AGENTIC FACTOR ENGINEERING: TURNING INVESTMENT QUESTIONS INTO FACTORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Factor construction turns investment ideas into numerical signals for quantitative investment. Automated methods can generate factor formulas and programs, but a single fixed formula cannot fully capture many investment judgments. For example, assessing whether a company's new capacity has begun to translate into deliveries requires examining disclosures, production data, and price records together. The key evidence differs across companies. We propose *Agentic Factor Engineering*, using investment questions to define *natural-language factors*. An executor agent follows common execution and scoring rules to read records, perform calculations, and assign each stock a score with supporting evidence. We also explore search over natural-language factors through return-guided evolution, using return evaluations and company cases to revise questions and expand the candidate bank. We then select ensembles from all generations using validation performance. Experiments on fundamental and technical factors in CSI300 and Nasdaq-100 support question-based measurement through formula comparisons, controlled evidence changes, and repeated execution. In chronologically held-out tests, selected ensembles achieve higher information coefficients (IC) and rank information coefficients (RankIC) than initial ensembles in three of four settings, suggesting predictive potential for this approach to factor construction.

1 INTRODUCTION

Factor construction turns investment ideas into numerical signals for comparing stocks and predicting returns. Many traditional factors use formulas over price–volume data (Kakushadze, 2016). Automated methods search for new formulas and programs through evolutionary search and reinforcement learning (Cui et al., 2021; Yu et al., 2023). More recently, large language models (LLMs) and agents have been used to propose investment hypotheses, write factor programs, and revise them using evaluation feedback (Li et al., 2024; Tang et al., 2025; Li et al., 2026; Lin et al., 2026). These methods broaden the ideas that can be explored and reduce the manual work needed to implement factors.

However, a single fixed formula cannot fully capture many investment judgments. For example, assessing whether a company's new capacity has begun to translate into deliveries requires examining disclosures, production data, and price records together. The key evidence differs across companies. A fixed formula may capture only part of this judgment, and extraction rules written for one company's reports may not apply to another's. Existing work explores extracting signals from text and turning trading ideas into programs (Wang et al., 2024; Choi et al., 2025; Yi et al., 2026). Agents can also retrieve records, perform calculations, and compare sources to form investment judgments (Yao et al., 2022; Zhang et al., 2024; Yu et al., 2024). This motivates our question: *Can investment questions directly define factors, with an agent assigning each stock a score and explaining the evidence behind it?*

We propose *Agentic Factor Engineering*, in which investment questions describe the properties to measure. Measurement here means answering a question using evidence and assigning a score that indicates how well a company meets the conditions in that question. An executor agent performs this assessment through autonomous reasoning and tool use. The first challenge is to obtain comparable scores when the available evidence varies across stocks and dates. The executor agent therefore uses only information available by the scoring date and follows a common scoring rubric for each

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.

question. Each score is accompanied by supporting evidence, counter-evidence, and information gaps. The question and these execution rules jointly define a *natural-language factor*; its scores are the factor values for individual stocks.

The second challenge is to incorporate question-defined factors into automated development and selection. We explore this through return-guided revision, recombination, and expansion. A reflection agent uses return evaluations, execution evidence, and company cases to suggest changes to economic hypotheses and evidence requirements. An evolution agent generates the next population through elite preservation, crossover, reflection mutation, and exploration. A factor bank retains candidates from all generations for selection on a separate validation period. The question set and aggregation rule are fixed before testing.

We evaluate measurement and prediction in fundamental and technical domains on CSI300 and Nasdaq-100 (NDX). Formula comparisons test whether scores reflect the intended properties; controlled evidence changes test whether scores respond to relevant information; repeated execution tests score consistency. In chronologically held-out tests, selected ensembles achieve higher IC and RankIC than initial ensembles in three of four settings. These results support the feasibility of natural-language factors and suggest potential for question evolution and selection in factor discovery. Our main contributions are as follows:

- **Natural-language factors and agentic measurement.** We define factors through investment questions and use an executor agent to score stocks from available records and document its reasoning.
- **Search over natural-language factors.** Return-guided evolution expands a cross-generation question bank for validation-based selection, demonstrating compatibility with automated factor development.
- **Empirical evaluation of measurement and prediction.** Across two stock universes and both domains, we test whether scores measure the intended properties, respond to relevant evidence, remain consistent across runs, and relate to out-of-sample returns.

2 Related Work

Automated Factor Discovery. Automated factor discovery searches for formulas and programs through evolutionary methods, reinforcement learning, and LLM-guided generation (Cui et al., 2021; Yu et al., 2023; Li et al., 2024; Tang et al., 2025). Recent work expands what can be revised: FactorEngine revises general factor programs and separates logic revision from parameter optimization (Lin et al., 2026). AlphaSchema searches structured trading plans, which an LLM then implements as factors (Yi et al., 2026). R&D-Agent-Quant jointly optimizes factors and prediction models (Li et al., 2026). AgonAlpha combines factor search with evidence records and independent agent review (Ye et al., 2026). We focus on how factors are measured: an agent answers a question using the evidence available for each stock and date, writing analytical code when needed.

Textual and Semantic Financial Signals. Financial text has long supplied signals through sentiment analysis and learned representations (Tetlock, 2007; Loughran & McDonald, 2011; Chen et al., 2022). LLMs extend this work to news interpretation and stock ratings (Lopez-Lira & Tang, 2026; Papasotiriou et al., 2024). LLMFactor extracts explanatory factors from news and combines them with price histories to predict stock movements (Wang et al., 2024). From Text to Alpha uses embeddings of metric-related passages to measure changes in the business metrics emphasized in disclosures (Choi et al., 2025). These studies show how language can support financial measurement. We use questions to guide retrieval and computation over available tables and text, then test separately whether the resulting scores reflect the intended properties and relate to subsequent returns.

Financial Research and Trading Agents. FinAgent combines multimodal information with tools (Zhang et al., 2024). FinCon coordinates managers and analysts through verbal feedback for financial decisions (Yu et al., 2024). FinMem uses layered memory to guide later actions (Yu et al., 2025). AlphaCrafter coordinates factor discovery, selection, and portfolio execution through specialized agents (Yuan et al., 2026). We draw on these capabilities to evaluate stock scores pro-

duced from investment questions. This lets us separately test which evidence the agent uses, how scores vary across runs, and whether they relate to stocks' subsequent returns.

Language-Based Optimization and Reflection. Prompt optimization searches for language instructions using task performance (Zhou et al., 2022; Yang et al., 2024). EvoPrompt and Promptbreeder use evolutionary operators. Reflexion turns outcomes into verbal feedback (Guo et al., 2024; Fernando et al., 2024; Shinn et al., 2023). In factor discovery, FactorMiner uses skills and experience memory (Wang et al., 2026), while AlphaMemo records which factor edits succeed or fail (Yu et al., 2026). QuantaAlpha applies mutation and crossover to end-to-end research trajectories (Han et al., 2026). Building on these methods, we use reflection and evolution to refine factor questions, then select candidates using validation performance. Refinement uses development-period feedback; the execution protocol and execution model parameters remain fixed.

3 Problem Formulation

Let $\mathcal{T}$ be a set of scoring dates and $\mathcal{U}_t$ the stock universe at date t. Each t is the historical date represented by the score; the agent may run later. For stock $i \in \mathcal{U}_t$, $x_{i,t} \in \mathcal{X}$ denotes the information available at that date, including structured records, financial reports, research text, and market history. Point-in-time access uses t as the information cutoff date. Let $\mathcal{H}$ be the set of return horizons and $r_{i,t,h}$ the subsequent return measured using the prescribed entry and exit dates for horizon $h \in \mathcal{H}$. These outcomes are available to the offline evaluator, not to the executor when scoring stocks for t. Cross-sectional evaluation compares exposures and subsequent returns across stocks at a fixed date t. We use chronologically separated training, validation, and test periods for candidate development, selection, and evaluation, respectively. Their date sets are denoted by $\mathcal{T}_{\text{dev}}$, $\mathcal{T}_{\text{val}}$, and $\mathcal{T}_{\text{test}}$; we refer to the training period as the development period below.

Factor Specification. For domain d, let $\mathcal{Q}_d$ be the set of candidate questions and $\mathcal{H}_d \subseteq \mathcal{H}$ its target return horizons. A natural-language question $q \in \mathcal{Q}_d$ describes an economic property. The same question is evaluated across stocks and dates under protocol θ:

$$s_{i,t}^{(q)} = F_\theta(q, x_{i,t})\,. \tag{1}$$

The protocol includes the model, tools, prompts, question context, output format, and scoring rubric. The pair (q, θ) defines a natural-language factor. $s_{i,t}^{(q)}$ is the score from one run. Scores lie in $\{0, 1, \ldots, 5\}$, with higher values indicating stronger evidence for the specified positive judgment.

The score expresses how well a company meets the conditions in the question. For example, a cash-flow coverage question asks whether operating cash flow can support the specified expenditure. A higher score means stronger support for that condition. Self-rated confidence records the agent's certainty for review and reflection; the aggregation below uses only property scores. Subsequent returns test whether the property has predictive value.

Factor Aggregation. A question's raw scores are its factor values (exposures) across stocks. For a nonempty question set $\mathcal{C}_d$ with $K_d = |\mathcal{C}_d|$, let $\mathcal{B}_{d,t}$ contain stocks with valid scores for every question. Percentile ranks across stocks place scores from different questions on a common scale:

$$z_{i,t}^{(q)} = \frac{\text{rank}_{j \in \mathcal{B}_{d,t}}\left(s_{i,t}^{(q)}\right) - \frac{1}{2}}{|\mathcal{B}_{d,t}|}. \tag{2}$$

We rank scores from lowest to highest, starting at one. Tied scores receive their average rank. Averaging the percentile ranks gives

$$S_{i,t}^{(d)} = \frac{1}{K_d} \sum_{q \in \mathcal{C}_d} z_{i,t}^{(q)}. \tag{3}$$

The same questions and weights apply to all stocks in the common set. Individual IC uses raw scores s, whereas ensemble IC uses S. Formula comparisons test whether scores agree with numerical measurements of the intended property; return-based metrics assess predictive value. Full definitions appear in Appendix D.

4 METHOD

Figure 1 shows how Agentic Factor Engineering measures natural-language factors and revises their definitions. The executor agent scores stocks against manually specified or evolved questions under common rules. Return evaluation and reflection on company cases guide question refinement and expansion.

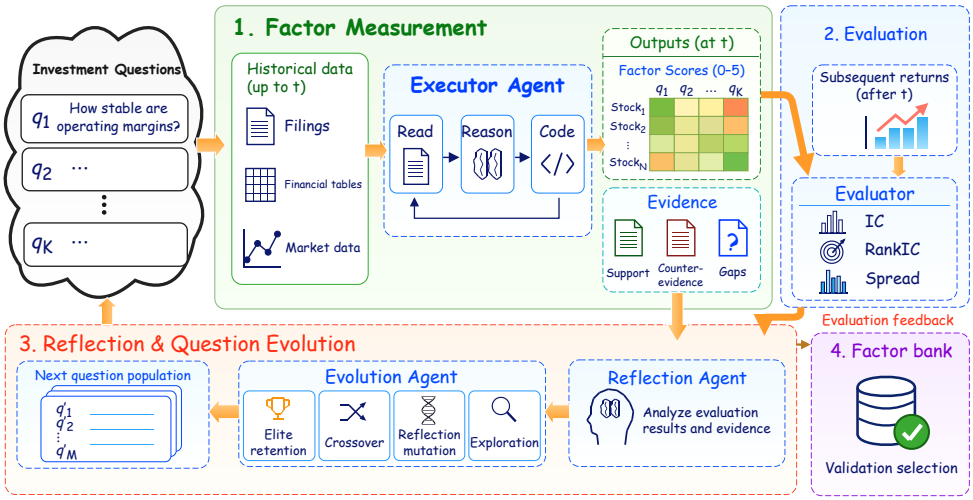

Figure 1: Agentic factor measurement from historical evidence, return-guided question evolution, and validation-based selection from the factor bank.

4.1 FACTOR EXECUTION

Agentic Measurement. The executor agent turns an investment question into numerical scores by answering it for each stock. It first checks the stock and scoring date, then reads the data guide and retrieves relevant tables and text. When needed, it writes code for calculations and checks sources against one another before scoring. The question determines which facts matter and which conditions they must satisfy. This process draws on interleaved reasoning and tool use (Yao et al., 2022) and financial agents' analysis of multiple sources (Zhang et al., 2024; Yu et al., 2024). The historical data view filters external records by their associated availability dates up to t.

The Huali case illustrates this process. The question asks whether new business or capacity has begun to produce observable results. The agent combines disclosures of newly operating factories with shipments to new customers. Growth in shipment volumes and revenue provides further support. These facts support a high score, while margin pressure during factory ramp-up is retained as counter-evidence. The score thus describes the progress of operating activity; future profitability and stock returns are separate outcomes. Appendix F.1 gives the condensed response.

Each run returns a property score, a rationale, supporting and opposing evidence, and information gaps. Scores are used for quantitative evaluation; the other outputs explain the judgment and inform reflection. The model version, agent runtime, and data interfaces remain fixed within each comparison. Appendix C provides the prompt, output format, execution settings, and a complete execution trace.

Return Evaluation. The evaluator uses deterministic calculations to pair factor values with subsequent returns at each scoring date and horizon and compute IC, RankIC, and Spread (Appendix D). It summarizes test performance using Equation 15 and retains question statistics for each horizon to guide search (Equation 5). It also selects company cases from different return ranges. The reflection agent uses their execution evidence together with the aggregate statistics to compare and refine candidates.

4.2 RETURN-GUIDED QUESTION EVOLUTION

As in prior factor search (Tang et al., 2025; Han et al., 2026; Yi et al., 2026), we use return feedback to revise candidates. Candidates are questions q under a fixed execution protocol θ. Fundamental and technical questions target longer and shorter horizons, respectively. Predefined questions cover diverse ideas and evidence types; development-period returns and case-based reflection guide revision, recombination, and expansion.

Reflection and Operators. The reflection agent uses development-period results to suggest changes to questions. Return statistics show which questions produce scores associated with subsequent returns. Company cases show which facts led to those scores. The agent reads both to check whether the economic explanation fits the execution evidence. It then identifies evidence requirements or conditions that need clarification. Related approaches include verbal reflection in Reflexion (Shinn et al., 2023), investment belief updates in FinCon (Yu et al., 2024), and reward-guided prompt optimization in ATLAS (Papadakis et al., 2025).

Reflection has three outputs. A mechanism summary explains which information the agent used and why it may relate to returns. An empirical assessment describes the relationship's direction, performance across dates, and conditions of use. Evolution recommendations propose specific changes to questions. For example, a broad question about business expansion could be refined to require delivery evidence or specify the period to examine. Such recommendations turn return feedback into requirements that the next generation of questions can ask the executor to check.

The evolution agent reads evaluations and reflections to identify investment ideas to retain, evidence to combine, wording to revise, and new directions to explore. It then generates the next population through four operators:

- *Elite preservation* carries forward high-performing questions unchanged.
- *Crossover* combines distinct information from two related parents into one investment judgment, such as an economic explanation and the evidence needed to assess it.
- *Reflection mutation* preserves a parent's core economic meaning while revising evidence requirements, judgment direction, or scope using empirical feedback.
- *Exploration* proposes new perspectives within the domain, expanding investment ideas and evidence types covered by the questions.

Appendix C.4 gives formal definitions of the search objective, population update, and factor bank.

Selection on Validation Data. A cross-generation factor bank retains candidates and their evaluation records, merging texts that are identical after normalization. Earlier questions remain eligible for selection even when no longer active in the evolving population. After evolution, candidates are scored on the separate validation period:

$$F_{\mathrm{val},d}(q) = \min_{h \in \mathcal{H}_d} \mathrm{Median}_{t \in \mathcal{T}_{\mathrm{val}}} \mathrm{RankIC}_{q,t,h}. \tag{4}$$

Here $\mathrm{RankIC}_{q,t,h}$ is the Spearman correlation between question scores and subsequent returns at date t and horizon h. The median summarizes typical performance across dates for each horizon. We then use the weaker horizon as the selection score. For example, median RankIC values of 0.08 and 0.02 give a selection score of 0.02. Stronger performance at one horizon therefore cannot mask weaker performance at the other. The top K_d questions form $\mathcal{C}_d$ and are fixed before testing. Their percentile ranks are averaged using Equation 3.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Data and Protocol. Experiments on CSI300 and NDX, two large, liquid stock universes, examine what the agent measures, whether scores relate to returns, and how questions evolve in fundamental and technical domains. Sensitivity analyses mainly use NDX. Data include price and volume records, fundamental indicators, announcements, financial statements, and other structured records.

Relatively complete company information and historical data in both universes make them suitable for evaluating measurement that uses evidence from multiple sources. Data are preprocessed, cleaned, and organized by scoring date so each run accesses only records available by that date. Appendix B describes the sources and organization.

Training spans April 2024 to March 2025, validation July to September 2025, and testing January to March 2026. These chronologically separated periods support candidate development, selection, and final evaluation, respectively. During training, we develop questions without updating the execution model's parameters. Measurement takes place at the beginning of each month, with a three-month interval between adjacent stages to accommodate subsequent-return windows.

Execution and Search Settings. Each universe–domain setting starts with 12 predefined general investment questions chosen to cover diverse investment hypotheses and evidence types. The agent measures these questions in batches. Monthly training samples are stratified by industry and market capitalization, yielding 300 agent-task samples across scoring dates and stock characteristics. Validation and test panels are determined and frozen in advance. DeepSeek-v4-flash (Xu et al., 2026) is the main executor, returning a 0–5 property score, confidence, an answer, and supporting and opposing evidence for each question. Appendix C provides the prompt and execution settings. Evolution runs for five updates per universe–domain setting, with elite preservation, crossover, reflection mutation, and exploration in proportions of 1/6, 2/6, 2/6, and 1/6.

Metrics and Comparisons. Fundamental and technical horizons are 30/90 and 5/15 calendar days, respectively. IC and RankIC are the Pearson and Spearman correlations of exposures with subsequent returns; Spread is the mean-return difference between the highest- and lowest-exposure 20% of stocks. Metrics are averaged across scoring dates, then equally across the horizons within each domain. Numerical baselines evaluate individual formulas, covering classic factor categories such as size and value (Fama & French, 1993), profitability and investment (Fama & French, 2015), gross profitability (Novy-Marx, 2013), and momentum (Jegadeesh & Titman, 1993). Appendix D.2 gives the specific definitions. We use three question sets: Initial NL contains the starting questions; Classic NL describes classic numerical attributes in words; Consolidated NL contains questions selected from the factor bank using validation performance. Each set produces an ensemble by averaging each stock's percentile ranks across questions. The final question set and aggregation rule are fixed before testing.

These comparisons answer different questions. Classic NL and its matched formulas test whether the agent scores the property asked about. Initial NL and Consolidated NL compare the starting questions with the result of question development and validation selection. Individual numerical factors provide predictive references for classic attributes. Controlled evidence tests hold the question fixed while changing the records. Repeated runs hold both fixed. They examine, respectively, how scores respond to evidence and how consistently the agent scores the same task.

5.2 Measurement Properties

Formula Agreement. Figure 2 uses six classic factors per domain as measurement references. Fundamental factors cover profitability, investment, valuation, and size. Gross profitability uses assets as its base; operating profitability uses book equity. Negative asset growth indicates slower asset expansion. Earnings-to-price and book-to-market compare earnings and book equity with market value. Negative market capitalization gives higher values to smaller companies. Technical factors cover risk, liquidity, and price trends. Negative realized volatility and negative market beta indicate lower historical volatility and market sensitivity, respectively. Amihud illiquidity measures absolute returns per unit of trading value; higher values indicate lower liquidity. The 6–1 and 12–1 momentum factors measure historical returns excluding the latest month. One-month reversal negates the latest month's return. Detailed definitions appear in Appendix D.2.

For each Classic NL question and its prespecified formula counterpart, we compute the cross-sectional Spearman correlation between stock scores and formula values at each scoring date and average it across dates. This measures whether the two representations produce consistent stock rankings. The full cross-correlation matrices appear in Appendix E.

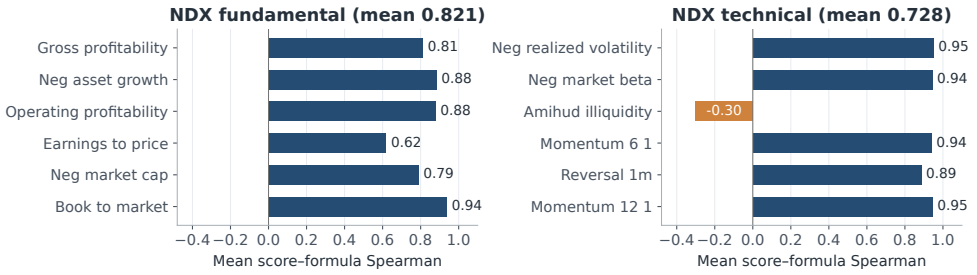

Figure 2: Mean date-wise Spearman agreement between Classic NL scores and matched numerical factors in each NDX domain.

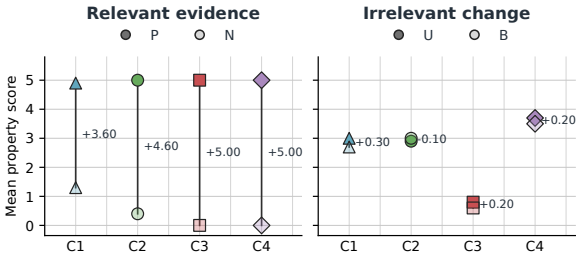

Figure 3: Mean scores over ten executions, with paired P/N and U/B markers and annotated score differences. Lines connect paired conditions.

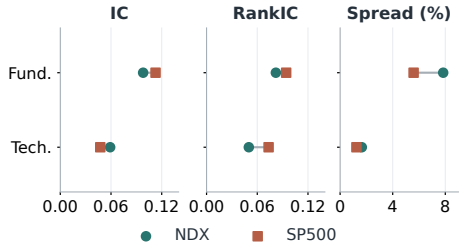

Figure 4: Consolidated NL on NDX and SP500 excluding NDX constituents: fundamental (Fund.) and technical (Tech.).

Mean agreement is 0.821 for fundamentals and 0.728 for technicals (Figure 2). High agreement for book-to-market, momentum, and low volatility indicates that the agent ranks stocks similarly to the corresponding formulas. Amihud illiquidity is the only negatively correlated attribute (-0.300), primarily because some runs assign high scores to the investment benefits of greater liquidity. Overall, these comparisons support the agent's ability to measure several specified properties.

Controlled Evidence. Cases C1–C4 examine cash-flow support, net-margin improvement, up/down-day volume, and drawdown recovery. We keep the question context fixed and compare baseline evidence (B), stronger relevant evidence (P), weaker relevant evidence (N), and an irrelevant change to document identifiers (U). Figure 3 shows mean scores and differences between conditions over repeated runs. Mean U–B differences are close to zero, whereas P–N differences span much of the rating scale. Strict P>B>N holds in 33/40 comparisons matched by run index. These results indicate that scores respond to stronger and weaker relevant evidence while remaining relatively unaffected by irrelevant edits.

Repeated Execution. We execute the initial questions five times on the same NDX validation panels to test consistency in both stock scores and question evaluation. Score ρ, ICC(3,1), and Rank MAE compare stock scores and rankings across runs. Q-RankIC ρ and sign agreement test whether questions retain similar performance rankings and predictive directions when evaluated against fixed validation returns. Appendix D.1 provides definitions and aggregation rules.

Score correlations are 0.824/0.695 and ICC values are 0.862/0.704 for fundamentals/technicals (Table 1). Together with the other metrics, these results indicate a degree of consistency across repeated runs. Fundamentals are more stable than technicals, suggesting that longer-horizon signals and textual information may provide more stable measurement inputs.

5.3 Predictive Performance

Test Results. Table 2 reports fundamental and technical test performance in Panels A and B. Methods use the same return dates and are evaluated on valid observations for each method and horizon.

Table 1: Score and question-ranking stability across five executions of fixed NDX validation inputs.

Domain	Score $\rho \uparrow$	ICC(3,1)↑	Rank MAE↓	Q-RankIC $\rho \uparrow$	Sign agree↑
Fundamental	0.824	0.862	0.110	0.916	88.3%
Technical	0.695	0.704	0.150	0.490	76.7%

Table 2: Historical test performance of individual numerical factors and natural-language ensembles. Spread (%) is averaged equally across dates and horizons. Best and second-best values within each panel and column are bold and underlined, respectively.

Method		**CSI300**			**NDX**		
		IC↑	RankIC↑	Spread↑	IC↑	RankIC↑	Spread↑
Panel A: Fundamental (30/90 days)							
Numeric	Gross profitability	-0.090	-0.064	-0.45	-0.030	-0.052	-5.70
	Neg asset growth	-0.019	-0.006	-3.40	<u>0.090</u>	**0.123**	<u>4.88</u>
	Operating profitability	0.049	0.086	3.01	-0.034	0.045	-4.15
	Earnings to price	0.061	**0.136**	2.08	0.067	0.030	0.38
	Neg market cap	-0.035	-0.016	-0.73	0.003	-0.143	-8.87
	Book to market	0.010	0.101	1.41	0.086	0.079	4.26
Agent-based	Classic NL	0.089	<u>0.121</u>	2.53	-0.038	-0.012	-6.95
	Initial NL	<u>0.094</u>	0.039	**5.63**	0.002	0.000	3.17
	Consolidated NL	**0.102**	0.052	<u>5.61</u>	**0.098**	<u>0.082</u>	**7.84**
Panel B: Technical (5/15 days)							
Numeric	Neg realized volatility	0.021	0.011	-0.18	-0.047	-0.016	0.31
	Neg market beta	0.036	<u>0.051</u>	-0.03	-0.014	0.003	0.21
	Amihud illiquidity	0.042	**0.106**	**1.12**	0.056	0.036	1.18
	Momentum 6 1	-0.092	-0.053	-0.49	0.027	-0.009	0.67
	Reversal 1m	-0.097	-0.107	-2.09	-0.082	-0.068	-2.31
	Momentum 12 1	-0.122	-0.119	-0.79	0.034	-0.004	1.18
Agent-based	Classic NL	-0.042	-0.044	-0.87	-0.040	-0.010	-1.07
	Initial NL	<u>0.057</u>	0.018	<u>0.52</u>	**0.072**	**0.064**	**2.30**
	Consolidated NL	**0.063**	0.043	0.17	<u>0.059</u>	<u>0.050</u>	<u>1.65</u>

Numeric rows are individual formulas; Agent-based rows are question ensembles. The main workflow comparison is Initial NL versus Consolidated NL.

In this historical test, Consolidated NL has higher IC and RankIC than Initial NL in three of four settings. CSI300 fundamental IC/RankIC rises from 0.094/0.039 to 0.102/0.052, and technical IC/RankIC from 0.057/0.018 to 0.063/0.043. NDX fundamentals show the largest difference: IC increases from 0.002 to 0.098, RankIC from 0.000 to 0.082, and Spread from 3.17% to 7.84%. These results indicate that the selected question ensembles can provide return-related signals in these settings, offering evidence of the workflow's usefulness for factor construction.

Changes differ across settings and metrics. NDX technical IC/RankIC/Spread falls from 0.072/0.064/2.30% to 0.059/0.050/1.65%, with Initial NL leading on all three metrics. In CSI300, IC and RankIC rise while Spread falls from 5.63% to 5.61% for fundamentals and from 0.52% to 0.17% for technicals. IC and RankIC assess the score–return relationship across all stocks; Spread compares average returns in the highest- and lowest-scoring groups. An increase in overall correlation therefore need not increase the return difference between these two groups. Classic NL agrees closely with several classic attributes in formula comparisons, yet its technical ensemble has negative test IC and RankIC in both universes, while Initial NL and Consolidated NL are positive.

Universe Transfer. To examine performance beyond NDX, we sample agent tasks from S&P 500 (SP500) stocks excluding NDX constituents, apply the original selected question ensembles, and compare the results with those for NDX in the main table. Figure 4 reports IC, RankIC, and Spread for both domains.

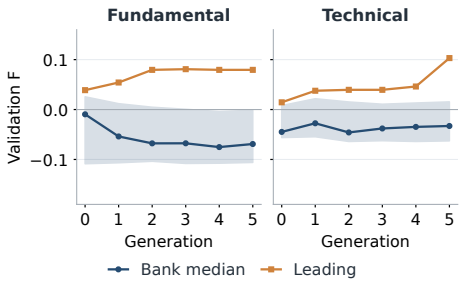

Figure 5: NDX bank and leading-candidate validation fitness, with bank interquartile shading.

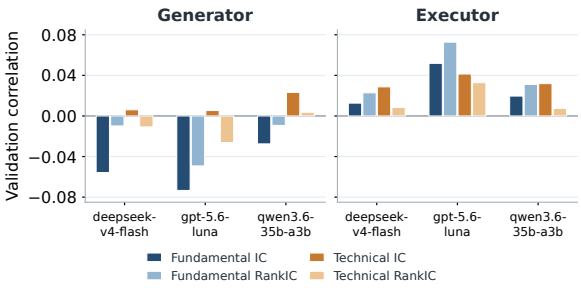

Figure 6: NDX validation IC and RankIC under generator and executor changes for fundamental and technical factors.

On the sampled SP500 agent tasks, the fundamental ensemble has higher IC and RankIC but lower Spread than on NDX. The technical ensemble has higher RankIC but lower IC and Spread. All three metrics are positive in both domains, indicating positive relationships between scores and subsequent returns. Performance broadly comparable to NDX supports the feasibility of reusing natural-language factors across stock sets.

5.4 Search and Sensitivity

Question Evolution. Figure 5 compares the mean validation score of leading candidates with the median and interquartile range of the full factor bank. For NDX fundamentals, leading-candidate scores improve while the bank median falls; for technicals, both improve. Question refinement and exploration expand the candidate pool, and validation-based selection chooses questions for the final ensemble. These trajectories describe candidate accumulation and selection; the frozen-ensemble test results above assess the predictive potential of the complete workflow. Appendix E.3 provides detailed definitions.

Model Sensitivity. We test model sensitivity in two roles. Executors score the same questions. Generators propose new questions from the same evaluation results and reflections. DeepSeek-v4-flash scores all generated questions. Figure 6 summarizes validation IC and RankIC in both NDX domains. The gpt-5.6-luna (OpenAI, 2026) executor achieves the highest RankIC in both domains. In the technical generator comparison, qwen3.6-35b-a3b (Qwen Team, 2026) offspring have the highest RankIC, whereas all three fundamental offspring ensembles have negative RankIC. Model rankings differ between the two roles: generators affect the questions proposed, while executors affect how given questions are measured. Model choice is therefore part of the agent measurement protocol and should remain fixed during its application. Appendix E.3 provides detailed analysis and experimental settings.

6 Conclusion

We introduce *Agentic Factor Engineering*, defining natural-language factors through investment questions and a fixed execution protocol that turns investment ideas into evidence-backed stock-level exposures. Formula comparisons, controlled evidence changes, and repeated execution examine property agreement, responsiveness, and consistency, respectively. These complementary tests support question-based measurement while distinguishing measurement validity from return predictiveness.

Return-guided search demonstrates compatibility with automated factor discovery through candidate expansion and validation-based selection. Selected ensembles achieve higher held-out IC and RankIC than initial ensembles in three of four settings, providing preliminary evidence of predictive potential. Appendix G discusses limitations. As models improve and inference costs fall, natural-language factors could cover more companies and longer evaluation periods. Future work could improve execution efficiency and extend return-guided question evolution to a wider range of investment settings.

REFERENCES

Yakov Amihud. Illiquidity and stock returns: cross-section and time-series effects. *Journal of financial markets*, 5(1):31–56, 2002.

Yifei Chen, Bryan T Kelly, and Dacheng Xiu. Expected returns and large language models. *Available at SSRN 4416687*, 2022.

Chanyeol Choi, Yoon Kim, Yu Yu, Young Cha, V Zach Golkhou, Igor Halperin, Georgios Papaioannou, Minkyu Kim, Zhangyang Wang, Jihoon Kwon, et al. From text to alpha: Can llms track evolving signals in corporate disclosures? *arXiv preprint arXiv:2510.03195*, 2025.

Can Cui, Wei Wang, Meihui Zhang, Gang Chen, Zhaojing Luo, and Beng Chin Ooi. Alphaevolve: A learning framework to discover novel alphas in quantitative investment. In *Proceedings of the 2021 International conference on management of data*, pp. 2208–2216, 2021.

Eugene F Fama and Kenneth R French. Common risk factors in the returns on stocks and bonds. *Journal of financial economics*, 33(1):3–56, 1993.

Eugene F Fama and Kenneth R French. A five-factor asset pricing model. *Journal of financial economics*, 116(1):1–22, 2015.

Chrisantha Fernando, Dylan Sunil Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 13481–13544. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/fernando24a.html.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 34133–34156, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/9156b0f6dfa9bbd18c79cc459ef5d61c-Paper-Conference.pdf.

Jun Han, Shuo Zhang, Wei Li, Yifan Dong, Tu Hu, Yumo Zhu, Xiaomin Yu, Xin Guo, Zhaowei Liu, Kunyi Wang, et al. Quantaalpha: An evolutionary framework for llm-driven alpha mining. *arXiv preprint arXiv:2602.07085*, 2026.

Narasimhan Jegadeesh and Sheridan Titman. Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance*, 48(1):65–91, 1993.

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

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. *Advances in Neural Information Processing Systems*, 38, 2026.

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

Qinhong Lin, Ruitao Feng, Yinglun Feng, Zhenxin Huang, Yukun Chen, Zhongliang Yang, Linna Zhou, Binjie Fei, Jiaqi Liu, and Yu Li. Factorengine: A program-level knowledge-infused factor mining framework for quantitative investment. *arXiv preprint arXiv:2603.16365*, 2026.

Alejandro Lopez-Lira and Yuehua Tang. Can chatgpt forecast stock price movements? return predictability and large language models. *Journal of Financial Economics*, 184:104335, 2026. ISSN 0304-405X. doi: https://doi.org/10.1016/j.jfineco.2026.104335. URL https://www.sciencedirect.com/science/article/pii/S0304405X26001066.

Tim Loughran and Bill McDonald. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65, 2011. doi: https://doi.org/10.1111/j.1540-6261.2010.01625.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2010.01625.x.

Robert Novy-Marx. The other side of value: The gross profitability premium. *Journal of financial economics*, 108(1):1–28, 2013.

OpenAI. GPT-5.6 System Card, July 2026. URL https://deploymentsafety.openai.com/gpt-5-6.

Charidimos Papadakis, Angeliki Dimitriou, Giorgos Filandrianos, Maria Lymperaiou, Konstantinos Thomas, and Giorgos Stamou. Atlas: Adaptive trading with llm agents through dynamic prompt optimization and multi-agent coordination. *arXiv preprint arXiv:2510.15949*, 2025.

Kassiani Papasotiriou, Srijan Sood, Shayleen Reynolds, and Tucker Balch. Ai in investment analysis: Llms for equity stock ratings. In *Proceedings of the 5th ACM International Conference on AI in Finance*, ICAIF '24, pp. 419–427, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400710810. doi: 10.1145/3677052.3698694. URL https://doi.org/10.1145/3677052.3698694.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Patrick E Shrout and Joseph L Fleiss. Intraclass correlations: uses in assessing rater reliability. *Psychological bulletin*, 86(2):420, 1979.

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.

Paul C. Tetlock. Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, 62(3):1139–1168, 2007. doi: https://doi.org/10.1111/j.1540-6261.2007.01232.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2007.01232.x.

Meiyun Wang, Kiyoshi Izumi, and Hiroki Sakaji. LLMFactor: Extracting profitable factors through prompts for explainable stock movement prediction, 2024. URL https://arxiv.org/abs/2406.10811. arXiv preprint arXiv:2406.10811.

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

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, volume 2024, pp. 12028–12068, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Weicheng Ye, Youran Sun, Xingyu Ren, Shunyao Yu, Chugang Yi, and Haizhao Yang. Agonalpha: Autonomous alpha discovery via prompt economy and scalable agentic search. *arXiv preprint arXiv:2608.11250*, 2026.

Jingyang Yi, Jian Yang, Yifei Jin, Yuqi Li, and Jian Li. Alphaschema: Exploring the space of trading semantics for llm-based alpha mining. *arXiv preprint arXiv:2607.26642*, 2026.

Hang Yu, Zifan Zheng, Jeff Z Pan, Tongliang Liu, Zhiyong Wang, and Fengxiang He. Alphamemo: Structured search-process memory for self-evolving alpha mining agents. *arXiv preprint arXiv:2606.20625*, 2026.

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.

Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan W. Suchow, Zhenyu Cui, Rong Liu, Zhaozhuo Xu, Denghui Zhang, Koduvayur Subbalakshmi, Guojun Xiong, Yueru He, Jimin Huang, Dong Li, and Qianqian Xie. Fincon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 137010–137045. Curran Associates, Inc., 2024. doi: 10.52202/079017-4354. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/f7ae4fe91d96f50abc2211f09b6a7e49-Paper-Conference.pdf.

Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Jordan W Suchow, Denghui Zhang, and Khaldoun Khashanah. Finmem: A performance-enhanced llm trading agent with layered memory and character design. *IEEE Transactions on Big Data*, 2025.

Yishuo Yuan, Jiayi Sheng, Sirui Zeng, Jiaqi Wang, and Jiaheng Liu. Alphacrafter: A full-stack multi-agent framework for cross-sectional quantitative trading. *arXiv preprint arXiv:2605.05580*, 2026.

Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, et al. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist. In *Proceedings of the 30th acm sigkdd conference on knowledge discovery and data mining*, pp. 4314–4325, 2024.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*, 2022.

A AI USE STATEMENT

We used large language models to assist with literature search, writing code, and refining the manuscript to improve efficiency, concision, and readability. All methodological design, experiments, visualizations, and substantive manuscript writing were carried out by the authors. All LLM-assisted outputs were reviewed and verified by the authors. In accordance with the ICLR 2027 policy, the authors take full responsibility for the final content of this work.

B DATA ORGANIZATION

Data are organized by stock, information type, and time. Tushare supplies individual stock prices, financial records and other data; the US data pipeline uses yfinance for market-index prices. Textual records include news and corporate announcements. The following describes their storage, date filtering, and retrieval.

Tabular data include price and volume observations, technical indicators, fundamental indicators, financial statements, industry classifications, and index constituents, stored in CSV format. Date filtering uses trading dates or associated disclosure dates, distinguishing disclosure dates from the accounting periods described by the records.

Textual data include news and corporate announcements, stored in Markdown format with date information. A text index summarizes each document so the agent can identify relevant events before reading the full text for supporting and opposing evidence about the property being measured. An LLM generates document summaries from source text and directory summaries from file names and summaries in lower-level indexes.

A data index records file locations and labels. The agent reads it to understand the available materials before locating records relevant to a question. The data index and document summaries serve different purposes. The index helps locate files, while summaries help identify documents relevant to the question. The agent can then read the selected documents in full and check their claims against tabular values. Different questions can therefore use different records through the same data access process.

C AGENT EXECUTION

This appendix gives tool interfaces, prompts, run settings, execution statistics, and a case trace. It also provides the mathematical details of question evolution.

C.1 AGENT PROTOCOL

Architecture and Tools. Each task answers a batch of questions for one stock at one scoring date. Questions share retrieval and analysis, but each has its own score and evidence. The task workspace stores scripts and intermediate results.

Tool results inform the next action. The agent may use a retrieved record to locate further evidence, or inspect a calculation to check units, signs, and reporting periods. If a program fails or its output conflicts with the source records, the agent can edit the script and run it again. The Zscaler trace below shows specific calculation repairs within this process.

The executor agent uses the following tools:

- **Read**: access source records and their context.
- **Write**: save analytical scripts, computed results, and intermediate materials.
- **Edit**: revise scripts or intermediate materials to adjust calculations and correct errors.
- **Execute**: retrieve information, run programs, and inspect results through the command line.
- **Structured Output**: submit per-question answers, scores, and evidence in the required structure.

We implement these interfaces in a custom agent. Other agent frameworks can implement the protocol if they support the required data access, code execution, and outputs.

Input Prompt. CSI300 tasks use relevant tables and text; NDX tasks mainly use structured price, index, and financial records. The following condensed English translation gives the task instructions and scoring rules.

Input Prompt

```
Examine the available data for [company] ([ticker]) and answer
[ordered questions] using evidence available by the specified cutoff
date.

Read the data documentation first and check units and reporting
periods.  Retrieve materials relevant to each question, perform
calculations when needed, and compare supporting and opposing
evidence.  Resolve conflicting information by considering publication
time and source authority.  Keep analysis scripts and intermediate
results in the designated workspace, and record unavailable
information and tool failures.

Answer every question in its original order.  Provide a direct answer,
an independent score from 0 to 5, a confidence rating from 0 to 5,
supporting evidence, counter-evidence, and information gaps.  Evaluate
the property specified by the question and explain how the evidence
supports the assigned score.

Score rubric:  0---clearly negative; 1---mostly negative; 2---weak,
warranting caution; 3---neutral, mixed, or insufficiently directional;
4---mostly positive; 5---clearly positive.

Confidence rubric:  0---almost no basis for a judgment; 1---major data
gaps or very weak evidence; 2---limited evidence; 3---some support
with substantial uncertainty; 4---substantial support; 5---sufficient
evidence consistent across sources.

Summarize the analytical steps, useful findings, and unsuccessful
attempts.  Identify reusable computations where applicable, and
connect each conclusion to its evidence.
```

Output Format. The template below shows the report structure. Question responses follow the input order.

Table 3: Mean report-level execution costs for fundamental and technical factors.

Measure	Fundamental	Technical
Tool calls	27.77	17.57
Cumulative input tokens	832,716	351,658
Cumulative output tokens	23,341	17,362
Peak tokens per call	65,056	43,315

Output Format

```
title: report title
report metadata:
  company name: company being evaluated
  ticker: stock identifier
  data cutoff: information cutoff date
company profile:
  industry: industry classification
  main business: principal products or services
  profile confidence: high / medium / low
question responses: one entry per input question
  question identifier: position in the input list
  question: original question wording
  answer: direct response to the question
  score: integer from 0 to 5
  score rationale: evidence-based justification
  confidence: integer from 0 to 5
  confidence rationale: basis for the confidence rating
  supporting evidence: list of evidence items
  counter-evidence: list of evidence items
  missing information: unresolved information gaps
workflow:
  steps: analytical operations in execution order
  successful approaches: effective analytical practices
  unsuccessful attempts: failures and their implications
evidence item:
  summary: relevant facts and interpretation
  sources: references to supporting materials
  strength: strong / medium / weak
```

Sampling and Execution Settings. The custom executor defaults to a sampling temperature of 0.8 and a nucleus sampling threshold of 1.0, with a maximum of 32,768 completion tokens per model response and a context budget of 262,144 tokens. Reasoning is enabled. Execution is subject to step and time limits. Failed tasks may be retried, and execution records are retained.

C.2 Aggregate Execution Statistics

Table 3 shows that fundamental reports require more tool calls and cumulative input and output tokens than technical reports. Cumulative token counts sum usage over all model interactions for a report. Peak tokens per call is the largest input-plus-output token count in any single model call across that report's execution attempts, including retries. Costs are averaged across reports within each universe and then equally across CSI300 and NDX.

The command-execution tool (Execute) accounts for most tool calls, covering document reading, content retrieval, program execution, and result writing. Command-level statistics provide a closer view of how tool operations support evidence analysis.

Table 4 summarizes command usage by category. Calls/report is the mean number of Execute calls using a command category per report. Execute calls (%) is the percentage of Execute calls using that category; Reports (%) is the percentage of reports using it at least once. Each category is counted at most once per Execute call; Other groups commands occurring no more than once per report on average. Categories can overlap because a single call may combine directory navigation,

Table 4: Command usage across fundamental and technical reports.

Command	Calls/report	Execute calls (%)	Reports (%)
Program execution, retrieval and text processing			
`python`	9.22	46.21	99.52
`grep` / `rg`	4.33	19.63	99.56
`head`	7.68	34.89	99.80
`tail`	2.59	13.47	85.13
`cat`	4.84	22.77	98.21
`sed`	1.25	5.43	64.64
`wc`	1.66	8.30	81.80
Navigation, workspace setup and output			
`ls`	4.45	21.67	100.00
`cd` / `pwd`	20.70	99.27	100.00
`echo` / `printf`	10.06	47.59	100.00
Other	1.84	8.79	70.02

Table 5: Question-level score distributions, self-rated confidence, and evidence reporting across R0–R5.

Rating	Property score (%)	Confidence (%)
0	7.73	0.15
1	27.99	1.01
2	28.02	5.49
3	13.98	21.79
4	17.14	53.73
5	5.13	17.84
Mean rating (0–5)	2.20	3.81
Output item	Answers (%)	
Supporting evidence	99.94	
Counter-evidence	68.86	
Information-gap notes	73.76	

document reading, and program execution. Percentages therefore need not sum to 100% across rows. The command profile shows that report generation routinely combines evidence retrieval, content inspection, and numerical computation.

Figure 7 plots the two tables. Each command share is its mean count per report divided by the sum across categories; it is not Execute-call coverage. Table 5 gives the score and confidence distributions, together with the frequency of evidence-related outputs.

Property scores span the full 0–5 scale, with a mean of 2.20 and 56.02% at ratings 1–2. Self-rated confidence has a mean of 3.81, with 71.57% at ratings 4–5. Property scores concentrate more heavily in the lower range, while confidence concentrates in the upper range. This difference reflects their distinct meanings: the former assesses how well the evidence satisfies the specified attribute, and the latter expresses the agent's certainty in that assessment.

Supporting evidence appears in 99.94% of question responses, opposing evidence in 68.86%, and information-gap notes in 73.76%. These records provide an inspectable basis for numerical scores, documenting supporting facts, contradictory information, and unresolved evidence needs.

C.3 Execution Trace

The following trace shows how the executor agent retrieves records, checks financial data, corrects calculation errors, and answers each question. Related consecutive operations are grouped to show how it calculates measures from source records and checks whether the results support the conditions in a question.

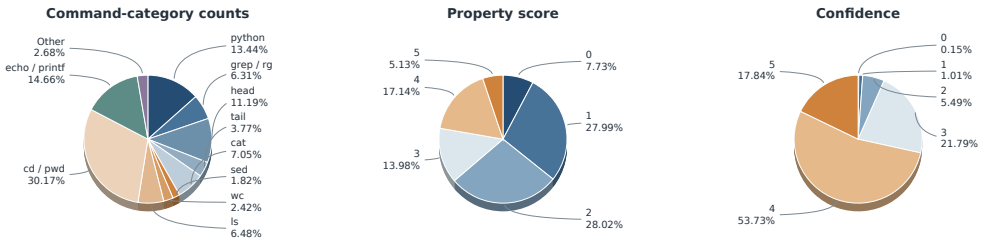

Figure 7: Command-category count shares and distributions of property scores and confidence ratings.

Financial-statement reconciliation and repair. The Zscaler case shows how the executor checks financial records, corrects a sign error in free-cash-flow calculations, and assesses the conditions in each question.

> **Task.** Analyze a batch of questions about Zscaler's profitability, cash conversion, and free cash flow using the available historical evidence.
>
> 1. **Execute, Read: Identify the available evidence.** Read documentation and locate the company, prices, financial indicators, and statements. Check their temporal coverage and note that financial statements are less recent than the price history.
> 2. **Execute: Reconcile periods and accounts.** Extract matching fiscal quarters from the income, cash-flow, and balance-sheet statements. Distinguish quarterly, cumulative, and annual reports; inspect operating cash flow, capital expenditure, receivables, and payables.
> 3. **Write, Execute: Compute financial measures.** Write and run a program for free cash flow, cash conversion, cash-flow coverage, and margin changes. Its first annual free-cash-flow calculation subtracts a negative expenditure amount, overstating free cash flow.
> 4. **Edit, Execute: Correct and verify the calculation.** Check the capital-expenditure sign convention and recalculate free cash flow. The corrected annual free cash flow is positive and has increased year over year. Quarterly operating cash flow is positive, but net income remains negative.
> 5. **Structured Output: Apply each question's conditions.** Distinguish improving net margin from meeting the requested cash-flow coverage condition: positive operating cash flow alongside a net loss does not satisfy the specified positive coverage threshold. Separately, identify gross-margin contraction and cost growth exceeding revenue growth, rather than the requested margin expansion. Return independent answers with scores, confidence ratings, supporting and opposing evidence, and information gaps.

C.4 Evolutionary Search

Objective. For each question, the median RankIC across scoring dates summarizes performance at each return horizon:

$$G_{q,h}(\mathcal{D}) = \mathrm{Median}_{t\in\mathcal{D}}(\mathrm{RankIC}_{q,t,h}), \qquad \mathbf{G}_d(q;\mathcal{D}) = (G_{q,h}(\mathcal{D}))_{h\in\mathcal{H}_d}. \tag{5}$$

Here $\mathcal{D}$ is the development or validation period, and $\mathbf{G}_d$ retains the result for each horizon. During development, signed RankIC statistics and execution evidence guide the reflection and evolution agents in revising questions and choosing parents. The agents consider these materials jointly. Validation selection uses the deterministic rule in Equation 4; reported performance uses the equal-weight means in Equation 15.

For a population-based procedure, write the n_g active candidates at generation g as

$$\mathcal{P}_d^{(g)} = \{q_{d,1}^{(g)}, \ldots, q_{d,n_g}^{(g)}\}, \qquad q_{d,k}^{(g)} \in \mathcal{Q}_d. \tag{6}$$

The definitions below distinguish the active population $\mathcal{P}_d^{(g)}$, the factor bank $\mathcal{A}_d^{(g)}$, and the final question set $\mathcal{C}_d = \mathcal{Q}_d^\star$.

Domain constraints. Questions within a domain use the specified information types and return horizons. The initial questions cover different investment ideas; later generations follow the same domain constraints.

Case Sampling. For each question, valid responses within each source period are divided into return quantiles. Equal numbers of cases are randomly sampled from each group to cover different return outcomes. Sorting uses the mean return over the selected horizons by default, using available horizons when some are missing. Each case retains the original answer, score, confidence, and returns so the reflection agent can compare judgments with outcomes.

Reflection. The main text describes the reflection outputs. Here $\mathcal{R}_d^{(g)}$ denotes the reflections for generation g, computed from questions, their evaluations, and execution evidence:

$$\mathcal{R}_d^{(g)} = \mathrm{Reflect}\Big(\Big\{q, \mathbf{G}_d(q; \mathcal{T}_{\mathrm{dev}}), \mathcal{E}(q) : q \in \mathcal{P}_d^{(g)}\Big\}\Big), \tag{7}$$

$\mathrm{Reflect}$ is the reflection agent. $\mathbf{G}_d(q; \mathcal{T}_{\mathrm{dev}})$ contains the question's development-period RankIC summaries for each horizon, and $\mathcal{E}(q)$ contains its execution evidence.

In practice, the reflection agent reads the original question alongside execution rationales, statistics across dates, and company cases. It compares supporting evidence, opposing evidence, and unresolved points to assess possible economic explanations for the return relationship and identify conditions to adjust. The original question lets recommendations refer to specific wording. The execution records let them account for the evidence the agent actually used. The condensed translation below describes reflection using only the supplied materials, without tool calls.

Reflection Instructions

```
Assess whether the question's scores relate to subsequent returns
using the evaluation summary and responses sampled across return
groups.  Distinguish relationship strength from direction:  a
consistent negative relationship can also be informative.
Compare answers, evidence quality, scores, and returns across all
supplied cases.  Identify supporting evidence, counterexamples, and
missing evidence.  Retain a small set of representative case
observations.
Propose concrete question changes, combinations, splits, or
conditions, with reasons and risks.  Revise the question while keeping
the execution protocol fixed.
```

The reflection summarizes the question's relationship with returns, its direction, and the confidence in that assessment. Representative cases support the proposed changes.

Evolutionary operators. Let Elite, Cross, Mutate, and $\mathrm{Explore}$ denote the four operators defined in the main text. The next generation is

$$\mathcal{P}_d^{(g+1)} = \mathrm{Elite}\Big(\mathcal{P}_d^{(g)}\Big) \cup \mathrm{Cross}\Big(\mathcal{P}_d^{(g)}, \mathcal{R}_d^{(g)}\Big) \cup \mathrm{Mutate}\Big(\mathcal{P}_d^{(g)}, \mathcal{R}_d^{(g)}\Big) \cup \mathrm{Explore}\Big(\mathcal{Q}_d, \mathcal{R}_d^{(g)}\Big). \tag{8}$$

The union in Equation 8 combines candidates from the four operators into the next population. The evolution agent first writes a population analysis, identifying investment ideas to retain, evidence to combine, wording to adjust, and directions to explore. It then uses this analysis to write questions and describe their origins. Retaining the analysis alongside the questions allows later inspection of how population-level judgments led to particular changes.

The evolution agent selects parents using reflections on the current questions and may also draw on questions and reflection summaries from other groups. Elite preservation keeps a question unchanged, mutation modifies one parent, crossover combines two distinct parents, and exploration introduces new questions.

The condensed translation below covers analysis and candidate generation; neither stage calls tools.

Evolution Instructions

```
Analyze the current questions using their evaluation statistics and
reflections.  Identify ideas to retain, evidence to combine, wording
to change, and uncovered directions.  Plan the required number of
candidates for each enabled operator.
Formulate candidates so higher scores indicate more positive expected
returns.  A consistently negative signal may be revised by reversing
the question's meaning.  Keep one main signal and at most one
necessary confirmation condition per question; place detailed
reasoning in the analysis.
Turn the analysis into concise questions without adding further
conditions.  Explain the origin of each question and the reasons for
any changes.
```

The generated candidates are checked for the required counts, parent relationships, and duplicate questions.

Factor Bank and Selection. After generation g, the factor bank is

$$\mathcal{A}_d^{(g)} = \mathrm{Unique}\left(\bigcup_{j=0}^{g} \mathcal{P}_d^{(j)}\right), \tag{9}$$

Unique merges questions with identical normalized text and links each candidate to its execution, evaluation, and reflection records.

Using $F_{\mathrm{val},d}$ from the main text, we select the top K_d questions from the bank:

$$\mathcal{Q}_d^{\star} = \mathrm{TopK}_{q \in \mathcal{A}_d^{(g)}}\left(F_{\mathrm{val},d}(q); K_d\right). \tag{10}$$

D METRIC DETAILS

This appendix provides detailed evaluation-metric definitions and explains the variables, orientation, and economic meaning of each classic numerical factor.

D.1 EVALUATION METRICS

A numerical factor maps structured features $u_{i,t} \in \mathbb{R}^p$ to an exposure through a deterministic formula f_ω, where ω specifies its parameters or operations:

$$a_{i,t}^{(f)} = f_\omega(u_{i,t}), \qquad f_\omega : \mathbb{R}^p \to \mathbb{R}. \tag{11}$$

Predictive Evaluation. Let $e_{i,t}$ denote the exposure being evaluated: $s_{i,t}^{(q)}$ for a single question, $a_{i,t}^{(f)}$ for a numerical factor, or $S_{i,t}^{(d)}$ for an ensemble. For each scoring date and horizon, let $\mathcal{V}_{e,t,h} \subseteq \mathcal{U}_t$ contain stocks with both a valid exposure and a valid subsequent return. All three metrics use this same method- and horizon-specific set.

Predictive Metrics. The information coefficient (IC) measures the cross-sectional linear association between factor exposure and subsequent return:

$$\mathrm{IC}_{t,h}(e) = \mathrm{Corr}_{i \in \mathcal{V}_{e,t,h}}(e_{i,t}, r_{i,t,h}). \tag{12}$$

RankIC measures agreement between their cross-sectional orderings:

$$\mathrm{RankIC}_{t,h}(e) = \mathrm{Corr}_{i \in \mathcal{V}_{e,t,h}}\big(\mathrm{rank}_{\mathcal{V}_{e,t,h}}(e_{i,t}), \mathrm{rank}_{\mathcal{V}_{e,t,h}}(r_{i,t,h})\big). \tag{13}$$

Corr denotes Pearson correlation, so the second expression is Spearman correlation. Positive RankIC means that higher-scored stocks tend to rank higher in subsequent returns. For an individual question, write $\mathrm{RankIC}_{q,t,h} = \mathrm{RankIC}_{t,h}(s^{(q)})$. Single-question IC uses the raw score s, and ensemble IC uses the aggregate exposure S. The normalized score z is used to construct ensembles, whereas RankIC ranks exposures within the corresponding valid evaluation set.

Spread compares the mean returns of the highest- and lowest-exposure 20% of stocks:

$$\mathrm{Spread}_{t,h}(e) = \frac{\sum_{i \in \mathcal{V}_{e,t,h}} w_{i,t,h}^{+}(e)\, r_{i,t,h}}{\sum_{i \in \mathcal{V}_{e,t,h}} w_{i,t,h}^{+}(e)} - \frac{\sum_{i \in \mathcal{V}_{e,t,h}} w_{i,t,h}^{-}(e)\, r_{i,t,h}}{\sum_{i \in \mathcal{V}_{e,t,h}} w_{i,t,h}^{-}(e)}. \tag{14}$$

The weights $w^+, w^- \in [0,1]$ assign membership to the top and bottom groups, respectively. Ties at group boundaries receive fractional membership so each side has total weight $0.2|\mathcal{V}_{e,t,h}|$. IC and RankIC characterize association across the full cross-section, whereas Spread captures the return difference between its exposure extremes.

Performance Summary. For an evaluation period $\mathcal{D} \subseteq \mathcal{T}$, we compute each metric at every scoring date and horizon and summarize it as

$$\overline{M}_d(e; \mathcal{D}) = \frac{1}{|\mathcal{H}_d|} \sum_{h \in \mathcal{H}_d} \frac{1}{|\mathcal{D}|} \sum_{t \in \mathcal{D}} M_{t,h}(e), \qquad M \in \{\mathrm{IC}, \mathrm{RankIC}, \mathrm{Spread}\}. \tag{15}$$

We first average equally across scoring dates and then across domain-specific horizons. Cross-sectional metrics are computed separately at each date. Longer return windows can overlap across adjacent scoring dates, so the summary describes average performance over the evaluated dates. Spread is the equal-weight mean of horizon-specific return differences, rather than a single-horizon or annualized portfolio return.

Measurement Agreement. When question q has a prescribed formula counterpart $f(q)$, let $\mathcal{W}_{q,t}$ be the subset of $\mathcal{U}_t$ with valid values for both measurements. Define

$$\mathrm{Fidelity}_{q,t} = \mathrm{Spearman}_{i \in \mathcal{W}_{q,t}}\left(s_{i,t}^{(q)}, a_{i,t}^{(f(q))}\right). \tag{16}$$

This metric compares the stock rankings from a question and its matched formula on the same date. Higher values indicate closer rankings.

Repeated Execution. Pairwise comparisons cover all ten pairs of runs. We report medians for score correlation, ICC, and question-ranking correlation, and means for rank error and sign agreement.

Repeated-execution evaluation distinguishes stock-level measurement consistency from question-level predictive-performance stability. For stock-level measurement, fix a question and scoring date and let $x_{i,r}$ denote the raw score for stock i in run r, with n commonly valid stocks and k runs. Score correlation is the Spearman correlation between the stock-score vectors from two runs, measuring cross-sectional ranking consistency. The mean absolute error in percentile ranks is

$$p_{i,r} = \frac{\mathrm{rank}(x_{i,r}) - \frac{1}{2}}{n}, \qquad \mathrm{MAE}_{r,s} = \frac{1}{n}\sum_{i=1}^{n} |p_{i,r} - p_{i,s}|. \tag{17}$$

Ranks start at 1, with average ranks for ties. Rank MAE measures changes in a stock's percentile position between runs, rather than differences on the raw 0–5 score scale. Lower values indicate smaller changes in relative position. Score correlations are summarized using medians as specified above.

ICC(3,1) is the single-measure consistency coefficient under a two-way mixed-effects model. Let $\bar{x}_{i,\cdot}$, $\bar{x}_{\cdot,r}$, and $\bar{x}_{\cdot,\cdot}$ denote the across-run mean for stock i, the across-stock mean for run r, and the grand mean, respectively. The stock mean square, residual mean square, and ICC are

$$\mathrm{MS}_S = \frac{k}{n-1}\sum_{i}(\bar{x}_{i,\cdot} - \bar{x}_{\cdot,\cdot})^2, \tag{18}$$

$$\mathrm{MS}_E = \frac{\sum_{i,r}(x_{i,r} - \bar{x}_{i,\cdot} - \bar{x}_{\cdot,r} + \bar{x}_{\cdot,\cdot})^2}{(n-1)(k-1)}, \tag{19}$$

$$\mathrm{ICC}(3,1) = \frac{\mathrm{MS}_S - \mathrm{MS}_E}{\mathrm{MS}_S + (k-1)\mathrm{MS}_E}. \tag{20}$$

ICC is computed from raw scores for each question and date and then aggregated. It removes systematic differences in mean scores between runs to assess how reliably a single run distinguishes stocks. It measures single-run consistency, rather than absolute agreement, which also counts these mean shifts, or the reliability of an average over repeated measurements (Shrout & Fleiss, 1979).

Question-level stability assesses whether different runs rank questions similarly by predictive performance. For each question in each run, let $R_{q,r}$ be RankIC averaged across the designated dates and horizons. Question-RankIC correlation compares the rankings of these question-quality vectors between runs. For m questions, sign agreement is

$$\mathrm{SignAgree} = \frac{1}{m\binom{k}{2}}\sum_{r<s}\sum_{q=1}^{m}\mathbf{1}[\mathrm{sign}(R_{q,r}) = \mathrm{sign}(R_{q,s})]\,. \tag{21}$$

Sign agreement measures how often a question keeps the same predictive direction across runs. Question-ranking correlation tests whether different runs order questions similarly by predictive performance.

D.2 Factor Definitions

Fundamental Factors. Let A denote total assets, B book equity attributable to common shareholders, and M equity market capitalization. Let GP^{TTM}, OP^{TTM}, and NI^{TTM} denote trailing-twelve-month gross profit, operating profit, and net income attributable to common shareholders, respectively. Assets and book equity at the corresponding prior-year date are A^- and B^-. Their averages with the current balances are

$$\bar{A} = \frac{A + A^-}{2}, \qquad \bar{B} = \frac{B + B^-}{2}.$$

The following definitions specify each factor and its economic interpretation. Numerators and denominators use consistent currencies and accounting conventions, and each factor's orientation follows its target attribute.

1. *Gross profitability.*
$$f_{\mathrm{GP}} = \frac{GP^{\mathrm{TTM}}}{\bar{A}}.$$
Gross profit is revenue less the corresponding cost of goods sold. Gross profitability measures gross profit per unit of average assets, with $\bar{A} > 0$. Higher values indicate greater ability to generate gross profit from assets through production and sales, before period expenses, interest, and income taxes.

2. *Negative asset growth.*
$$f_{\mathrm{AG}} = -\left(\frac{A}{A^{-}} - 1\right).$$
The expression $A/A^{-} - 1$ is year-over-year total asset growth, with $A^{-} > 0$. Negation assigns higher values to slower asset expansion or greater contraction, and lower values to faster expansion. The factor measures asset expansion from balance-sheet changes.

3. *Operating profitability.*
$$f_{\mathrm{OP}} = \frac{OP^{\mathrm{TTM}}}{\bar{B}}.$$
Operating profitability measures operating profit per unit of average book equity, with $\bar{B} > 0$. Higher values indicate greater operating profitability relative to shareholders' book capital. Operating profit reflects earnings from operating activities under the applicable accounting definition. This factor and gross profitability capture different levels of profitability: they use operating profit and gross profit, respectively, and normalize by book equity and assets.

4. *Earnings-to-price.*
$$f_{\mathrm{EP}} = \frac{NI^{\mathrm{TTM}}}{M}.$$
Earnings-to-price, also called earnings yield, measures trailing-twelve-month net income per unit of equity market capitalization, with $M > 0$. Under consistent earnings and market-value definitions and nonzero net income, it is the reciprocal of the TTM price-to-earnings ratio. For profitable companies, higher values indicate lower market valuations relative to current earnings; for loss-making companies, the ratio is negative.

5. *Negative market capitalization.*
$$f_{\mathrm{Size}} = -M, \qquad M = P\,N.$$
Here P is the stock price and N the number of outstanding shares measured on a consistent basis, so M is equity market capitalization. Negating positive market capitalization assigns higher values to smaller companies.

6. *Book-to-market.*
$$f_{\mathrm{BM}} = \frac{B}{M}.$$
Book-to-market measures book equity per unit of equity market capitalization, with $M > 0$. Under consistent equity and market-value definitions and nonzero book equity, it is the reciprocal of the price-to-book ratio. For companies with positive book equity, higher values indicate lower market valuations relative to book net assets, while lower values indicate a larger valuation premium.

Technical Factors. Let W_t be the wealth index at the scoring date, representing the value of a unit investment accumulated under a consistent return convention. Let r_τ denote historical daily stock returns, r_τ^m corresponding market-benchmark returns, and V_τ trading value. In the formulas below, $t - j$ denotes j trading observations before t, rather than j calendar days; 21, 126, and 252 trading observations approximate one month, six months, and one year, respectively.

1. *Negative realized volatility.*
$$f_{\mathrm{Vol}} = -\sqrt{\frac{1}{251}\sum_{j=0}^{251}(r_{t-j} - \bar{r})^2}, \qquad \bar{r} = \frac{1}{252}\sum_{j=0}^{251} r_{t-j}.$$

The mean $\bar{r}$ is computed over the past 252 trading observations, and the quantity under the square root is the sample variance with denominator 251. Negating the standard deviation assigns higher values to lower historical total volatility. The measure captures dispersion of daily returns around their mean, including both market-wide and stock-specific variation, and treats upward and downward deviations symmetrically.

2. *Negative market beta.*

$$f_\beta = -\frac{\mathrm{Cov}_{\tau\in\mathcal{J}_t}(r_\tau, r_\tau^m)}{\mathrm{Var}_{\tau\in\mathcal{J}_t}(r_\tau^m)}.$$

The set $\mathcal{J}_t$ contains dates over approximately the past year with matched stock and market returns. Covariance and variance are computed on this set, with positive market-return variance required. Their ratio is the slope from regressing stock returns on market returns, measuring sensitivity to common market movements. Negation assigns higher values to lower signed betas. The beta sign is retained, so stocks with negative betas can receive higher factor values.

3. *Amihud illiquidity.*

$$f_{\mathrm{ILLIQ}} = \frac{1}{|\mathcal{L}_t|}\sum_{\tau\in\mathcal{L}_t}\frac{|r_\tau|}{V_\tau}.$$

The nonempty set $\mathcal{L}_t$ contains dates among the most recent 20 trading observations with valid daily returns and positive trading value. Amihud illiquidity averages the daily ratio of absolute return to trading value as a proxy for price movement per unit of trading value (Amihud, 2002). Higher values indicate larger price changes per unit traded and hence lower liquidity. Daily ratios receive equal weight. Because the numerical scale depends on the return and monetary units, cross-stock comparisons use consistent units.

4. *6–1 momentum.*

$$f_{\mathrm{Mom6}} = \frac{W_{t-21}}{W_{t-126}} - 1.$$

The 6–1 momentum factor measures cumulative returns from approximately six months to one month before the scoring date. The wealth indices W_{t-126} and W_{t-21} define the beginning and end of this interval, with a positive starting value required. Higher values indicate stronger historical returns over this intermediate horizon. Skipping the most recent month emphasizes earlier price trends and reduces direct overlap with the short-term reversal window.

5. *One-month reversal.*

$$f_{\mathrm{Rev1}} = -\left(\frac{W_t}{W_{t-21}} - 1\right).$$

One-month reversal negates cumulative returns over approximately the most recent 21 trading observations, with $W_{t-21} > 0$. Higher values correspond to stocks with lower recent returns, and lower values to stronger recent performers.

6. *12–1 momentum.*

$$f_{\mathrm{Mom12}} = \frac{W_{t-21}}{W_{t-252}} - 1.$$

The 12–1 momentum factor measures cumulative returns from approximately one year to one month before the scoring date, with $W_{t-252} > 0$. Higher values indicate stronger returns over this longer historical interval. It shares the endpoint of 6–1 momentum and likewise skips the most recent month, but uses a longer lookback to capture price trends at a different timescale.

Together, these factors provide numerical references for the classic attributes specified by the natural-language questions, covering profitability, expansion, valuation, size, risk, liquidity, and historical return trends.

E SUPPLEMENTARY RESULTS

This appendix provides detailed results on score–formula agreement, ensemble prediction, question evolution, and model sensitivity.

E.1 MEASUREMENT RESULTS

Both panels use NDX stocks. Rows retain the original Classic NL question identifiers, and columns F1–F6 follow the six numerical baselines for the corresponding domain in the main table. Each cell is the stock-level Spearman correlation computed at each scoring date and then averaged across dates. Factor definitions are given in Appendix D.2.

Beyond the matched question–formula comparisons in the main text, Figure 8 shows relationships between different properties in its off-diagonal entries. For example, momentum questions correlate strongly with formulas using different lookback windows, suggesting that they capture similar price trends. These comparisons help explain what each question measures and how it relates to other factors.

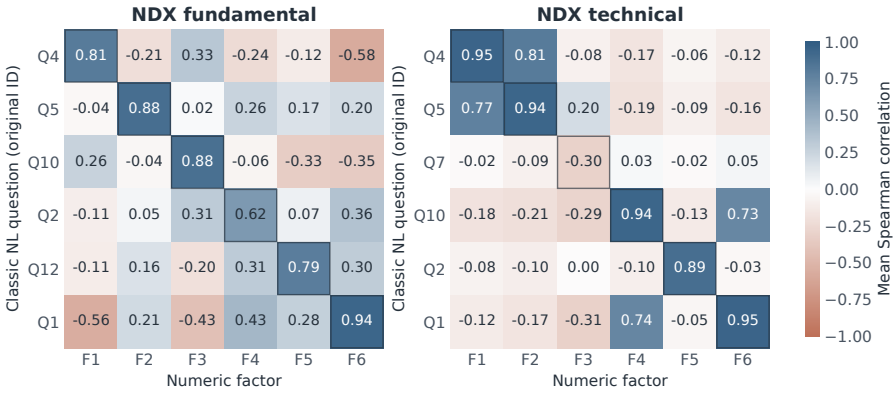

Figure 8: Mean date-wise Spearman correlations between Classic NL questions and numerical factors in each NDX domain.

E.2 PREDICTIVE RESULTS

The monthly curves show validation and test IC and RankIC for the final CSI300 and NDX question ensembles, smoothed using trailing three-month means computed separately within each stage.

Both fundamental ensembles retain positive IC and RankIC throughout the displayed smoothed test curves, with CSI300 fundamentals increasing over the test months. Technical performance varies more, and NDX technical performance rises before falling during testing. The monthly curves complement stage averages by showing how predictive performance changes over time.

E.3 SEARCH AND SENSITIVITY

Factor Bank. In Figure 5, leading candidates are the six questions with the highest validation fitness F, and their curve reports the mean of these values. The bank curve and shading report the median and interquartile range across all distinct candidates. Fitness F is the smaller of the two horizon-specific medians of monthly RankIC.

From the initial to the final generation, mean leading-candidate fitness changes from 0.0388 to 0.0795 for NDX fundamentals, while the bank median changes from -0.0095 to -0.0691. The corresponding technical values change from 0.0145 to 0.1031 and from -0.0448 to -0.0329, respectively.

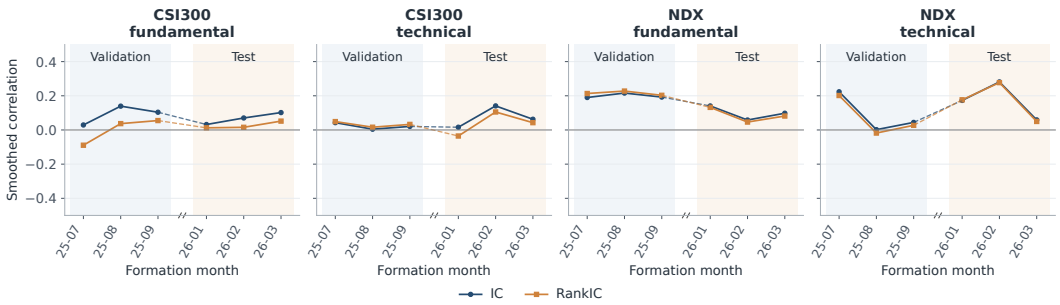

Figure 9: Validation and test IC/RankIC of final ensembles, with stage-wise three-month smoothing and dashed links across omitted months.

Table 6: Median fitness increments and positive-increment rates by evolutionary operator on NDX validation data.

	Fundamental		Technical	
Operator	Median $\Delta F \uparrow$	$\Delta F > 0 \uparrow$	Median $\Delta F \uparrow$	$\Delta F > 0 \uparrow$
Elite	0.036	70.0%	-0.043	20.0%
Crossover	-0.001	45.0%	0.005	55.0%
Mutation	0.028	65.0%	-0.005	45.0%
Exploration	0.019	50.0%	0.035	70.0%

Evolutionary Operators. To compare candidates produced by different operators, we aggregate offspring evaluations across the two NDX domains. Table 6 reports the median validation-fitness increment relative to each offspring's reference and the proportion of positive increments.

Elite preservation and mutation use the declared parent's validation F as the reference; crossover uses the mean validation F of its two parents; exploration uses the median validation F of the preceding population. For fundamentals, Mutation has a median validation increment of 0.0284, with 65% of offspring outperforming their parents. For technicals, Exploration shows the strongest increment, with a median of 0.0346 and a positive-increment rate of 70%. Mutation has a median increment of -0.0046 in technicals, while Crossover is near zero in both domains. In these runs, local revisions in fundamentals and new ideas in technicals produce relatively many candidates that outperform their references. Operator performance therefore differs between domains.

Elite preserves the same question text, so its increments reflect evaluation changes after re-execution rather than changes due to question revision. It provides a reference for evaluation variation under fixed wording. The table describes observed candidate changes relative to their references; test evaluation of the frozen ensembles assesses the predictive performance of the complete workflow.

Generation-wise Increments. The operator curves report each generation's median validation IC increment relative to the operator-specific reference, complementing the aggregate validation-fitness comparison in Table 6.

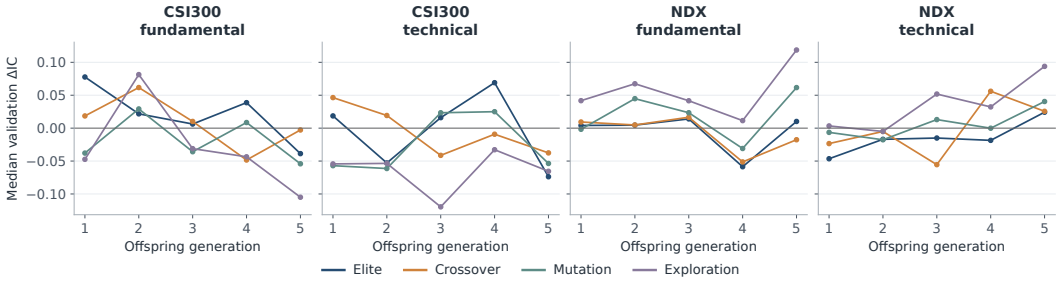

Figure 10: Median offspring increments in validation IC by operator and generation across CSI300 and NDX.

Exploration produces positive IC increments in every generation for NDX fundamentals, but consistently negative increments for CSI300 technicals. Crossover and reflection mutation vary across generations. Thus, the better-performing operators differ across universes, domains, and generations. Together with the aggregate statistics, these curves show which improvements recur across generations and which are concentrated in particular generations.

Model Sensitivity. Figure 6 follows the metrics and aggregation rules in Section 5.1.

Beyond the leading models discussed in the main text, the qwen3.6-35b-a3b executor lies between deepseek-v4-flash and gpt-5.6-luna on fundamental IC and RankIC and technical IC. Its technical RankIC is slightly below deepseek-v4-flash. Its relative performance therefore depends on the domain and metric.

F CASE STUDY

We present one fundamental and one technical case to show how the agent uses evidence to answer questions and assign scores. The final responses are condensed below, preserving the questions' meaning, main evidence, and scores. The cases also show how the agent assesses questions with several conditions.

F.1 FUNDAMENTAL CASE

Capacity entering operation—Huali. The Huali case examines whether new business activity is beginning to deliver results. Factory openings show that capacity has entered operation; shipments to new customers show actual deliveries; and higher shipment volumes and revenue provide further support. These observations indicate that planned expansion has become operating activity. The response also records margin pressure from factory ramp-up and uncertainty about the new capacity's eventual profitability. The case shows how the agent combines disclosed events and operating data to assess whether new business is delivering results.

> **Context.** Huali (300979).
>
> **Question.** Are the company's new products, projects, or added capacity beginning to deliver observable results?
>
> **Answer and evidence.** Yes. Investor-relations disclosures describe newly operating factories and production shipments to new customers. The interim operating results also show higher shipment volumes and revenue. Together, these observations support the transition from planned expansion to operating activity.
>
> **Counter-evidence and uncertainty.** New factories are still ramping up, which can weigh on margins. Their eventual profitability and ramp-up progress remain to be established in subsequent reports.
>
> **Output.** Property score: 5/5. Confidence: 5/5.

F.2 TECHNICAL CASE

Lower-volume pullback with support—COSCO Shipping. The COSCO Shipping Holdings case examines two conditions: lower volume during a pullback and prices remaining above important support levels. Volume below that of preceding breakout-area sessions supports the first condition. A pullback low above the earlier price platform and moving average supports the second. These observations support a positive assessment of the price–volume pattern, while large-order net outflows later in the observed historical window suggest continued selling pressure. The property score of 5 describes the pattern; self-rated confidence of 4 describes the agent's certainty. The case shows how separate conditions are checked and opposing evidence is retained in the score explanation.

> **Context.** COSCO Shipping Holdings (601919).
>
> **Question.** Does the stock pull back on lower volume while holding important price levels?
>
> **Answer and evidence.** Yes. Within the observed historical window, the stock pulls back on lower volume than on the preceding breakout-area sessions. The pullback low remains above the earlier price platform and the reference moving average, and the stock subsequently resumes its advance.
>
> **Counter-evidence and uncertainty.** Later in the same historical window, net outflows from large orders indicate continued selling pressure. The answer combines a positive assessment of the observed price–volume pattern with qualified confidence.
>
> **Output.** Property score: 5/5. Confidence: 4/5.

G LIMITATIONS

Natural-language factors can use records from multiple sources and explain what they measure and why each score is assigned. Retrieval, model inference, and tool calls also add costs, which grow with the number of stocks, scoring dates, and candidate questions. These costs constrain the evaluation period in this study, so the predictive results provide evidence of feasibility and potential. Repeated runs and model comparisons show that execution settings affect scores. Reusing a question therefore requires recording the model, prompts, tools, and scoring rubric. As models improve and inference costs per unit fall, we expect this approach to become more practical and easier to use in larger studies.