum & Hardt, 2015; Dwork et al., 2015a).

3 Method

AlphaGEAR follows a two-stage pipeline (Figure 2). During online search, library-relative residual information guides generation and admission updates the library. After search, refinement reconstructs the final equal-weight predictor from recorded candidates. A restricted certification interface bounds held-out feedback during search.

3.1 Problem Setup

Let $X_{\leq t}$ denote market observations available by day t, and let $y_{t,i}$ be the subsequent return of stock i. A factor program f maps $X_{\leq t}$ to a cross-sectional signal $s^f_{t,\cdot}$. Given finite generation and certification budgets, the task is to construct a library $\mathcal{L}$ whose combined signal ranks future returns. The search maintains a program archive $\mathcal{A}$, an admitted library $\mathcal{L}$, and separate certification records. Admission determines library membership; certification assesses a candidate against a reference library on held-out data.

Search and library construction use a visible window $\mathcal{W}_{\mathrm{dev}}$. A certification service accesses $\mathcal{W}_{\mathrm{sealed}}$ and returns only discrete feedback. The final predictor is frozen before evaluation on $\mathcal{W}_{\mathrm{test}}$. These

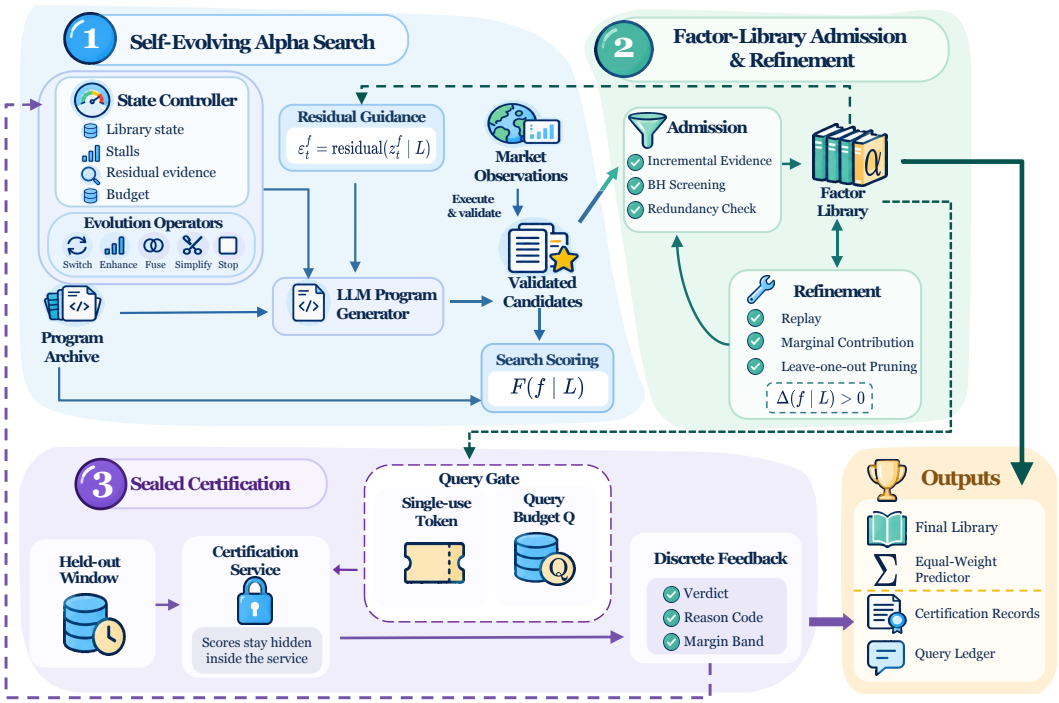


Figure 2: **AlphaGEAR integrates evolutionary search, library refinement, and bounded certification feedback.** **(1)** A state-dependent controller directs LLM program evolution. Residuals remove variation explained by the current library $\mathcal{L}$, and $F(f \mid \mathcal{L})$ scores candidates. **(2)** Statistical screening and redundancy checks govern admission. After online search, replay and pruning retain positive contributions to the equal-weight predictor on the visible window, expressed by $\Delta(f \mid \mathcal{L}) > 0$. **(3)** A token-based query gate limits sealed certification to Q queries. Only discrete feedback returns to the controller (purple dashed arrow); numerical scores remain hidden.

roles separate unrestricted search feedback, restricted certification feedback, and final assessment; experimental windows and metrics are defined in Section 4.1.

3.2 Self-Evolving Alpha Search

Search for complementary information. For candidate f, let $z_t^f = \text{rank}(s_{t,\cdot}^f)$, and let $Z_t^{\mathcal{L}}$ contain an intercept and current library ranks on common valid observations. The residual signal and its daily predictive correlation are

$$e_t^f = z_t^f - Z_t^{\mathcal{L}}(Z_t^{\mathcal{L}})^\dagger z_t^f, \qquad c_t^f = \rho_t(e_t^f, y_{t,\cdot}), \tag{1}$$

where $\dagger$ denotes the pseudoinverse and ρ_t is Spearman correlation. This projection measures information beyond the linear span of existing factor ranks. An empty library gives standalone evaluation. Candidate fitness is

$$F(f \mid \mathcal{L}) = a\, t_{\text{NW}}(c^f) + b\ \text{RankICIR}(s^f) - d\,[\rho_{\max}(f, \mathcal{L}) - \rho_0]_+, \tag{2}$$

where $a, b, d > 0$, t_{NW} summarizes residual evidence, RankICIR measures prediction stability, and $\rho_{\max}$ is the maximum absolute mean daily rank correlation with a library member. Each admission changes the residual target for subsequent candidates. Residuals guide search; admitted programs retain their original signals for combination.

State-dependent evolution. A rule-based controller uses remaining budget, admission stalls, library size, recent residual evidence, and unsuccessful improvement steps to choose an operator and batch size. SWITCH explores a new construction direction; ENHANCE revises a parent; FUSE recombines programs from different search branches when available; SIMPLIFY simplifies a program;

and STOP terminates a stalled search. Archive scores and an exploration bonus guide parent selection. The LLM receives parent programs, metrics, and available critiques, allowing later proposals to build on earlier evaluations. Candidates execute without return labels and undergo validity and prefix-truncation checks before admission. Controller rules and pseudocode appear in Appendix A.

3.3 FACTOR-LIBRARY ADMISSION AND REFINEMENT

Admission based on incremental evidence. We summarize c^f using $t_{\mathrm{NW}}(c) = \bar{c}/\sqrt{\hat{\Omega}/T}$, where $\hat{\Omega}$ is a Newey–West estimate of long-run variance (Newey & West, 1987). The screening family includes evaluated candidates beyond the admitted library. For m recorded one-sided reference p-values, Benjamini–Hochberg screening (Benjamini & Hochberg, 1995) selects

$$k^\star = \max\{k : p_{(k)} \le kq/m\}. \tag{3}$$

A candidate must belong to this rejection set and satisfy $\rho_{\max}(f, \mathcal{L}) < \rho_0$; an empty rejection set admits none. Including rejected candidates makes admission depend on the recorded search effort. Adaptive reuse of observations means this is a screening rule, without a formal online FDR guarantee.

Refinement for the combined predictor. The deployed signal averages the percentile ranks of available members:

$$S_{\mathcal{L},t,i} = \frac{1}{|\mathcal{L}_{t,i}|} \sum_{f \in \mathcal{L}_{t,i}} \mathrm{rank}_{01}(s^f_{t,\cdot})_i, \qquad J(\mathcal{L}) = \mathrm{RankIC}_{\mathcal{W}_{\mathrm{dev}}}(S_{\mathcal{L}}), \tag{4}$$

where $\mathcal{L}_{t,i}$ contains members with finite signals. No combination weights are fitted. After online search ends, refinement rebuilds the library from the recorded candidate stream using only $\mathcal{W}_{\mathrm{dev}}$: additional residual information need not improve this predictor. This distinction between incremental evidence and the quality of a fixed combination is also central in the forecast-combination literature (Timmermann, 2006). It recomputes statistical eligibility against the evolving library and requires a positive marginal contribution,

$$\Delta(f \mid \mathcal{L}) = J(\mathcal{L} \cup \{f\}) - J(\mathcal{L}) > 0. \tag{5}$$

After replay, leave-one-out refinement repeatedly removes the member with the smallest nonpositive contribution, stopping when all contributions are positive or one member remains. When a merged library is further refined, it undergoes the same check. Refinement aligns selection with the actual combination objective; its improvement criterion is defined on $\mathcal{W}_{\mathrm{dev}}$.

3.4 CERTIFICATION AND BOUNDED FEEDBACK

An admitted candidate can request certification against its reference library on $\mathcal{W}_{\mathrm{sealed}}$. The service evaluates residual predictive evidence and returns a tuple containing a verdict, reason code, and margin band. Numerical scores remain inside the service. Each query consumes a single-use token, and the total number of queries is capped at Q. Certification records retain the reference library used at query time; subsequent refinement does not re-certify a candidate.

Let Σ be the finite feedback alphabet, $V_{1:Q}$ the transcript padded after stopping, and Z the visible inputs and search randomness. If all certification-dependent feedback reaches search through this interface, adaptive queries are determined by Z and earlier verdicts. Therefore,

$$I(D_{\mathrm{seal}}; V_{1:Q} \mid Z) \le H(V_{1:Q} \mid Z) \le Q \log_2 |\Sigma|. \tag{6}$$

This finite-feedback bound (Dwork et al., 2015b; Blum & Hardt, 2015) limits additional information returned by certification. It is not a temporal generalization guarantee. Charges add across runs that reuse the window. After budget exhaustion, visible-window search can continue without certification queries. The final outputs include the library, equal-weight predictor, certification records, and a frozen query ledger. Appendix D specifies the interface and accounting details.

Table 1: **Automated factor-library discovery on CSI 300.** All exported libraries use equal-weight rank averaging. RankIC (higher is better) is reported as mean ± sample SD across three seeds; member counts are ranges. Bold marks the highest mean in each window. Forward scores evaluate frozen libraries on a future window. Neither data nor evaluation metrics from this window were available to the agent during search or refinement.

Method	Members	Historical	Development	Forward
QuantaAlpha	23–29	-0.0036 ± 0.0209	0.0036 ± 0.0189	0.0033 ± 0.0146
R&D-Agent(Q)	23–44	-0.0087 ± 0.0192	-0.0195 ± 0.0261	-0.0229 ± 0.0125
AlphaAgent	29–30	0.0222 ± 0.0100	0.0043 ± 0.0118	0.0037 ± 0.0068
AlphaGEAR	4–8	$\mathbf{0.0328 \pm 0.0046}$	$\mathbf{0.0517 \pm 0.0029}$	$\mathbf{0.0430 \pm 0.0035}$

4 EXPERIMENTS

We evaluate whether aligning factor discovery with the final equal-weight predictor yields compact libraries that remain useful after selection. The experiments examine library-level performance, temporal persistence, and the effect of construction on fixed candidate collections. Component analyses distinguish search from selection, while certification diagnostics account for the held-out feedback used during discovery.

4.1 EXPERIMENTAL SETUP

Data and protocol. We use daily A-share price–volume data with daily constituent membership for CSI 300, CSI 500, and CSI 1000. The four windows are historical $\mathcal{W}_{\text{train}}$ (2015–2019), certification $\mathcal{W}_{\text{sealed}}$ (January 2020–June 2021), development $\mathcal{W}_{\text{dev}}$ (July 2021–June 2023), and forward $\mathcal{W}_{\text{test}}$ (July 2023–May 2026). Search accesses a snapshot ending at $\mathcal{W}_{\text{dev}}$; certification thus tests an earlier period. The label is $y_{t,i} = \text{zscore}_i(P_{t+2,i}/P_{t+1,i} - 1)$, with execution at the next close. Appendix A.1 specifies embargoes and access rules.

Methods and metrics. We compare with QuantaAlpha (Han et al., 2026), R&D-Agent(Q) (Li et al., 2025), and AlphaAgent (Tang et al., 2025) on CSI 300, using three seeds per method, a common search budget of 100 iterations per seed, and the same LLM family. Every exported factor library is evaluated with equal-weight rank averaging (Appendix A.3). AlphaGEAR uses three seeds per universe and the same two-stage pipeline across all three universes: online library-guided search with admission, followed by refinement of recorded candidates on $\mathcal{W}_{\text{dev}}$. Each resulting library is frozen before forward evaluation (Appendix A.1). A union pools distinct members of the three refined seed libraries into one equal-weight predictor; the CSI 300 union undergoes an additional refinement pass. The primary metric is

$$\text{RankIC}_{\mathcal{W}}(s) = \frac{1}{|\mathcal{W}_s|} \sum_{t \in \mathcal{W}_s} \rho_t(s_{t,\cdot}, y_{t,\cdot}), \tag{7}$$

where ρ_t is cross-sectional Spearman correlation and $\mathcal{W}_s$ contains valid scoring days. We also report $\text{RankIC}^{\perp}$ after daily projection of signal ranks onto low-volatility, momentum, and traded-amount ranks; amount proxies liquidity and size, not market capitalization directly (Appendix A.2). Table 1 reports means ± sample standard deviations across three seeds; Table 2 reports one pooled, frozen predictor per universe.

4.2 OVERALL PERFORMANCE

Joint predictive usefulness. AlphaGEAR achieves the highest mean development and forward RankIC while retaining only 4–8 members, compared with 23–44 in the competing libraries (Table 1). Its forward mean is 0.0430, versus 0.0037 for the strongest comparator. The common equal-weight readout evaluates the usefulness of each exported library as a combination, with no additional weight-fitting stage. The pairing of stronger prediction and smaller libraries supports the goal of selecting factors for their joint contribution: retaining more candidates does not by itself

Table 2: **Forward RankIC remains around 0.05 and retains at least 80% of development RankIC across three CSI universes.** Each row evaluates one frozen equal-weight predictor pooled from three refined seed libraries; the CSI 300 union receives an additional development-window refinement pass. The libraries are searched independently within each universe. Retained is forward RankIC divided by development RankIC, rounded to the nearest percentage point; this ratio is descriptive.

Universe	Development RankIC	Forward RankIC	**Retained**
CSI 300	0.0584	0.0473	**81%**
CSI 500	0.0528	0.0503	**95%**
CSI 1000	0.0572	0.0589	**103%**

yield a better predictor. Table 1 compares complete agent outputs and therefore combines differences in candidate generation with differences in library construction; Table 3 holds each candidate collection and the equal-weight combiner fixed to assess the full construction policy.

Prediction beyond the selected style exposures. After style projection, AlphaGEAR retains the highest development and historical means, with development RankIC 0.0331 versus 0.0019 for AlphaAgent (Appendix C, Table 8). These exposures explain part of the raw signal but do not fully account for the advantage. Section 4.4 examines the role of construction more directly by holding the available candidates fixed.

4.3 Temporal Evaluation across Market Universes

We apply the two-stage search and refinement pipeline independently to CSI 300, CSI 500, and CSI 1000. The frozen pooled predictors achieve forward RankICs of 0.0473, 0.0503, and 0.0589, respectively (Table 2). All three remain around 0.05, retaining 81%, 95%, and 103% of their corresponding development RankICs. Thus, in each evaluated universe the final predictor preserves at least four-fifths of its development-window predictive rank association on the later forward window. These results support the effectiveness of applying the pipeline separately across the three constituent universes; they do not test transfer of one fixed library between universes.

Additional results for certified sublibraries and completed individual exports appear in Appendix Table 9; these assess certification rather than the frozen pooled predictor.

4.4 Ablations

Construction with fixed candidates. Table 3 tests whether standalone scores and pairwise correlation screening suffice to assemble a useful predictor. With the candidate collection and combiner fixed for each source, full admission and refinement raise mean forward RankIC by 0.0088–0.0149 while retaining 78–92% fewer members. These gains require no additional candidate generation and occur across all four reported sources. They directly support the distinction in Figure 1: selecting individually promising, nonredundant factors does not ensure a strong combination. The comparison evaluates the complete construction policy. The shared initialization-only reference (Full: 0.031583) provides the baseline for interpreting how much each constructed library adds beyond its initial signals (Appendix C.2).

Search strategy. Table 4 separates changes to candidate generation from replay on fixed streams. Removing the adaptive controller lowers both development and forward means, with forward RankIC falling from 0.0430 to 0.0392, supporting state-dependent scheduling in the evaluated search. Removing residual-guided search leaves the forward mean close to Full. With downstream construction retained, the evidence for adaptive scheduling is therefore clearer than for an additional average forward gain from residual guidance.

Admission and refinement. The replay variants show how selection rules shape the predictor. Omitting residual-based admission or restricting BH to the library family admits more members and raises mean forward RankIC. These constraints impose selectivity but do not maximize the observed

Table 3: **Fixed-candidate library construction on CSI 300.** Simple score-and-correlation screening and Full admission plus refinement share each source's frozen candidate collection and equal-weight combiner. Forward RankIC is mean ± sample SD across three seed exports; $|\mathcal{L}|$ is mean library size, and Δ is Full minus Simple.

Candidate source	Simple screening		Full admission + refinement		Δ
	RankIC	$\lvert\mathcal{L}\rvert$	RankIC	$\lvert\mathcal{L}\rvert$	RankIC
AlphaGEAR	$\mathbf{0.028836 \pm 0.008960}$	38.33	$\mathbf{0.042966 \pm 0.006586}$	5.33	$+0.014130$
QuantaAlpha	0.019869 ± 0.011689	19.33	0.034054 ± 0.003734	4.33	$+0.014185$
R&D-Agent(Q)	0.022587 ± 0.005119	21.00	0.031364 ± 0.00277	3.67	$+0.008777$
AlphaAgent	0.023910 ± 0.008822	28.33	0.038820 ± 0.009236	2.33	$+0.014910$

Table 4: **Component analysis for AlphaGEAR on CSI 300.** Search variants change candidate generation; admission variants replay fixed candidate streams. Library size is the mean member count; RankIC is mean ± sample SD across three seeds. The library-family BH variant restricts the testing set to current members and the candidate, excluding rejected trials.

AlphaGEAR variant	$\lvert\mathcal{L}\rvert$	Historical	Development	Forward
A. Search parts				
AlphaGEAR	6.3	0.0328 ± 0.0046	0.0517 ± 0.0029	0.0430 ± 0.0035
AlphaGEAR w/o residual-guided search	7.0	0.0354 ± 0.0050	0.0513 ± 0.0042	0.0436 ± 0.0049
AlphaGEAR w/o adaptive controller	5.3	0.0318 ± 0.0059	0.0495 ± 0.0007	0.0392 ± 0.0049
B. Admission parts				
AlphaGEAR	6.3	0.0328 ± 0.0046	0.0517 ± 0.0029	0.0430 ± 0.0035
AlphaGEAR w/o residual-based admission	8.0	0.0330 ± 0.0020	0.0546 ± 0.0017	0.0471 ± 0.0038
AlphaGEAR w/o marginal filter	6.0	0.0323 ± 0.0019	0.0511 ± 0.0023	0.0429 ± 0.0024
AlphaGEAR w/o LOO pruning	7.7	0.0336 ± 0.0016	0.0509 ± 0.0024	0.0438 ± 0.0028
AlphaGEAR w/o stream-wise BH	8.3	0.0326 ± 0.0042	0.0541 ± 0.0028	0.0451 ± 0.0049

forward score. Leave-one-out pruning reduces mean membership from 7.7 to 6.3, while forward means remain close (0.0438 without pruning, 0.0430 with it), illustrating the compactness obtained by refinement. The marginal filter has little effect on the forward mean when the other steps remain active. Together, the fixed-candidate and component comparisons support the complete policy over simple screening while distinguishing its overall benefit from the separate effects of its constraints.

4.5 Sensitivity and Certification Diagnostics

Threshold sensitivity. Figure 3 links admission stringency to the resulting library. Across the tested settings, relaxing BH screening increases development RankIC with nondecreasing membership, while raising the fixed NW threshold reduces both membership and development RankIC. Historical scores remain positive but vary non-monotonically.

Certification usage. The ledger makes held-out feedback an explicit part of the experimental record. All reported runs stay within the 20-query limit, using at most 12 queries per seed (Table 5). Under the interface assumptions of Eq. 6, each query incurs a conservative charge of $\log_2 96$ bits, accumulated across runs that reuse the window. Verdicts retain their candidate and reference-library provenance, including when retained through final refinement without new queries. These records document compliance with the feedback budget; forward prediction is assessed separately in Table 2. Appendix D details the accounting.

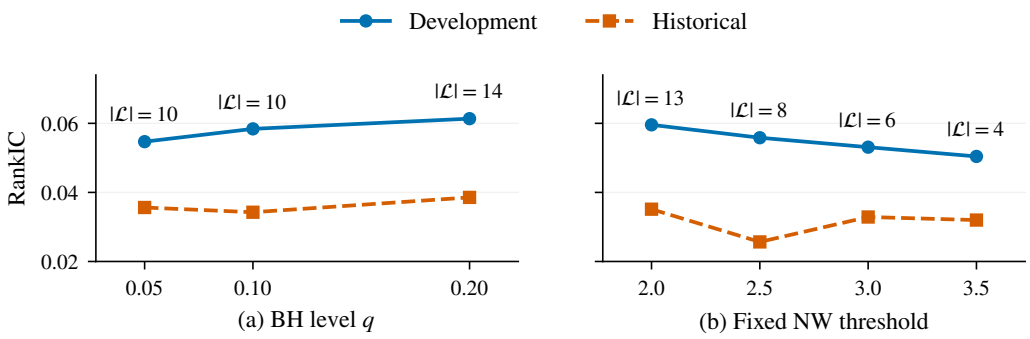


Figure 3: **Admission sensitivity on CSI 300.** Development and historical RankIC for merged libraries with identical refinement, varying (a) BH level or (b) fixed NW threshold. Point annotations give library size. Each point represents one merged predictor rather than a seed mean.

Table 5: **Certification usage during factor discovery.** Charged bits equal total queries times $\log_2 96$, rounded once; the limit is 20 queries per seed. Certified memberships count entries across seed libraries, not distinct factors in a union.

Universe	Queries by seed	Certified by seed	Charged bits, total
CSI 300	3/3/3	2/1/2	59.3
CSI 500	7/12/9	3/5/4	184.4
CSI 1000	12/7/10	7/3/4	191.0

5 Discussion

Across four fixed candidate collections, full library construction improves mean forward RankIC over score-and-correlation screening with substantially fewer members. This supports evaluating factors by their contribution to the deployed equal-weight predictor. Ablations qualify this result: the complete construction policy is effective in these comparisons, but individual constraints do not uniformly improve forward RankIC. In particular, removing residual-guided search or residual-based admission does not lower the observed mean. Certification addresses a separate concern. Its ledger documents that held-out feedback stayed within the query budget and preserves the candidate and reference library for each verdict; it does not establish a predictive benefit from certification.

6 Conclusion

AlphaGEAR treats an LLM factor-search agent as an adaptive analyst and aligns library-guided exploration with incremental-evidence screening and refinement for the deployed equal-weight combination. A query-limited sealed interface and frozen ledger bound and record additional held-out feedback. Under a preregistered protocol, libraries constructed independently across CSI 300, CSI 500, and CSI 1000 via online search then refinement retain positive forward RankIC. Under common equal-weight evaluation, the refined CSI 300 libraries achieve higher forward RankIC than three external-agent libraries as originally exported. Across four fixed candidate sources, the complete construction policy improves mean forward RankIC over simple score-and-correlation screening with substantially fewer factors. The framework offers a concrete way to assess agentic alpha discovery through both the predictor deployed and the evaluation feedback consumed during its construction. Future work will isolate the predictive effects of residual-guided search and certification and examine robustness across additional forward periods.

AI USE STATEMENT

During paper preparation, Large Language Models (LLMs) assisted with language editing, visualization code, and recalculation of supplementary statistics from author-provided experimental outputs. The experimental agents and competing implementations also invoked LLMs as part of their operation. The research ideas, experimental design, and original experimental data were provided by the authors.

ETHICS STATEMENT

The work uses publicly available daily market data and involves no human subjects. The paper's contribution is a mechanism that meters and discloses that risk. Nothing here constitutes investment advice.

REPRODUCIBILITY STATEMENT

All artifacts, including code, configuration files, and other relevant files will be released upon paper acceptance.

REFERENCES

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x.

Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1006–1014, Lille, France, 07–09 Jul 2015. PMLR. URL `https://proceedings.mlr.press/v37/blum15.html`.

Bryzgalova, Huang, and Julliard. Bayesian solutions for the factor zoo: We just ran two quadrillion models. *The Journal of Finance*, 78(1):487–557, 2022. doi: 10.1111/jofi.13197.

Andrew Y. Chen and Chukwuma Dim. High-Throughput asset pricing. *arXiv preprint arXiv:2311.10685*, 2023.

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. arXiv:2509.25055v3.

C. Dwork, V. Feldman, Moritz Hardt, T. Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. In *Advances in Neural Information Processing Systems 28*, 2015a. arXiv:1506.02629.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015b. doi: 10.1126/science.aaa9375.

Guanhao Feng, Stefano Giglio, and Dacheng Xiu. Taming the factor zoo: A test of new factors. *The Journal of Finance*, 75(3):1327–1370, 2020. doi: 10.1111/jofi.12883.

Gu, Kelly, and Xiu. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273, 2020. doi: 10.1093/rfs/hhaa009.

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 Chen, and Huacan Wang. QuantaAlpha: An evolutionary framework for LLM-Driven alpha mining. *arXiv preprint arXiv:2602.07085*, 2026.

Hansen. A test for superior predictive ability. *Journal of Business & Economic Statistics*, 23(4): 365–380, 2005. doi: 10.1198/073500105000000063.

Harvey and Liu. False (and missed) discoveries in financial economics. *The Journal of Finance*, 75 (5):2503–2553, 2020. doi: 10.1111/jofi.12951.

Campbell R. Harvey, Yan Liu, and Heqing Zhu. ... and the Cross-Section of expected returns. *Review of Financial Studies*, 29(1):5–68, 2016. doi: 10.1093/rfs/hhv059.

Jensen, Kelly, and Pedersen. Is there a replication crisis in finance? *The Journal of Finance*, 78(5): 2465–2518, 2023. doi: 10.1111/jofi.13249.

Yuante Li, Xu Yang, Xiao Yang, Minrui Xu, 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 38*, 2025. arXiv:2505.15155.

Zongqian Li, Yaoyiran Li, Yaohui Guo, Ming Zhang, Nigel Collier, and Eugene Ie. AutoScientist-Quant: Self-Evolving coding agents for automatic research in quantitative investment. *arXiv preprint arXiv:2608.28632*, 2026.

Newey and West. A simple, positive Semi-Definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica*, 55(3):703, 1987. doi: 10.2307/1913610.

Alexander Novikov, Ngân Vu, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, S. Shirobokov, B. Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. P. Kumar, Abigail See, Swarat Chaudhuri, George Holland, A. Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Aaditya Ramdas, Tijana Zrnic, Martin Wainwright, and Michael Jordan. SAFFRON: an adaptive algorithm for online control of the false discovery rate. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4286–4294. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/ramdas18a.html`.

Romano and Wolf. Stepwise multiple testing as formalized data snooping. *Econometrica*, 73(4): 1237–1282, 2005. doi: 10.1111/j.1468-0262.2005.00615.x.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024. doi: 10.1038/s41586-023-06924-6.

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. arXiv:2406.18394.

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, 2026. doi: 10.1609/aaai.v40i2.37069. arXiv:2505.11122.

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*, pp. 2813–2822, 2025. doi: 10.1145/3711896.3736838. arXiv:2502.16789.

Timmermann. Forecast combinations. *Handbook of Economic Forecasting*, pp. 135–196, 2006. doi: 10.1016/S1574-0706(05)01004-9.

Sai Wang, Hang Yuan, Leon Zhou, L. Ni, Harry Shum, and Jian Guo. Alpha-GPT: Human-AI interactive alpha mining for quantitative investment. *arXiv preprint arXiv:2308.00016*, 2023.

Halbert White. A reality check for data snooping. *Econometrica*, 68(5):1097–1126, 2000. doi: 10.1111/1468-0262.00152.

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.

A Experimental Protocol and Implementation

A.1 Data and execution protocol

Table 6: Common windows for CSI 300, CSI 500, and CSI 1000. The final two dates of each window are removed before scoring to keep $t + 2$ labels within the window.

Window	Dates	Use
$\mathcal{W}_{\text{train}}$ historical	2015-01-01 to 2019-12-31	baseline fitting; retrospective evaluation
$\mathcal{W}_{\text{sealed}}$ sealed	2020-01-01 to 2021-06-30	restricted certification
$\mathcal{W}_{\text{dev}}$ development	2021-07-01 to 2023-06-30	search, admission, and refinement
$\mathcal{W}_{\text{test}}$ forward	2023-07-01 to 2026-05-31	frozen-predictor evaluation

Labels and access. The label is $y_{t,i} = P^{\text{close}}_{t+2,i}/P^{\text{close}}_{t+1,i} - 1$, standardized cross-sectionally each day; a signal at close t is executed at close $t+1$. The development window contains 483 raw trading dates and at most 481 eligible scoring dates after removing its final two dates. Missing signals or labels can reduce the actual count further. The agent's data store is truncated at 2023-06-30. A separate forward evaluator accesses the full snapshot, consumes one pre-registered batch, and records the query before returning results.

Two-stage construction. All three universes use online library-guided search followed by development-window refinement of the recorded candidate stream. Seeds are random, with a nominal budget of 100 iterations and up to three candidates per batch; finishing a dispatched batch can exceed that budget. CSI 300/500/1000 initialize one/four/three programs, respectively, each blending three Alpha158 columns. Online admission updates the library that guides subsequent proposals. After search, NW evidence, stream BH, redundancy control, positive marginal contribution, and final leave-one-out pruning rebuild the library without new generation. Libraries are then frozen; the CSI 300 union receives an additional refinement pass after pooling seed libraries. The fixed-candidate control in Appendix C.2 instead gives every source the same four initial programs.

Verdict provenance and coverage. Refinement carries earlier certification verdicts by program fingerprint together with their query-time reference libraries; newly retained members receive no new verdict. An absent verdict therefore does not indicate failure. Historical evaluation is retrospective because learned baselines fit on $\mathcal{W}_{\text{train}}$. The third registered individual export for CSI 500/1000 lacks a forward result after the final-query budget was exhausted. One CSI 1000 union member failed execution checks, leaving 36 forward members. Historical unions have 1217 valid days for CSI 300/500 and 1120 for CSI 1000; completed individual forward signals have 702 valid days within the 704-date calendar window.

A.2 Metrics

Daily RankIC requires at least 20 valid stocks, and candidate evaluation requires at least 50 valid days. RankICIR divides mean daily RankIC by its sample standard deviation. For T valid daily correlations c_t, the lag-two Newey–West statistic is

$$t_{\text{NW}}(c) = \frac{\bar{c}}{\sqrt{\hat{\Omega}/T}}, \qquad \hat{\Omega} = \hat{\gamma}_0 + 2\sum_{\ell=1}^{2}\left(1 - \frac{\ell}{3}\right)\hat{\gamma}_\ell, \tag{8}$$

where $\hat{\gamma}_\ell$ is the sample autocovariance. The one-sided admission reference uses t_{480} with fixed $n_{\text{ref}} = 481$, the maximum eligible development count; this reference does not replace the actual T in Eq. 8.

For style projection, let $z_t = \text{rank}(s_{t,\cdot})$ and let Z_t^{style} contain an intercept and ranks of $-\text{STD}_{20}$, ROC_{20}, and $-\log \text{Amount}$. Every method uses

$$e_t^{\text{style}} = z_t - Z_t^{\text{style}}(Z_t^{\text{style}})^{\dagger} z_t, \qquad \text{RankIC}^{\perp}_{\mathcal{W}}(s) = \text{RankIC}_{\mathcal{W}}(e^{\text{style}}). \tag{9}$$

These controls represent low volatility, momentum, and traded amount; the last proxies liquidity and size, not market capitalization directly.

A.3 IMPLEMENTATION SETTINGS

All four agents use the same `deepseek-v4.1-flash` interface and a 100-iteration search budget per seed. Table 7 lists the remaining settings; the CSI 300 online fitness uses the ordinary residual t-statistic, while final refinement uses NW evidence in every universe.

Table 7: Search, evaluation, and construction settings.

Search	100 iterations; up to 3 parallel candidates; 1 island; checkpoint each iteration; seeds are random
Controller	Initial switches 6; enhance window 5; fuse every 6; simplify after 2 failures or > 4 terms, cooldown 6; switch after stall 8, cooldown 4; stop if remaining fraction < 0.3, stall ≥ 20, and recent $t < 1.5$
Parent selection	Archive UCB; $P(\text{best unvisited}) = 0.65$; exploration weight $1.0 \rightarrow 0.4$ over 100 iterations; 2 crossover inspirations
Online admission	CSI 300: ordinary $t \geq 3$. CSI 500/1000: NW lag 2, stream BH $q = 0.10$, $n_{\text{ref}} = 481$. All: $\rho_{\max} < 0.7$, target capacity 40
Final refinement	All universes: NW lag 2; stream BH $q = 0.10$; $n_{\text{ref}} = 481$; $\rho_{\max} < 0.7$; capacity 40; $\Delta_{\text{EW}} > 0$; final LOO pruning
Certification	20 queries per run; ordinary residual t threshold 2.0; margin band width 0.5; minimum query gap 1 iteration
Fitness (Eq. 2)	$a = 1/3$, $b = 0.5/0.30$, $d = 2$, $\rho_0 = 0.7$
Execution	4 prefix-truncation probe days; minimum 50 IC days and 20 stocks/day; sandbox timeout 1200 s
LLM interface	OpenAI-compatible API; temperature 1.0; 3 retries; timeout 6000 s
Learned baseline	LightGBM on Alpha158, fitted on $\mathcal{W}_{\text{train}}$ with fixed boosting rounds

A.4 ALGORITHM

Algorithm 1 AlphaGEAR: online search with certification, followed by library refinement

Require: Search budget M; certification limit Q; fixed protocol and admission rule
Initialize archive $\mathcal{A}$, library $\mathcal{L}$, controller state, and query ledger
while search budget remains **do** ▷ Stage 1: online search
  $(\text{op}, k) \leftarrow \textsc{Controller}(\mathcal{L}, \text{state}, M)$
  if $\text{op} = \textsc{stop}$ **then**
    break
  end if
  Generate up to k programs from selected parents and operator context
  for each generated program f **do**
    Execute and validate f; record visible feedback in $\mathcal{A}$
    if f is valid **then**
      Compute residual evidence and redundancy against $\mathcal{L}$
      Update the recorded test family when stream screening is enabled
      if f passes admission **then**
        Add f to $\mathcal{L}$ and apply capacity management
        if query budget and pacing permit **then**
          Consume a token; certify f against $\mathcal{L} \setminus \{f\}$ on $\mathcal{W}_{\text{sealed}}$
          Record the discrete verdict and feedback charge
        end if
      end if
    end if
    Update controller state from visible feedback
  end for
end while
Replay recorded candidates on $\mathcal{W}_{\text{dev}}$ ▷ Stage 2: final refinement
Rebuild $\mathcal{L}$ using NW tests, stream BH, $\rho_{\max} < \rho_0$, and $\Delta(f \mid \mathcal{L}) > 0$
Apply final leave-one-out refinement; retain earlier verdict provenance
Freeze artifacts; **return** $\mathcal{L}$, $S_{\mathcal{L}}$, certification records, and query ledger

B GENERATION PROMPT

This consolidated template summarizes the operator-specific prompts rather than transcribing every call. Braced fields are populated from available context; missing fields are omitted. The controller selects the operation, and STOP makes no generator call.

> **Task.** Produce one executable stock-ranking program that adds predictive information relative to the current library. The final predictor equally averages retained signals' cross-sectional percentile ranks, without fitted combination weights.
>
> **Context.** Permitted data and APIs: {TASK_SPECIFICATION}. Operation: {OPERATION}. Parent and supporting programs: {PARENTS}. Explored directions: {SEARCH_CONTEXT}. Development metrics, residual evidence, redundancy, errors, and critiques: {VISIBLE_FEEDBACK}. Optional certification tuple: {DISCRETE_FEEDBACK}. Output and editing rules: {OUTPUT_CONTRACT}. Omitted feedback is unavailable, not evidence of success or failure.
>
> **Operation.** SWITCH: explore a different construction direction. ENHANCE: revise a parent using its metrics and critiques. FUSE: integrate complementary mechanisms from supporting programs into the anchor. SIMPLIFY: remove exactly one term, family, or post-processing step, preserving remaining code and weights and a nonempty executable signal. Simplification edits a program, not library membership.
>
> **Program contract.** Implement `compute_signal(panel)` returning a floating-point array of shape `(n_days, n_instruments)`; larger values predict higher returns. Handle missing inputs as specified and support varying dates and universe sizes. At t, use only observations available through t and causal rolling or cross-sectional operations. Do not use future-fitted normalization, labels, external files, network access, environment variables, or evaluator internals. Use only permitted dependencies.
>
> **Feedback boundary.** Return the original signal; the evaluator handles residualization, admission, marginal contribution, and LOO pruning. Use development feedback and supplied discrete certification tuples only, without requesting numerical sealed scores or using forward feedback. Standalone performance is not evidence of combination improvement.
>
> **Output.** Return a complete program or permitted `EVOLVE-BLOCK`, preserving the interface and code outside the editable region. Include a brief comment explaining the intended contribution. Do not invent scores, admission decisions, or certification claims.

C ADDITIONAL RESULTS AND CONSTRUCTION CONTROLS

Table 8: **CSI 300 style-projected RankIC.** Signal ranks are projected on an intercept and low-volatility, momentum, and traded-amount ranks as defined in Appendix A.2. Values are means ± sample SD across three seeds. Forward style-projected results are not available.

Library	Development $\mathrm{RankIC}^{\perp}$	Historical $\mathrm{RankIC}^{\perp}$
QuantaAlpha	0.0005 ± 0.0059	-0.0082 ± 0.0107
R&D-Agent(Q)	-0.0101 ± 0.0140	-0.0042 ± 0.0083
AlphaAgent	0.0019 ± 0.0061	0.0143 ± 0.0070
AlphaGEAR	$\mathbf{0.0331 \pm 0.0028}$	$\mathbf{0.0199 \pm 0.0044}$

C.1 TEMPORAL AND CERTIFICATION DIAGNOSTICS

Certified sublibraries and completed individual exports also retain positive forward RankIC in each universe (Table 9). Certification assesses a candidate relative to its reference library at query time, whereas Table 2 evaluates the frozen pooled predictor. A certified subset therefore does not substitute for the final predictor, and factors retained through refinement without a new verdict should not be interpreted as having failed certification.

C.2 FIXED-CANDIDATE CONSTRUCTION AND ABLATIONS

Candidate collections. Table 3 includes four sources and three runs per source (12 Simple/Full pairs). The separate AlphaGEAR cohort uses three random seeds and 100 attempts per seed. Quan-

Table 9: **Additional temporal and certification results across CSI universes.** Union rows repeat the final-predictor results in Table 2 for context. Certified sublibraries retain factors that passed certification, and certified-individual rows summarize completed exports. Historical results are retrospective. Individual results are means ± sample SD, and n counts predictors.

Universe	Output	n	Historical	Development	Forward
CSI 300	union	1	0.0343	0.0584	0.0473
	Certified sublibrary	1	0.0405	0.0402	0.0291
	Certified individual	3	0.0387 ± 0.0019	0.0326 ± 0.0070	0.0348 ± 0.0063
CSI 500	Union	1	0.0592	0.0528	0.0503
	Certified sublibrary	1	0.0591	0.0436	0.0471
	Certified individual	2	0.0414 ± 0.0010	0.0373 ± 0.0029	0.0416 ± 0.0052
CSI 1000	Union	1	0.0660	0.0572	0.0589
	Certified sublibrary	1	0.0685	0.0545	0.0567
	Certified individual	2	0.0438 ± 0.0234	0.0474 ± 0.0064	0.0494 ± 0.0020

taAlpha ,R&D-Agent(Q) and AlphaAgent provide 23/29/24 ,23/44/30 and 100/93/100 exported candidates. Successful-export counts need not equal attempt counts. Each collection also receives the same four initial programs, which remain subject to selection.

Construction policies. Simple orients signals by development RankIC, sorts by its absolute value, and greedily retains candidates with absolute mean daily rank correlation below 0.7, up to 40 members. Full retains native directions and applies incremental NW tests, stream BH ($q = 0.10$, $n_{\mathrm{ref}} = 481$), the same correlation cap and capacity, positive marginal contribution, and final LOO pruning. Both select only on the eligible development dates: 483 raw dates minus the final two, leaving at most 481 scoring dates. Memberships are frozen before evaluation. External replay follows export-library order and tests only available exports and initial programs; unrecorded native trials cannot be reconstructed. The comparison therefore evaluates complete construction policies, including their direction handling, on the supplied collections.

Initialization and component controls. The initialization-only predictor has forward RankIC 0.029493 under Simple and 0.031583 under Full. Full retains on average 2.00, 3.00, and 2.33 initial members for AlphaGEAR, QuantaAlpha, and R&D-Agent(Q), respectively; the completed AlphaAgent full library retains one generated factor and no initial member. In Table 4, only admission variants hold generation fixed. Marginal filtering and final LOO pruning are omitted separately; the library-family BH variant excludes rejected trials. These controls do not establish a uniform performance benefit or formal validity guarantee for every component.

Paired statistical analysis. All 12 pairs improve in forward mean RankIC, while 10 of 12 decline in historical mean RankIC. Forward series share 702 dates (2023-07-03 to 2026-05-27), and historical series share 1217 dates (2015-01-05 to 2019-12-27), with no missing daily scores. Table 10 uses raw Spearman RankIC. We average the three runs within each source on each date, then apply a circular moving-block bootstrap with 20-day blocks, 5000 replicates, and a random seed. Each replicate samples the same date blocks for all sources and policies, preserving within-run pairing and contemporaneous dependence; run numbers do not imply cross-source pairing. Intervals are pointwise percentile 95% intervals, conditional on the observed runs, and do not quantify uncertainty from new search runs.

For contrast estimate $\hat{\theta}$ and bootstrap estimates $\hat{\theta}^*_b$, the two-sided null-centered value is $p = (1 + \sum_{b=1}^{5000} \mathbf{1}\{|\hat{\theta}^*_b - \hat{\theta}| \geq |\hat{\theta}|\})/5001$. Holm correction covers ten contrasts: four construction effects, three source differences under Full, and three source–construction interactions. At adjusted $\alpha = 0.05$, construction gains are supported for AlphaGEAR and QuantaAlpha; AlphaAgent's interval excludes zero but its adjusted $p = 0.0560$, while the R&D-Agent(Q) interval includes zero. No interaction survives correction. Source differences compare these fixed collections and do not isolate generator quality or establish additional synergy with our construction policy.

Table 10: **Forward construction contrasts.** Estimates and pointwise 95% intervals use daily raw RankIC. The final column adjusts two-sided bootstrap p-values over all ten contrasts; intervals themselves are not multiplicity-adjusted.

Source / comparator	Estimate	95% interval	p_{Holm}
Construction: Full − Simple			
AlphaGEAR	$+0.01413$	$[0.00427, 0.02412]$	0.0288
QuantaAlpha	$+0.01419$	$[0.00415, 0.02458]$	0.0364
R&D-Agent(Q)	$+0.00878$	$[-0.00104, 0.01902]$	0.2490
AlphaAgent	$+0.01491$	$[0.00354, 0.02659]$	0.0560
Full source difference: AlphaGEAR − comparator			
QuantaAlpha	$+0.00891$	$[0.00386, 0.01397]$	0.0072
R&D-Agent(Q)	$+0.01160$	$[0.00716, 0.01607]$	0.0020
AlphaAgent	$+0.00415$	$[0.00105, 0.00710]$	0.0420
Interaction: AlphaGEAR construction gain − comparator gain			
QuantaAlpha	-0.00006	$[-0.00490, 0.00511]$	1.0000
R&D-Agent(Q)	$+0.00535$	$[-0.00014, 0.01084]$	0.2224
AlphaAgent	-0.00078	$[-0.00579, 0.00406]$	1.0000

C.3 External-library screening

Table 11 screens frozen external libraries in discovery order using residual evidence, the correlation cap, and candidate-stream BH at $q = 0.10$, then certifies admitted candidates. These auxiliary results use the archived exports, distinct from full construction with refinement. Certification ran in a separate script; its bit charges are accounting references, not online search-ledger entries.

Table 11: **External-agent admission and certification.** Certification refers to the candidate and reference library at query time.

Method	Generated	Admitted	Certified	Queries	Bits
QuantaAlpha run 0	23	0	0	0	0.0
QuantaAlpha run 1	29	1	0	1	6.6
QuantaAlpha run 2	24	1	0	1	6.6
R&D-Agent(Q) run 0	23	1	0	1	6.6
R&D-Agent(Q) run 1	44	1	0	1	6.6
R&D-Agent(Q) run 2	30	1	0	1	6.6
AlphaAgent run 0	29	1	1	1	6.6
AlphaAgent run 1	30	1	0	1	6.6
AlphaAgent run 2	29	3	0	3	19.8

D Certification Interface and Accounting

The verdict alphabet is $\Sigma = \mathcal{V} \times \mathcal{R} \times \mathcal{B}$ with $\mathcal{V} = \{\text{PROMOTE}, \text{INCONCLUSIVE}, \text{REJECT}\}$, $\mathcal{R}$ the eight reason codes (robust improvement; insufficient power; no improvement; unstable across seeds; regime fragile; cost unacceptable; protocol violation; evaluation error), and $\mathcal{B} = \{\text{CLEAR_PASS}, \text{MARGINAL}, \text{CLEAR_FAIL}, \text{UNKNOWN}\}$. The factor-library worker computes an ordinary residual t-statistic against the reference library on $\mathcal{W}_{\mathrm{sealed}}$, excluding the candidate itself. This scalar is the margin μ: $\mu \geq \delta = 2.0 \Rightarrow$ PROMOTE; $0 \leq \mu < \delta \Rightarrow$ INCONCLUSIVE; $\mu < 0 \Rightarrow$ REJECT. Bands are clear pass for $\mu \geq 2.5$, marginal for $0 \leq \mu < 2.5$, and clear fail for $\mu < 0$. Failed evaluations or protocol checks can return an unknown band. The deterministic factor worker supplies one statistic; the generic interface also contains reason codes for other evaluation modes. Only the discrete tuple is returned to search. The bound $\log_2 96$ counts the full interface alphabet even when some tuples are unreachable in this mode.

Feedback bound. Conditional on visible inputs and search randomness Z, each query depends on preceding discrete verdicts. Padding the transcript after stopping yields at most $|\Sigma|^Q$ possibilities

and hence Eq. 6. Raw scores, timing, and error traces must remain unavailable to search through other channels. With three verdicts, eight reason codes, and four bands, each query is charged $\log_2 96 \approx 6.585$ bits, at most 131.7 bits for 20 queries. Charges add across runs reusing the window; the bound controls interface feedback, not temporal generalization.