# DASH: DIAGNOSTIC-GUIDED QUALITY-DIVERSITY SEARCH FOR FINANCIAL ML AGENTS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Automated LLM research agents iteratively propose, evaluate, and refine machine-learning configurations using empirical feedback. In financial forecasting, low signal-to-noise ratios and nonstationarity make apparent improvements uncertain and potentially transient, allowing search to reinforce spurious or regime-specific gains. We introduce DASH (Diagnostic Agentic Search with Hypotheses), a diagnostic-guided search framework that evaluates configurations across temporal regimes to construct condition-specific, multi-axis behavioral profiles. Before refining a candidate, the agent formulates hypotheses with pre-specified expected outcomes and excludes those refuted by computed evidence. A quality-diversity archive organizes evaluated states by behavioral descriptors while separately accounting for predictive strength and redundancy when retaining and selecting alternatives. Empirically, DASH achieves strong held-out OOD predictive and portfolio performance under matched search budgets. In joint factor–model search mode, it improves IC over the strongest evaluated agentic baseline by 8.7% on NASDAQ-100 and 38.2% on S&P 500, while achieving the highest portfolio IR on both markets. Factor-only analyses show a high concentration of strong OOD factors with low predictive redundancy in the discovered factor pools. In a separate NASDAQ-100 ablation, replacing externally computed evidence with LLM-only judgments consistently degrades predictive performance across two backbones.

## 1 INTRODUCTION

Large language models have enabled autonomous research agents that iteratively propose machine-learning configurations, evaluate them, and refine subsequent proposals based on empirical feedback (Jiang et al., 2025; Yang et al., 2025). Evolutionary approaches such as AlphaEvolve extend this paradigm with population-based search, rich multi-objective evaluation, and diversity-preserving mechanisms (Novikov et al., 2025). These systems have demonstrated strong results across a range of optimization and program-synthesis tasks. Their effectiveness, however, depends on how reliably evaluation feedback distinguishes genuine improvements from spurious ones—a property that cannot be taken for granted in every domain.

Financial forecasting presents a particularly challenging setting for iterative search. Two related but distinct properties of financial data work against the feedback loop. First, predictive signals are weak relative to noise, making candidate comparisons uncertain: the configuration with the highest validation performance may partly owe its advantage to sampling variation rather than a real predictive relationship (Gu et al., 2020). Repeated selection on such noisy measurements can overfit the selection criterion (Cawley & Talbot, 2010; Bailey et al., 2017). Second, market conditions change over time—volatility regimes shift and the relationships between predictors and returns evolve (López de Prado, 2018). A historical advantage may be genuine within the historical periods used for search yet fail to persist in subsequent markets. Noise makes it difficult to identify a real improvement; nonstationarity makes it difficult to know whether a real improvement will last.

These difficulties can distort not only final evaluation but also the search trajectory. A favorable aggregate score may lead the agent to generate further variants in the same direction, even when the apparent advantage is accidental or concentrated in particular market conditions. For example, an improved average information coefficient (IC) may conceal deterioration during volatile periods. This motivates retaining diverse candidates rather than committing the search to a single potentially fragile strategy. However, population diversity offers limited protection when members make highly



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.

correlated predictions: superficially different feature expressions that produce similar forecasts still share the same vulnerabilities.

Our central idea is to *diagnose the conditions under which a predictor's historical performance breaks down before extending its search trajectory*. DASH (Diagnostic Agentic Search with Hypotheses) evaluates each configuration on shared splits and sampled diagnostic windows, producing a condition-specific, multi-axis behavioral profile. Before refinement, the agent proposes hypotheses with pre-specified metric requests and expected outcomes; code computes the corresponding evidence, and refuted hypotheses are excluded before edits are proposed. This mechanism subjects the agent's rationale for a proposed edit to an evidential consistency check against observed behavior, making it harder for plausible-sounding but ungrounded narratives to drive the search. Quality-diversity search complements this process by preserving high-quality alternatives using behavioral descriptors, allowing the search to revisit distinct predictive strategies.

Our contributions are as follows:

- **Diagnostic-guided hypothesis refinement.** We introduce a refinement mechanism in which temporal and behavioral diagnostics identify condition-specific failure modes. The agent formulates explanatory hypotheses and specifies expected metric outcomes before the requested computations are executed. It then assesses these hypotheses against *externally computed evidence* and excludes refuted hypotheses before proposing configuration edits.
- **Quality-diversity archive.** We design an archive that organizes candidates by three behavioral descriptors while using predictive correlation and strength for admission and parent selection, retaining behaviorally distinct alternatives even when their summary descriptors overlap.
- **Evaluation across search modes and equity universes.** Experiments in joint factor–model and factor-only search show strong forward-test performance under matched budgets. Factor-pool analyses assess predictive redundancy, and ablations across two LLM backbones support the value of externally computed evidence for refinement.

## 2 Related Work

**LLM agents for ML and evolutionary search.** LLM-based research agents increasingly automate ML workflows through iterative code search. AIDE (Jiang et al., 2025), SELA (Chi et al., 2024), and I-MCTS (Liang et al., 2026) use validation-guided tree search, while ML-Master (Liu et al., 2025) and its successor (Zhu et al., 2026) incorporate accumulated experience and memory. AIRA (Toledo et al., 2025) systematically studies search policies and refinement operators, highlighting the generalization gap between validation-based selection and held-out performance. Within evolutionary and quality-diversity search, AlphaEvolve (Novikov et al., 2025) supports multiple evaluation metrics and a database inspired by MAP-Elites; its published evaluation cascade assesses newly generated programs. DASH complements this paradigm by checking hypotheses about the current parent's failure modes against requested measurements and pre-committed expectations before editing. Its archive considers prediction redundancy in admission, while diagnostic cells separately organize behavioral alternatives.

**LLM agents for quantitative finance.** LLM-based quantitative research agents increasingly automate alpha discovery and forecasting. A line of work focuses specifically on factor mining: AlphaAgent (Tang et al., 2025), FactorMiner (Wang et al., 2026), and QuantaAlpha (Han et al., 2026) iteratively generate and refine predictive factors using structural constraints, accumulated search experience, or evolutionary mechanisms. Other systems consider a broader search space; R&D-Agent (Li et al., 2025), for example, jointly explores factor construction and predictive models. Despite different search strategies, these methods primarily use backtest or validation metrics as feedback for proposing and selecting candidates. AlphaBench (Luo et al., 2026) further shows that LLMs can be unreliable zero-shot judges of factor quality. Our method also considers a general search space over both factors and models, while using structured diagnostics to guide search toward more promising directions under a limited budget.

## 3 Method

### 3.1 Problem Formulation

**Configuration space.** We study how an LLM-based ML research agent can discover high-quality forecasting configurations in a nonstationary time-series setting. The agent operates over a configuration space $\mathcal{C} = \mathcal{F} \times \mathcal{G}$, where each configuration $C = (\Phi, g_\theta)$ pairs a feature-construction

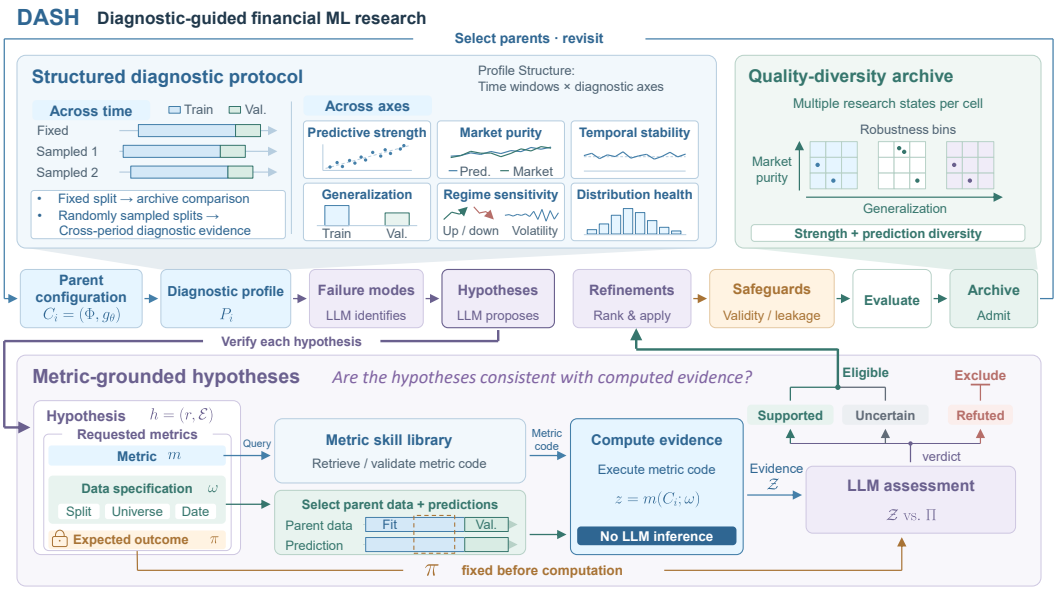


Figure 1: **Overview of DASH**. Structured evaluation produces a diagnostic profile for a selected parent configuration. The agent identifies failure modes and assesses explanatory hypotheses against code-computed evidence and pre-specified expectations. Eligible hypotheses guide refinement, and evaluated configurations enter the archive subject to admission criteria for subsequent selection.

component $\Phi$ with a model and training specification $g_\theta$, producing predictions $\hat{y}_{t,i} = g_\theta(\Phi(X_{t,i}))$. Given a panel dataset $\mathcal{D} = \{(X_{t,i}, y_{t,i})\}_{t=1,i=1}^{T,N}$ of historical market features and return targets, the agent trains each proposed configuration on a training period and evaluates it on a validation period. Under a prescribed search budget, the goal is to discover configurations that generalize to a strictly held-out forward test period. This period remains unseen during search and configuration selection. We consider two search settings. In *joint factor–model search*, both the factor construction $\Phi$ and the model specification $g_\theta$ may be refined. In *factor-only search*, the model specification is fixed and the search is restricted to $\mathcal{F} \times \{g_0\}$, corresponding to the factor-mining setting in which the agent searches only over predictive factors.

**Research state.** We associate each evaluated configuration with a structured research state $s_i = (C_i, P_i, \mathcal{R}_i)$, where $P_i$ is its diagnostic profile and $\mathcal{R}_i$ contains the linked diagnostic records—observations, hypotheses, evidence assessments, and intervention plans—associated with that configuration. Search proceeds by expanding these states. Diagnostics from the current state guide the generation of successor states through targeted interventions on its configuration.

**Overview.** DASH iteratively selects parent states from the archive, diagnoses their configurations, and generates evidence-grounded refinements (Figure 1). The resulting configurations are evaluated for archive admission, and retained states remain available for subsequent expansion.

### 3.2 Diagnostic-Guided Search

DASH uses structured diagnostic evidence as the interface between evaluation and search. The evaluation protocol constructs a behavioral profile for each configuration, which provides the basis for failure-mode identification and subsequent refinement.

#### 3.2.1 Diagnostic Evaluation Protocol

An aggregate performance score indicates how well a configuration performs, but provides limited information about where it fails or what should be changed. DASH therefore evaluates configurations along multiple diagnostic axes and across multiple periods. The diagnostic axes characterize different aspects of configuration behavior, while hybrid temporal evaluation helps distinguish recurring weaknesses from period-specific patterns. The resulting profile provides the LLM with structured evidence for subsequent hypothesis generation and targeted interventions.

**Hybrid temporal evaluation.** We combine shared reference splits $\mathcal{V}_{\text{ref}}$ with randomly sampled diagnostic splits $\mathcal{V}_{\text{diag},i}$. Each split pairs a historical training window with a subsequent validation

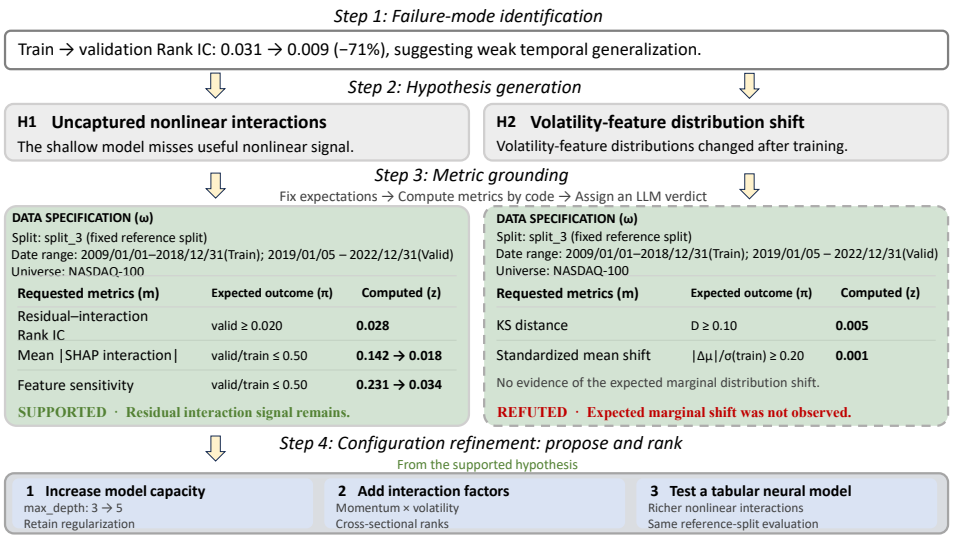


Figure 2: **Case study of diagnostic-guided refinement.** A training–validation Rank IC gap motivates two hypotheses: uncaptured nonlinear interactions and volatility-feature distribution shift. The LLM compares code-computed metrics with pre-specified expectations, supporting the first hypothesis based on residual–interaction Rank IC and refuting the second because the expected marginal shift is not observed. The supported hypothesis guides proposals to increase model capacity, add interaction factors, or test a tabular neural model.

window within the search data. Reference splits provide a common basis for candidate comparison, while diagnostic splits probe alternative temporal windows to distinguish persistent weaknesses from period-specific behavior. The resulting profile is

$$P_i = \text{Evaluate}\big(C_i; \mathcal{V}_{\text{ref}} \cup \mathcal{V}_{\text{diag},i}\big). \tag{1}$$

**Multi-axis diagnostic profile.** The profile characterizes predictive strength, market purity, temporal stability, generalization, regime sensitivity, and prediction-distribution health. The LLM receives a compact view of these diagnostics that preserves split-specific evidence and cross-period summaries, providing actionable feedback beyond a single scalar score. The full profile structure, metric definitions, and temporal split construction are detailed in Appendix C.

### 3.2.2 Diagnostic-Guided Refinement

Rather than generating modifications at random or guiding in-context optimization solely with a scalar reward, DASH adopts diagnostic-guided search: the current state's diagnostics guide the next refinement. Starting from a parent configuration, DASH identifies failure modes, generates explanatory hypotheses, and assesses them against computed evidence before proposing refinements.

**Failure-mode identification.** The agent inspects $P_i$ and identifies failure modes—patterns of underperformance or instability supported by specific metrics. Each failure mode specifies the affected diagnostic axis, temporal or regime scope, severity, and supporting evidence.

**Hypothesis generation and metric-grounding.** For each identified failure mode, the agent proposes distinct hypotheses linking the observed behavior to configuration choices. A hypothesis $h = (r, \mathcal{E})$ pairs an explanation $r$ with metric requests $\mathcal{E} = \{e_j\}_{j=1}^{M}$. Each request $e_j = (m_j, \omega_j, \pi_j)$ specifies a diagnostic metric $m_j$, a data specification $\omega_j$ on which it should be computed, and expected outcomes $\pi_j$ that would support or refute the hypothesis. These expectations form $\Pi = \{\pi_j\}_{j=1}^{M}$ and are fixed before requested computations are executed.

An external skill library computes $z_j = m_j(C_i; \omega_j)$ on the stored predictions and associated search-period data. It reuses an existing metric implementation or validates a newly generated one when necessary (Appendix D). Once the implementation is available, numerical execution requires no LLM inference: the evidence $\mathcal{Z} = \{z_j\}_{j=1}^{M}$ is computed directly from data by code. The LLM compares the computed evidence $\mathcal{Z}$ with the corresponding expectations $\Pi$ to assign a verdict

$$v(h|\mathcal{Z}) \in \{\text{SUPPORTED}, \text{REFUTED}, \text{UNCERTAIN}\}. \tag{2}$$

Refuted hypotheses are excluded, while supported and uncertain hypotheses remain eligible for refinement. Expectations are fixed before the corresponding measurements are computed, and the LLM assesses their consistency with returned evidence. This constrains the rationale for an edit using measured behavior of the current parent. Figure 2 illustrates this mechanism through a case study.

**Configuration refinement.** Eligible hypotheses guide refinement plans specifying changes in configurations. Plans are ranked by expected improvement, resistance to noise fitting, and implementation feasibility, and up to $k$ top-ranked plans are implemented. Each selected refinement $a_{ij}$ produces $C_j = \mathrm{Apply}(C_i, a_{ij})$, which undergoes validity and leakage checks and is evaluated to obtain $P_j$. Successful evaluation yields a successor state $s_j = (C_j, P_j, \mathcal{R}_j)$, where $\mathcal{R}_j$ extends the parent's research trace with the diagnostic rationale and applied changes.

### 3.3 QUALITY-DIVERSITY ARCHIVE

DASH maintains an archive of high-quality and diverse configurations. Each entry stores the research state $s_i = (C_i, P_i, \mathcal{R}_i)$ formed after evaluating configuration $C_i$ on search data, preserving its diagnostic profile and research trace for subsequent refinement.

**Archive structure and search.** The archive $\mathcal{A}$ groups configurations into cells using discretized generalization, market purity, and robustness scores from the shared reference splits $\mathcal{V}_{\mathrm{ref}}$. At each iteration, DASH selects parent configurations across occupied cells using a fitness score that combines predictive strength with prediction diversity. Selected configurations undergo diagnosis-guided refinement, and successfully evaluated candidates are added if they satisfy the archive admission criteria. Appendix E specifies the complete rules.

**Balancing exploitation and exploration.** Fitness-based ranking favors configurations with strong predictive performance and low prediction redundancy, concentrating search on promising configurations under a limited budget. Selection across diagnostic cells provides access to configurations with different diagnostic profiles. These profiles expose different failure modes to the LLM, supporting diverse hypotheses and refinement directions in subsequent iterations.

## 4 EXPERIMENTS

### 4.1 EXPERIMENTAL SETUP

**Datasets.** We evaluate on two daily-frequency equity universes: NASDAQ-100 and S&P 500 constituents. Each dataset spans January 2005 to October 2025, providing approximately two decades of OHLCV data across varied market regimes.

**Evaluation.** The timeline is split into two disjoint periods:

- During the train–search period (2005–2022), all methods use search-period data to develop and select candidates. In joint search, the baselines use mean validation IC over three fixed walk-forward splits. DASH uses shared reference splits for archive scoring and additional sampled splits for diagnostic refinement, with archive selection defined in Appendix E.
- The *final OOD evaluation* period (2023–2025) is never used during search. For OOD evaluation, we simulate chronological deployment: after each calendar year completes, it may be incorporated into expanding training history for the next year's forecast, as detailed in Table 10.

Joint-search results evaluate selected factor–model configurations, whereas factor-only results evaluate validation-selected factors and their prescribed ensembles.

**Metrics.** We report two groups: (1) predictive power: IC, ICIR, Rank IC, Rank ICIR; and (2) portfolio performance: annualized return (ARR), information ratio (IR), and maximum drawdown (MDD) from a daily-rebalanced TopK-Dropout backtest. Definitions and backtest parameters are in Appendix F.4.

**Baselines.** We consider two groups of baselines corresponding to the two search settings. For *factor-only search*, we compare against the following agentic factor-mining methods: AlphaAgent (Tang et al., 2025), FactorMiner (Wang et al., 2026), and R&D-Agent (Factor) (Li et al., 2025). For the broader *joint factor–model setting*, we include two types of baselines: standard predictive ML models, including XGBoost (Chen & Guestrin, 2016), MLP, and GRU (Cho et al., 2014); and agentic baselines, including AutoResearch (Karpathy, 2026), ML-Master 2.0 (Zhu et al., 2026), R&D-Agent (Li et al., 2025), and AIRA (Toledo et al., 2025).

### 4.2 MAIN RESULTS

**Factor–Model Search.** Tables 1 and 2 compare agentic methods under the eight-iteration budget across three repeated runs. On **NASDAQ-100**, our GPT-4.1 variant achieves the highest Rank IC, ICIR, and IR, and improves IC by 8.7% over the strongest agentic baseline. Our two variants rank first and second in ICIR and IR, demonstrating gains with both LLM backbones. On **S&P 500**, our GPT-5-mini variant leads all four metrics, with IC and ICIR exceeding the strongest baseline by 38.2% and 40.2%, respectively. These results show gains in predictive quality and portfolio performance. Additional metrics are reported in Appendix G.2.1. Under matched token budgets, our method also achieves the highest IR and ARR on both universes (Appendix G.2.2, Table 21).

Table 1: Out-of-distribution results on NASDAQ-100, reported as mean ± standard deviation across multiple runs. **Bold**: best; underline: second best. Higher is better for all metrics.

| | Models | Predictive Power | | | Portfolio Performance |
|---|---|---|---|---|---|
| | | **IC** | **Rank IC** | **ICIR** | **IR** |
| ML Baselines | $\text{MLP}_{\text{OHLCV}}$ | $0.0122 \pm 0.0059$ | $0.0105 \pm 0.0079$ | $0.0907 \pm 0.0456$ | $-0.1455 \pm 0.1391$ |
| | $\text{MLP}_{\text{Alpha158}}$ | $0.0109 \pm 0.0021$ | $0.0040 \pm 0.0047$ | $0.0755 \pm 0.0122$ | $0.1768 \pm 0.1732$ |
| | $\text{XGBoost}_{\text{OHLCV}}$ | $0.0062 \pm 0.0026$ | $0.0011 \pm 0.0028$ | $0.0448 \pm 0.0188$ | $0.2217 \pm 0.0434$ |
| | $\text{XGBoost}_{\text{Alpha158}}$ | $\mathbf{0.0323 \pm 0.0042}$ | $\underline{0.0230 \pm 0.0056}$ | $0.2071 \pm 0.0235$ | $0.8299 \pm 0.6723$ |
| | $\text{GRU}_{\text{OHLCV}}$ | $0.0143 \pm 0.0120$ | $0.0113 \pm 0.0024$ | $0.1141 \pm 0.0946$ | $0.0040 \pm 0.8751$ |
| | $\text{GRU}_{\text{Alpha158}}$ | $0.0044 \pm 0.0189$ | $0.0066 \pm 0.0167$ | $0.0375 \pm 0.1567$ | $-0.5428 \pm 0.7631$ |
| Agentic Frameworks | $\text{AutoResearch}_{\text{GPT-4.1}}$ | $0.0234 \pm 0.0007$ | $0.0181 \pm 0.0009$ | $0.1450 \pm 0.0055$ | $-0.6417 \pm 0.7035$ |
| | $\text{AutoResearch}_{\text{GPT-5-mini}}$ | $0.0176 \pm 0.0042$ | $0.0148 \pm 0.0014$ | $0.1254 \pm 0.0415$ | $-0.2929 \pm 0.1464$ |
| | $\text{R\&D-Agent}_{\text{GPT-4.1}}$ | $0.0287 \pm 0.0006$ | $0.0186 \pm 0.0023$ | $0.1833 \pm 0.0095$ | $0.8325 \pm 0.3454$ |
| | $\text{R\&D-Agent}_{\text{GPT-5-mini}}$ | $0.0251 \pm 0.0027$ | $0.0217 \pm 0.0010$ | $0.1691 \pm 0.0204$ | $0.5903 \pm 0.0390$ |
| | $\text{AIRA}_{\text{GPT-4.1}}$ | $0.0169 \pm 0.0031$ | $0.0115 \pm 0.0072$ | $0.1171 \pm 0.0190$ | $-0.0265 \pm 0.6028$ |
| | $\text{AIRA}_{\text{GPT-5-mini}}$ | $0.0165 \pm 0.0012$ | $0.0097 \pm 0.0047$ | $0.1209 \pm 0.0131$ | $0.1392 \pm 0.2452$ |
| | $\text{ML-Master2.0}_{\text{GPT-4.1}}$ | $0.0272 \pm 0.0014$ | $0.0179 \pm 0.0008$ | $0.1634 \pm 0.0042$ | $0.5287 \pm 0.2585$ |
| | $\text{ML-Master2.0}_{\text{GPT-5-mini}}$ | $0.0234 \pm 0.0039$ | $0.0113 \pm 0.0086$ | $0.1425 \pm 0.0164$ | $0.2266 \pm 0.8699$ |
| | **Ours**$_{\text{GPT-4.1}}$ | $\underline{0.0312 \pm 0.0013}$ | $\mathbf{0.0275 \pm 0.0028}$ | $\mathbf{0.2243 \pm 0.0088}$ | $\mathbf{1.0428 \pm 0.0417}$ |
| | **Ours**$_{\text{GPT-5-mini}}$ | $0.0301 \pm 0.0051$ | $0.0155 \pm 0.0082$ | $\underline{0.2116 \pm 0.0236}$ | $\underline{0.9801 \pm 0.5754}$ |

Table 2: Out-of-distribution results on S&P 500, reported as mean ± standard deviation across multiple runs. **Bold**: best; underline: second best. Higher is better for all metrics.

| | Models | Predictive Power | | | Portfolio Performance |
|---|---|---|---|---|---|
| | | **IC** | **Rank IC** | **ICIR** | **IR** |
| ML Baselines | $\text{MLP}_{\text{OHLCV}}$ | $0.0118 \pm 0.0004$ | $\underline{0.0120 \pm 0.0002}$ | $0.1301 \pm 0.0097$ | $-0.1043 \pm 0.2461$ |
| | $\text{MLP}_{\text{Alpha158}}$ | $0.0066 \pm 0.0011$ | $0.0068 \pm 0.0043$ | $0.0613 \pm 0.0120$ | $-0.6105 \pm 0.1364$ |
| | $\text{XGBoost}_{\text{OHLCV}}$ | $0.0036 \pm 0.0014$ | $-0.0021 \pm 0.0014$ | $0.0379 \pm 0.0150$ | $-0.2270 \pm 0.1683$ |
| | $\text{XGBoost}_{\text{Alpha158}}$ | $0.0108 \pm 0.0011$ | $0.0047 \pm 0.0030$ | $0.0943 \pm 0.0078$ | $0.2494 \pm 0.1475$ |
| | $\text{GRU}_{\text{OHLCV}}$ | $0.0101 \pm 0.0030$ | $0.0099 \pm 0.0035$ | $0.1128 \pm 0.0364$ | $-0.1927 \pm 0.2446$ |
| | $\text{GRU}_{\text{Alpha158}}$ | $0.0111 \pm 0.0055$ | $0.0055 \pm 0.0028$ | $0.1383 \pm 0.0744$ | $-0.0023 \pm 0.2545$ |
| Agentic Frameworks | $\text{AutoResearch}_{\text{GPT-4.1}}$ | $0.0146 \pm 0.0004$ | $0.0064 \pm 0.0004$ | $0.1162 \pm 0.0019$ | $\underline{0.5180 \pm 0.0624}$ |
| | $\text{AutoResearch}_{\text{GPT-5-mini}}$ | $0.0123 \pm 0.0054$ | $0.0016 \pm 0.0095$ | $0.0933 \pm 0.0411$ | $0.3035 \pm 0.1424$ |
| | $\text{R\&D-Agent}_{\text{GPT-4.1}}$ | $\underline{0.0178 \pm 0.0016}$ | $0.0105 \pm 0.0002$ | $\underline{0.1424 \pm 0.0090}$ | $0.1954 \pm 0.2714$ |
| | $\text{R\&D-Agent}_{\text{GPT-5-mini}}$ | $0.0163 \pm 0.0015$ | $0.0065 \pm 0.0008$ | $0.1254 \pm 0.0160$ | $0.2004 \pm 0.1692$ |
| | $\text{AIRA}_{\text{GPT-4.1}}$ | $0.0119 \pm 0.0034$ | $0.0039 \pm 0.0039$ | $0.1081 \pm 0.0256$ | $-0.3977 \pm 0.0833$ |
| | $\text{AIRA}_{\text{GPT-5-mini}}$ | $0.0118 \pm 0.0026$ | $0.0041 \pm 0.0002$ | $0.1038 \pm 0.0206$ | $-0.3175 \pm 0.3672$ |
| | $\text{ML-Master2.0}_{\text{GPT-4.1}}$ | $0.0172 \pm 0.0031$ | $0.0098 \pm 0.0021$ | $0.1401 \pm 0.0278$ | $0.0399 \pm 0.3596$ |
| | $\text{ML-Master2.0}_{\text{GPT-5-mini}}$ | $0.0139 \pm 0.0052$ | $0.0060 \pm 0.0029$ | $0.1009 \pm 0.0390$ | $-0.0801 \pm 0.5159$ |
| | **Ours**$_{\text{GPT-4.1}}$ | $0.0175 \pm 0.0005$ | $0.0101 \pm 0.0008$ | $0.1377 \pm 0.0010$ | $0.3150 \pm 0.2342$ |
| | **Ours**$_{\text{GPT-5-mini}}$ | $\mathbf{0.0246 \pm 0.0057}$ | $\mathbf{0.0185 \pm 0.0027}$ | $\mathbf{0.1996 \pm 0.0381}$ | $\mathbf{1.1214 \pm 0.1819}$ |

**Factor-Only Search**. Table 3 reports the best factor selected by absolute IC on the common validation split at each token budget. Our method leads all four metrics on both markets at 200K tokens and on S&P 500 at 400K, where it improves further. On NASDAQ-100, the same factor is selected at both budgets, achieving IC 0.0307—over twice that of the strongest 200K baseline (AlphaAgent, 0.0144). At 400K, it leads all metrics except ICIR, where R&D-Agent scores higher. Full factor and ensemble results are in Appendix G.1.1.

**Additional Evaluations.** We further assess DASH on cryptocurrency data and point-in-time (PIT) equity universes. On Crypto, its validation-selected factors lead all four predictive metrics at 200K and 400K tokens. In joint factor–model search, its 200K-token results exceed evaluated 400K-token

Table 3: Factor-only OOD performance of the best factor selected by absolute IC on the common validation split at 200K and 400K tokens. Our method selects the same factor at both budgets on NASDAQ-100.

| | | NASDAQ-100 | | | | S&P 500 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Method** | **IC** | **Rank IC** | **ICIR** | **Rank ICIR** | **IC** | **Rank IC** | **ICIR** | **Rank ICIR** |
| 200K tokens | FactorMiner | 0.0064 | 0.0091 | 0.0432 | 0.0570 | 0.0096 | 0.0081 | 0.0818 | 0.0636 |
| | AlphaAgent | 0.0144 | 0.0231 | 0.1156 | 0.1260 | −0.0063 | −0.0070 | −0.0465 | −0.0466 |
| | R&D-Agent | 0.0131 | 0.0141 | 0.0954 | 0.0911 | −0.0001 | −0.0011 | −0.0013 | −0.0130 |
| | **Ours** | **0.0307** | **0.0311** | **0.1650** | **0.1675** | **0.0159** | **0.0141** | **0.1262** | **0.1016** |
| 400K tokens | FactorMiner | 0.0064 | 0.0091 | 0.0432 | 0.0570 | 0.0096 | 0.0081 | 0.0818 | 0.0636 |
| | AlphaAgent | 0.0144 | 0.0231 | 0.1156 | 0.1260 | −0.0063 | −0.0070 | −0.0465 | −0.0466 |
| | R&D-Agent | 0.0263 | 0.0240 | **0.1853** | 0.1613 | −0.0001 | −0.0011 | −0.0013 | −0.0130 |
| | **Ours** | **0.0307** | **0.0311** | 0.1650 | **0.1675** | **0.0170** | **0.0156** | **0.1297** | **0.1110** |

baselines in mean IC and portfolio IR (Appendix G.3). On the PIT versions of NASDAQ-100 and S&P 500, DASH achieves the highest mean IC, ICIR, and NAV at both 100K and 400K tokens (Appendix G.4). These results extend its predictive and portfolio advantages to cryptocurrency forecasting and show that the advantages persist under point-in-time equity universe construction.

## 4.3 Search Behavior Analysis

### 4.3.1 Refinement Effectiveness and Search Trajectories

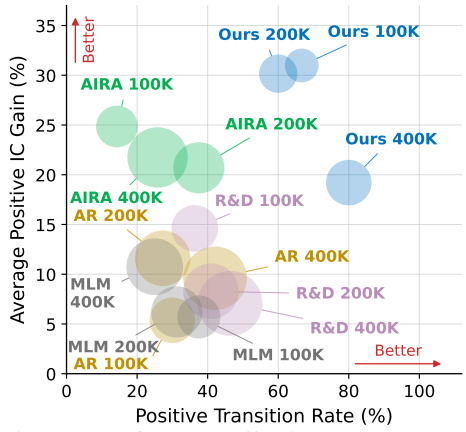


Figure 3: Refinement effectiveness across token budgets. Bubble size indicates the number of transitions.

**Refinement effectiveness.** We assess refinement effectiveness along two dimensions: how often a child configuration improves its parent and how large the improvement is when it succeeds. Figure 3 compares these quantities on NASDAQ-100 under matched token budgets. Each evaluated parent–child pair corresponds to one state transition. The x-axis reports the fraction of pairs in which the child achieves a higher OOD IC than its parent, while the y-axis reports the mean relative IC gain among improving pairs. At 100K and 200K tokens, DASH achieves both the highest improvement rate and the largest mean positive IC gain. At 400K, it retains the highest improvement rate while maintaining a positive gain comparable to the strongest baseline. Overall, DASH achieves a high refinement success rate while maintaining competitive gains among successful refinements.

**Search Trajectory**. Over the first three iterations of the 8-iteration search, DASH achieves mean parent-to-child IC gains of 28.64% on NASDAQ-100 and 37.60% on S&P 500, versus 18.50% and 22.61% for the strongest baselines, respectively, and the largest mean ICIR gains on both markets (Table 4). On NASDAQ-100, starting from the same initial configuration, DASH shows an overall upward trend with relatively small fluctuations, achieving the highest IC and ICIR at iteration 8 (Figure 4). R&D-Agent and ML-Master 2.0 plateau after early gains, while AutoResearch and AIRA show limited or declining progress.

### 4.3.2 High-Quality Factor Discovery

We next examine the quality distribution of factors discovered under matched token budgets. Figure 5 reports the fraction of evaluable valid factors whose OOD IC exceeds several thresholds. Our method achieves the highest proportion across all budgets and thresholds. At 400K tokens, 43.1% of its factors exceed an OOD IC of 0.01, compared with 16.3% for FactorMiner, and 29.3% exceed 0.015. This indicates more efficient discovery of high-quality factors under the same token budget. The NASDAQ-100 results reported in Appendix G.1.2. show a similar advantage at higher IC thresholds, with DASH leading at OOD IC above 0.02 across all evaluated budgets. These results indicate that DASH discovers a higher concentration of strongly predictive factors under matched token budgets.

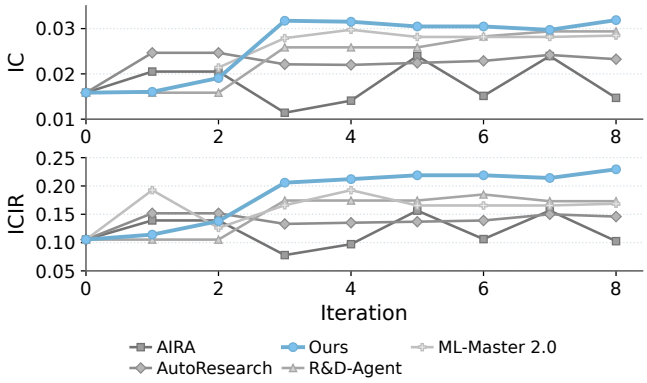


Figure 4: OOD performance by iteration on NASDAQ-100.

Table 4: Average of child-to-parent relative change over iterations 1–3.

| | Method | IC | ICIR |
|---|---|---|---|
| NASDAQ-100 | AutoResearch | 11.17% | 8.34% |
| | R&D-Agent | 15.80% | 16.40% |
| | AIRA | 2.12% | 3.23% |
| | ML-Master 2.0 | 18.50% | 24.06% |
| | **Ours** | **28.64%** | **26.18%** |
| S&P 500 | AutoResearch | 22.61% | 16.31% |
| | R&D-Agent | 20.68% | 12.51% |
| | AIRA | -1.09% | -0.98% |
| | ML-Master 2.0 | 3.38% | 7.47% |
| | **Ours** | **37.60%** | **29.70%** |

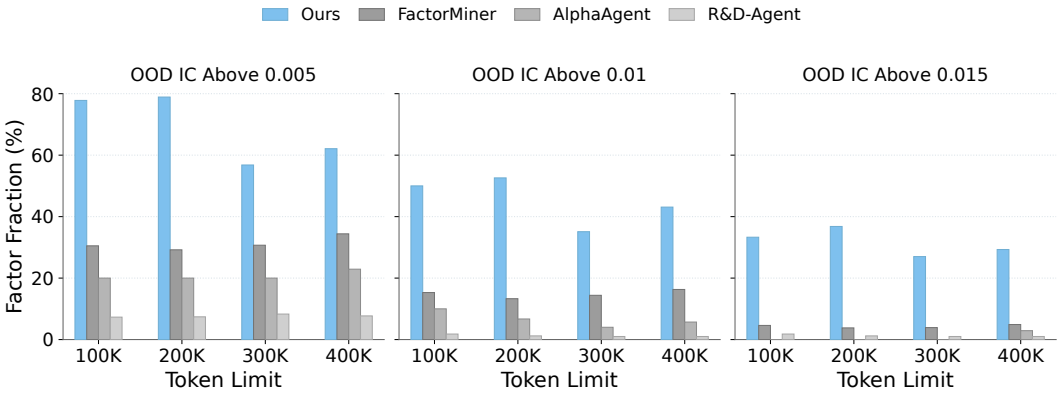


Figure 5: Percentage of OOD-evaluable valid factors with OOD IC above 0.005, 0.01, and 0.015 under different token budgets for S&P 500.

#### 4.3.3 Diversity and Redundancy

We examine diversity at two levels: behavioral coverage of the configuration archive and predictive redundancy within the discovered factor pools. Figure 6 shows that the archive expands from an initially narrow region to cover a broader range of generalization, robustness, and market-purity profiles over successive iterations. The retained configurations provide alternative starting points with different diagnostic behaviors for subsequent refinement. At the factor level, Appendix G.1.3 quantifies pairwise factor correlations and overlap with the Alpha158 library. At 400K tokens, only 2.5% and 2.4% of DASH's factor pairs have absolute correlation above 0.8 on NASDAQ-100 and S&P 500, respectively, indicating limited duplication through highly correlated signals. Combined with the preceding quality analysis, these results show that the discovered factor pools contain a high proportion of strong OOD factors while maintaining low overall predictive redundancy.

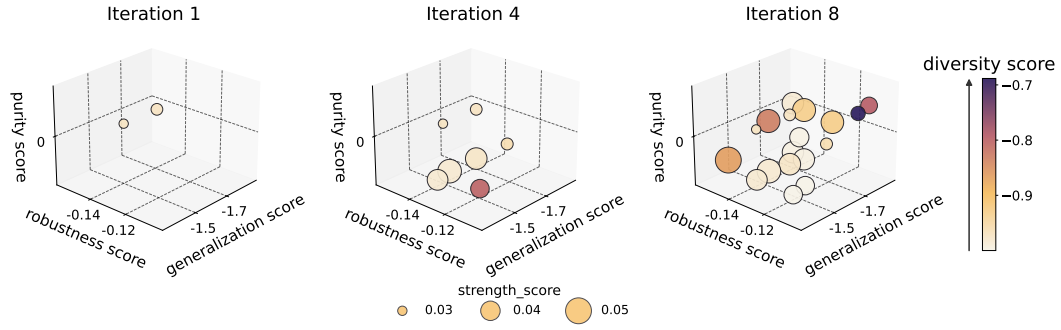


Figure 6: Archive evolution on NASDAQ-100 at iterations 1, 4, and 8. Axes: behavioral scores (generalization, robustness, purity). Marker size: strength score; color: diversity relative to existing elites (darker = higher).

### 4.4 Ablation Studies

**Component Ablation.** Table 5 examines the contribution of individual components. On NASDAQ-100, removing the diagnostic loop causes the largest drops across all four predictive metrics, including decreases of 0.0159 in IC and 0.1122 in ICIR. Removing the quality-diversity archive produces the largest reductions in ARR and IR, with IR decreasing by 1.8188. On S&P 500, the archive ablation yields the largest drops in IC, ICIR, ARR, and IR. All component ablations weaken predictive performance on both markets, while their effects on portfolio performance vary. In particular, removing the hybrid split protocol improves ARR and IR on S&P 500 despite reducing predictive metrics. Together, these results support the complementary contributions of diagnostic-guided refinement and the quality-diversity archive, while highlighting that their effects depend on the market and evaluation metric.

Table 5: Component ablation. The first row per universe reports absolute values for our full method; subsequent rows show signed differences (ablation − Ours).

| Variant | | Predictive Power | | | | Portfolio Performance | | |
|---|---|---|---|---|---|---|---|---|
| | | **IC** | **ICIR** | **Rank IC** | **Rank ICIR** | **ARR** | **IR** | **MDD** |
| **NASDAQ-100** | **Ours**$_{\text{GPT-4.1}}$ | 0.0319 | 0.2294 | 0.0291 | 0.2207 | 0.1046 | 1.5581 | -0.0474 |
| | w/o LLM Rank | −0.0055 | −0.0547 | −0.0128 | −0.1140 | −0.0200 | −0.7893 | −0.0936 |
| | w/o hybrid splits | −0.0092 | −0.0845 | −0.0124 | −0.1117 | −0.0357 | −0.9254 | −0.0981 |
| | w/o diagnostics | −0.0159 | −0.1122 | −0.0230 | −0.1778 | −0.0552 | −1.1363 | −0.1523 |
| | w/o archive | −0.0049 | −0.0370 | −0.0042 | −0.0572 | −0.1333 | −1.8188 | −0.1098 |
| **S&P 500** | **Ours**$_{\text{GPT-4.1}}$ | 0.0180 | 0.1373 | 0.0109 | 0.0767 | 0.0520 | 0.4505 | -0.0982 |
| | w/o LLM Rank | −0.0019 | −0.0088 | −0.0024 | −0.0150 | −0.0406 | −0.3424 | −0.0194 |
| | w/o hybrid splits | −0.0004 | −0.0083 | −0.0024 | −0.0129 | +0.0199 | +0.2056 | −0.0122 |
| | w/o diagnostics | −0.0013 | −0.0021 | −0.0017 | −0.0076 | −0.0284 | −0.2370 | −0.0170 |
| | w/o archive | −0.0029 | −0.0227 | −0.0012 | −0.0080 | −0.0566 | −0.4950 | −0.0403 |

**Metric-Grounding Ablation.** Table 6 isolates the effect of externally computed metric evidence. We compare against *LLM Verdict*, where the LLM proposes and assesses hypotheses without external measurement, and *No Verdict*, where hypothesis verification is removed. Replacing metric-grounded assessment with LLM-only judgment reduces IC from 0.0319 to 0.0189 (41%) for GPT-4.1 and from 0.0244 to 0.0193 (21%) for GPT-5-mini. These results support the value of grounding refinement decisions in externally computed diagnostic evidence.

Table 6: Metric-grounding ablation on NASDAQ-100. **Bold**: best; underline: second best.

| Variant | | Predictive Power | | | | Portfolio Performance | | |
|---|---|---|---|---|---|---|---|---|
| | | **IC** | **ICIR** | **Rank IC** | **Rank ICIR** | **ARR** | **IR** | **MDD** |
| **GPT-4.1** | **Ours** | **0.0319** | **0.2294** | **0.0291** | **0.2207** | **0.1046** | **1.5581** | **-0.0474** |
| | LLM Verdict | 0.0189 | 0.1272 | 0.0018 | 0.0130 | 0.0258 | 0.2439 | -0.1533 |
| | No Verdict | 0.0169 | 0.1198 | 0.0046 | 0.0314 | 0.0158 | 0.1478 | -0.1446 |
| **GPT-5-mini** | **Ours** | **0.0244** | **0.1849** | **0.0109** | **0.0782** | **0.1685** | **1.0252** | **-0.1244** |
| | LLM Verdict | 0.0193 | 0.1273 | 0.0084 | 0.0574 | 0.0424 | 0.3866 | -0.1486 |
| | No Verdict | 0.0223 | 0.1557 | 0.0078 | 0.0540 | 0.0188 | 0.2082 | -0.1785 |

## 5 Conclusion

We presented DASH, a framework that combines structured diagnostic feedback with quality-diversity search for financial ML research agents. Diagnostic evidence serves two roles: grounding hypothesis-driven refinement and directing exploration across behaviorally distinct configurations. Experiments show strong forward-test predictive and portfolio performance under matched budgets. Factor-level analyses reveal high-quality discoveries and low predictive redundancy in the overall factor pools, while additional market and universe evaluations extend the empirical findings. Ablations across two LLM backbones show weaker predictive performance without externally computed evidence, supporting the value of metric-grounded refinement.

**Limitations.** Our evaluation is limited to daily-frequency financial forecasting. Applying DASH to other domains requires task-appropriate diagnostics, behavioral descriptors, and refinement specifications; this broader applicability remains to be evaluated.

## AI USE STATEMENT

**AI as part of the research method.** Large language models are an integral component of the proposed agentic framework. Within DASH, LLMs inspect structured diagnostic feedback to identify failure modes, formulate explanatory hypotheses, and specify supplemental metric requests, including the data scope and expected outcomes used to assess each hypothesis. After the requested evidence is computed by code, the LLM compares the results with the pre-specified expectations and assigns each hypothesis a supported, refuted, or uncertain assessment. Eligible hypotheses are then used to generate and rank candidate refinement plans. The LLM also generates or modifies factor and model implementations within the permitted search space and, when a requested diagnostic metric is not already available, may generate a candidate metric implementation that is validated before use. Numerical metric computation and evaluation are executed programmatically rather than inferred by the LLM. These uses constitute part of the method studied and evaluated in this work and are described in the corresponding methodological and experimental sections.

**AI assistance in manuscript preparation.** Under human supervision, generative AI tools were used to assist with manuscript preparation, including improving clarity and readability, formatting, and refining the presentation of figures. All AI-assisted edits were reviewed and approved by the authors. The authors take full responsibility for the final content of this work.

## ETHICS STATEMENT

This study uses historical market data for the NASDAQ-100, S&P 500, and cryptocurrency assets obtained from third-party data providers. The data are used solely for research purposes and in accordance with the applicable provider terms and licensing restrictions. Because we do not hold redistribution rights for the underlying datasets, we will not publicly release or redistribute the raw market data used in our experiments. The study does not involve human subjects or personally identifiable information.

## REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide an anonymized code repository at `https://anonymous.4open.science/r/DASH_ICLR-8D93`, containing the implementation of DASH for reproducing the main experiments. Appendix A provides the complete search algorithm; Appendices C–E specify the diagnostic profiles, metric-grounding procedure, and archive mechanisms; and Appendix F documents the datasets, baselines, search and evaluation protocols, and evaluation metrics. Appendix H further describes the safeguards used to prevent look-ahead leakage and preserve the evaluation protocol. The market data used in our experiments are obtained from third-party providers and cannot be redistributed under their licensing terms. We therefore document the data scope, time periods, universe construction, target definition, and preprocessing procedures required to reconstruct the experimental setup with appropriately licensed data.

## REFERENCES

David H. Bailey, Jonathan M. Borwein, Marcos López de Prado, and Qiji Jim Zhu. The probability of backtest overfitting. *Journal of Computational Finance*, 20(4):39–69, 2017.

Gavin C. Cawley and Nicola L. C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(70):2079–2107, 2010. URL https://www.jmlr.org/papers/v11/cawley10a.html.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

Yizhou Chi, Yizhang Lin, Sirui Hong, Duyi Pan, Yaying Fei, Guanghao Mei, Bangbang Liu, Tianqi Pang, Jacky Kwok, Ceyao Zhang, et al. Sela: Tree-search enhanced llm agents for automated machine learning. *arXiv preprint arXiv:2410.17238*, 2024.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014. URL https://arxiv.org/abs/1406.1078.

Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

Yuanqi Du, Botao Yu, Tianyu Liu, Tony Shen, Junwu Chen, Jan G Rittig, Kunyang Sun, Yikun Zhang, Zhangde Song, Bo Zhou, et al. Accelerating scientific discovery with autonomous goal-evolving agents. *arXiv preprint arXiv:2512.21782*, 2025.

Haoyang Fang, Boran Han, Nick Erickson, Xiyuan Zhang, Su Zhou, Anirudh Dagar, Jiani Zhang, Ali Caner Turkmen, Cuixiong Hu, Huzefa Rangwala, et al. Mlzero: A multi-agent system for end-to-end machine learning automation. *arXiv preprint arXiv:2505.13941*, 2025.

Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273, 2020.

Siyuan Guo, Cheng Deng, Ying Wen, Hechang Chen, Yi Chang, and Jun Wang. Ds-agent: Automated data science by empowering large language models with case-based reasoning. *arXiv preprint arXiv:2402.17453*, 2024.

Karen Hambardzumyan, Nicolas Baldwin, Edan Toledo, Rishi Hazra, Michael Kuchnik, Bassel Al Omari, Thomas Simon Foster, Anton Protopopov, Jean-Christophe Gagnon-Audet, Ishita Mediratta, et al. Aira_2: Overcoming bottlenecks in ai research agents. *arXiv preprint arXiv:2603.26499*, 2026.

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

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. AIDE: AI-driven exploration in the space of code. *arXiv preprint arXiv:2502.13138*, 2025.

Andrej Karpathy. autoresearch: Ai agents running research on single-gpu nanochat training automatically. https://github.com/karpathy/autoresearch, 2026. GitHub repository.

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. *arXiv preprint arXiv:2505.15155*, 2025. URL https://arxiv.org/abs/2505.15155.

Zujie Liang, Feng Wei, Wujiang Xu, Yuxi Qian, Lin Chen, and Xinhui Wu. I-mcts: Enhancing agentic automl via introspective monte carlo tree search. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 189–210, 2026.

Fei Liu, Xialiang Tong, Mingxuan Yuan, Xi Lin, Fu Luo, Zhenkun Wang, Zhichao Lu, and Qingfu Zhang. Evolution of heuristics: Towards efficient automatic algorithm design using large language model. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32201–32223. PMLR, 2024.

Zexi Liu, Yuzhu Cai, Xinyu Zhu, Yujie Zheng, Runkun Chen, Ying Wen, Yanfeng Wang, Weinan E, and Siheng Chen. Ml-master: Towards ai-for-ai via integration of exploration and reasoning, 2025. URL https://arxiv.org/abs/2506.16499.

Marcos López de Prado. *Advances in Financial Machine Learning*. Wiley, 2018.

Haochen Luo, Ho Tin Ko, Jiandong Chen, David Sun, Yuan Zhang, and Chen Liu. Alphabench: Benchmarking large language models in formulaic alpha factor mining. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025. URL https://arxiv.org/abs/2506.13131.

Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016.

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.

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. *arXiv preprint arXiv:2502.16789*, 2025. URL https://arxiv.org/abs/2502.16789.

Edan Toledo, Karen Hambardzumyan, Martin Josifoski, Rishi Hazra, Nicolas Baldwin, Alexis Audran-Reiss, Michael Kuchnik, Despoina Magka, Minqi Jiang, Alisia Maria Lupidi, Andrei Lupu, Roberta Raileanu, Kelvin Niu, Tatiana Shavrina, Jean-Christophe Gagnon-Audet, Michael Shvartsman, Shagun Sodhani, Alexander H. Miller, Abhishek Charnalia, Derek Dunfield, Carole-Jean Wu, Pontus Stenetorp, Nicola Cancedda, Jakob Nicolaus Foerster, and Yoram Bachrach. Ai research agents for machine learning: Search, exploration, and generalization in mle-bench, 2025. URL https://arxiv.org/abs/2507.02554.

Patara Trirat, Wonyong Jeong, and Sung Ju Hwang. Automl-agent: A multi-agent llm framework for full-pipeline automl. *arXiv preprint arXiv:2410.02958*, 2024.

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. *arXiv preprint arXiv:2602.14670*, 2026. URL https://arxiv.org/abs/2602.14670.

Longkun Xu, Xiaochun Zhang, Qiantu Tuo, and Rui Li. Sea-ts: Self-evolving agent for autonomous code generation of time series forecasting algorithms. *arXiv preprint arXiv:2603.04873*, 2026.

Xiao Yang, Weiqing Liu, Dong Zhou, Jiang Bian, and Tie-Yan Liu. Qlib: An ai-oriented quantitative investment platform. *arXiv preprint arXiv:2009.11189*, 2020.

Xu Yang, Xiao Yang, Shikai Fang, Yifei Zhang, Jian Wang, Bowen Xian, Qizheng Li, Jingyuan Li, Minrui Xu, Yuante Li, et al. R&d-agent: An llm-agent framework towards autonomous data science. *arXiv preprint arXiv:2505.14738*, 2025.

Xinyu Zhu, Yuzhu Cai, Zexi Liu, Bingyang Zheng, Cheng Wang, Rui Ye, Yuzhi Zhang, Linfeng Zhang, Weinan E, Siheng Chen, and Yanfeng Wang. Toward ultra-long-horizon agentic science: Cognitive accumulation for machine learning engineering, 2026. URL https://arxiv.org/abs/2601.10402.

## A SEARCH ALGORITHM

Algorithm 1 presents the complete DASH search procedure. At each iteration, parent configurations are selected from the archive, evaluated to obtain diagnostic profiles, and refined through hypothesis generation and metric grounding. For each parent, up to $k$ top-ranked refinements are implemented, checked for validity and leakage, and evaluated. Eligible successors enter the archive as research states. Parent selection and archive admission follow Appendix E; each diagnostic cell may retain multiple states.

**Algorithm 1** DASH: Diagnostic-Guided Quality-Diversity Search

**Require:** Configuration space $\mathcal{C}$, seed configurations, train-search data $\mathcal{D}$, temporal protocol, iteration budget $T_{\text{search}}$, offspring budget $k$

Evaluate seed configurations; initialize archive $\mathcal{A}$ with their research states
**for** $t = 1, \ldots, T_{\text{search}}$ **do**
  Select parent states across occupied cells by fitness $f_i(\mathcal{A})$ (Appendix E)
  **for** each selected parent state $s_i = (C_i, P_i, \mathcal{R}_i)$ **do**
    Sample diagnostic splits $\mathcal{V}_{\text{diag},i}$ within $\mathcal{D}$
    $P_i \leftarrow \text{Evaluate}(C_i; \mathcal{V}_{\text{ref}} \cup \mathcal{V}_{\text{diag},i})$
    Derive strength $q_i$, fitness $f_i(\mathcal{A})$
    Identify failure modes from $P_i$
    $\mathcal{H}^+ \leftarrow \emptyset$
    **for** each identified failure mode **do**
      Generate diagnostic hypotheses $h = (r, \mathcal{E})$
      **for** each hypothesis $h$ **do**
        Fix requests $\mathcal{E} = \{(m_j, \omega_j, \pi_j)\}_{j=1}^{M}$ and expectations $\Pi = \{\pi_j\}_{j=1}^{M}$
        Compute $z_j = m_j(C_i; \omega_j)$ via metric skills, without LLM inference
        Collect evidence $\mathcal{Z} = \{z_j\}_{j=1}^{M}$
        The LLM compares $\mathcal{Z}$ with $\Pi$ to assign verdict $v(h \mid \mathcal{Z})$
        **if** $v(h \mid \mathcal{Z}) \in \{\texttt{supported}, \texttt{uncertain}\}$ **then**
          $\mathcal{H}^+ \leftarrow \mathcal{H}^+ \cup \{h\}$
        **end if**
      **end for**
    **end for**
    Generate and rank refinement plans conditioned on $\mathcal{H}^+$
    **for** each of up to $k$ top-ranked refinements $a_{ij}$ **do**
      Implement $C_j \leftarrow \text{Apply}(C_i, a_{ij})$
      Run validity and leakage checks; skip this candidate if they fail
      Evaluate $C_j$ to obtain $P_j$ under the parent protocol; skip if evaluation fails
      Extend $\mathcal{R}_i$ with the diagnostic rationale and $a_{ij}$ to obtain $\mathcal{R}_j$
      Construct successor state $s_j = (C_j, P_j, \mathcal{R}_j)$
      Derive strength $q_j$, fitness $f_j(\mathcal{A})$, and the diagnostic cell using $\mathcal{V}_{\text{ref}}$
      **if** archive scores are valid and admission criteria in Appendix E hold **then**
        Insert $s_j$ into $\mathcal{A}$
      **end if**
    **end for**
  **end for**
**end for**
**return** Final archive $\mathcal{A}$

## B FACTOR BASE MANAGEMENT

DASH maintains a clustered factor base for each state. When an intervention modifies the factor component, the corresponding successor inherits the parent's factor base and updates it with newly accepted factors. The root state starts with an empty factor base.

When an intervention proposes changes to the factor component, the agent generates candidate factor implementations. Each candidate must pass the following checks before being admitted to the successor's factor base.

1. **Safeguard checks.** The factor implementation is checked for execution validity, output consistency, and potential look-ahead leakage, as detailed in Appendix H.

2. **IC screening.** The candidate factor is evaluated against the target label over the search period using mean daily cross-sectional IC. A factor is admitted only if

$$|\,\mathrm{IC}(f)| \geq \tau_{\mathrm{IC}},$$

where $\tau_{\mathrm{IC}} = 0.01$ by default.

3. **Cluster-based diversity check.** For a candidate passing the IC screen, DASH compares the factor with the centroids of existing clusters and identifies the most similar one:

$$c^\star = \arg\max_c |\rho_w(f, \mu_c)|\,, \qquad r^\star = |\rho_w(f, \mu_{c^\star})|\,, \tag{3}$$

where $\mu_c$ denotes the centroid of cluster $c$ and $\rho_w$ is the mean daily weighted cross-sectional Pearson correlation. The clustering threshold is $\tau_{\mathrm{corr}} = 0.7$ by default. If $r^\star > \tau_{\mathrm{corr}}$, the factor is assigned to cluster $c^\star$; otherwise, it initializes a new cluster. Both operations are subject to the configured cluster capacity limits.

Accepted factors are appended to the factor set inherited from the parent state, and the resulting factor base is assigned to the successor state. The parent and sibling states remain unchanged. The successor's active factors are then concatenated with the permitted raw features to form the input used for model training.

# C Diagnostic Profile Details

This appendix details the diagnostic profile $P_i$ introduced in Section 3.2.1: the hybrid temporal evaluation, six behavioral axes, hierarchical profile schema, and axis score computation. It also distinguishes the composite diagnostic score from the reference-split quality score used for archive ranking.

## C.1 Hybrid Temporal Evaluation

**Fixed reference split.** The fixed split $\mathcal{V}_{\mathrm{ref}}$ uses a training period from January 1, 2009 to December 31, 2018 and a validation period from January 5, 2019 to December 31, 2022. This split is shared across all candidates and serves as the common reference for archive ranking.

**Random diagnostic splits.** Each evaluation round samples $K = 2$ random splits in $\mathcal{V}_{\mathrm{diag},i}$. Their validation start dates are drawn from January 1, 2015 to December 31, 2018, with a minimum separation of 183 calendar days. Each training window ends on the day before its validation start and extends backward by ten calendar years, with its start clipped to January 1, 2005 if necessary. Each validation window spans one calendar year. All random diagnostic windows therefore precede the 2023 OOD boundary. Together with the fixed reference split, this yields three splits per candidate.

Random splits aim to evaluate configurations more comprehensively across market conditions, reducing sensitivity to regime shifts and overfitting from repeatedly optimizing a single validation score, while exposing period-specific problems to the LLM to encourage diverse diagnostic reasoning paths; the fixed split provides a common basis for fair comparison and selection across states. Figure 7 illustrates DASH's evaluation protocol across the train–search and final OOD evaluation stages.

## C.2 Profile Axes

Table 7 summarizes the six axes. Diagnostics are computed independently per split, with generalization comparing training and validation performance. Selected statistics are aggregated in cross-split summaries.

## C.3 Profile Schema

For each configuration $C_i$, the evaluator constructs a hierarchical profile $P_i$ organized as follows:

**Hierarchical Organization.** The profile is organized into four levels. *Split-level diagnostics* report metrics for each temporal split, computed separately on the training and validation portions. *Train–validation gap diagnostics* characterize per-split differences through relative gaps ($m_{\mathrm{train}} -$

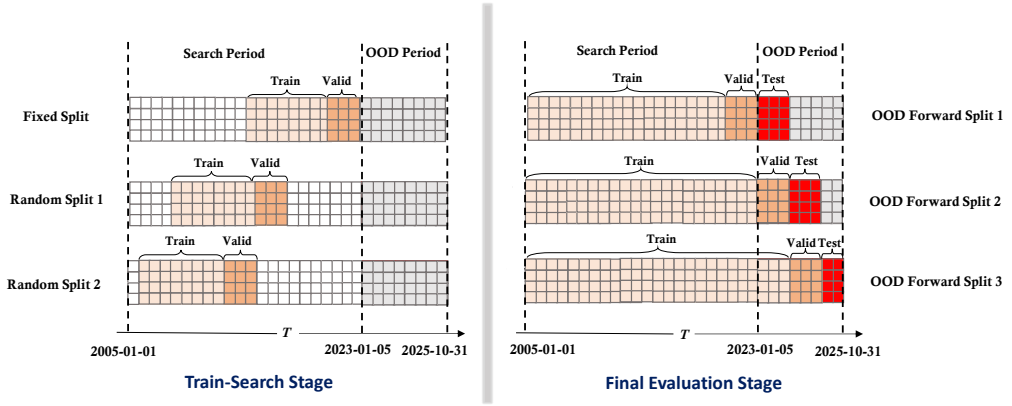


Figure 7: DASH evaluation protocol. Left: hybrid temporal evaluation during the train–search stage, consisting of one fixed reference split and randomly anchored diagnostic splits. Right: the OOD evaluation protocol shared by DASH and the baseline methods, comprising three forward evaluation cycles, each with training, validation, and a subsequent OOD test period.

Table 7: Diagnostic axes of the profile $P_i$. Each axis exposes a distinct dimension of configuration behavior and enables targeted failure diagnosis.

| Axis | What it captures | Representative fields |
|---|---|---|
| Predictive strength | Whether the signal reliably predicts cross-sectional returns | IC, Rank IC, ICIR, top–bottom spread |
| Market purity | Dependence of mean predictions on market direction | Beta exposure (market-alignment proxy) |
| Temporal stability | Whether predictive quality fluctuates within evaluation windows | Standard deviation of rolling mean IC and Rank IC |
| Generalization | Whether training performance transfers to validation | Train–validation gaps in IC, Rank IC, ICIR |
| Regime sensitivity | Under which market conditions the signal fails | High-/low-volatility IC gap, up-/down-market gap |
| Prediction-distribution health | Whether the prediction distribution is well-behaved | Cross-sectional standard deviation, skewness, kurtosis |

$m_{\text{valid}})/(|m_{\text{train}}| + \epsilon)$ for selected metrics $m$, with default $\epsilon = 10^{-12}$. *Cross-split summaries* aggregate validation metrics and train–validation gaps across splits. They report mean values and, for selected validation metrics, standard deviations and best/worst split values. *Global diagnostic scores* compress these summaries into compact axis-level signals. Archive quality is computed separately on the shared reference splits, as specified below.

**Regime Definitions.** Regime-conditioned metrics are computed by partitioning evaluation dates into regime subsets. Volatility regimes are defined by median-splitting the 20-observation rolling standard deviation of market returns: dates at or above the median are labeled high-volatility and dates below are low-volatility. Market-direction regimes are defined by the sign of market return, yielding up-market (nonnegative) and down-market (negative) subsets. These partitions are constructed within each training or validation window. Per-regime IC, Rank IC, and top–bottom spread are computed within each subset, and regime gaps measure the difference between paired regimes.

### C.4 Axis Score Computation

The four axis scores aggregate cross-split validation statistics and train–validation gaps into compact signals. Let $\mathcal{V}$ denote the evaluated split collection.

**Predictive strength** $S_{\text{strength}}(\mathcal{V})$: sum of cross-split mean validation Rank IC and mean validation IC. Higher values indicate stronger average predictive performance.

**Market purity** $S_{\text{purity}}(\mathcal{V})$: negative of the cross-split mean absolute validation beta exposure. Here beta exposure is a market-alignment proxy: the temporal correlation between daily mean predictions and trailing five-observation market returns. Configurations with stronger alignment in either direction receive lower scores.

**Robustness** $S_{\text{robust}}(\mathcal{V})$: negative sum of cross-split mean validation rolling Rank IC std and rolling IC std. Each rolling metric is the standard deviation of a 20-observation rolling-mean series. Configurations with temporally unstable performance receive lower scores.

**Generalization** $S_{\text{gen}}(\mathcal{V})$: negative sum of cross-split mean relative gaps in Rank IC, IC, and ICIR. Larger train-to-validation degradation lowers the score; the gaps retain their sign, so improved validation performance can contribute positively.

**Composite diagnostic score.** The scalar diagnostic summary combines the four axes:

$$S_{\text{summary}}(\mathcal{V}) = w_{\text{str}} S_{\text{strength}}(\mathcal{V}) + w_{\text{pur}} S_{\text{purity}}(\mathcal{V}) + w_{\text{rob}} S_{\text{robust}}(\mathcal{V}) + w_{\text{gen}} S_{\text{gen}}(\mathcal{V}),$$

with default weights $(w_{\text{str}}, w_{\text{pur}}, w_{\text{rob}}, w_{\text{gen}}) = (1.0, 1.0, 0.2, 0.05)$, configurable for each run. The full profile uses $\mathcal{V} = \mathcal{V}_{\text{ref}} \cup \mathcal{V}_{\text{diag},i}$.

## D External Metric Computation with a Skill Library

DASH obtains supplemental diagnostic evidence through a reusable library of metric skills. A skill defines a reusable computation, whereas a metric request specifies which computation to perform and the data on which it should operate.

**Metric Library.** Each skill contains an identifier, a description of the measured quantity, category tags, an executable function `compute_metric(df)`, and a default-metric flag. The description supports retrieval, while the function implements the computation. DASH initializes the library with the skills listed in Table 9. For example, the MSE skill has the following structure:

```
MSE:
  description: Weighted mean squared error on valid samples.
  tags: [prediction_error, loss]
  whether_default: true
  expression: |
    def compute_metric(df):
        error = df["yhat"] - df["y"]
        return float(
            np.average(error ** 2, weights=df["w"])
        )
```

Table 9: Initial metric skills in DASH, with descriptions of their computations and diagnostic category tags.

| Skill | Description | Tags |
|---|---|---|
| IC | Mean daily cross-sectional weighted Pearson information coefficient between yhat and y on valid samples. | predictive_power, information_coefficient |
| Rank IC | Mean daily cross-sectional weighted rank information coefficient between yhat and y on valid samples. | predictive_power, rank_ information_coefficient |
| MSE | Weighted mean squared error between yhat and y on valid samples. | prediction_error, loss |
| Mean | Weighted means of yhat, y, and residual yhat minus y on valid samples. | distribution, bias |
| ICIR | Information ratio of the daily cross-sectional weighted Pearson IC time series. | predictive_power, information_ratio |

Continued on next page

Table 9 continued

| Skill | Description | Tags |
|---|---|---|
| Rank ICIR | Information ratio of the daily cross-sectional weighted Rank IC time series. | predictive_power, rank_information_ratio |
| Rank IC Significance | Scope-local significance diagnostics for the daily cross-sectional weighted Rank IC series, including its mean, standard deviation, t-statistic, and positive-day ratio. | predictive_power, rank_information_coefficient, statistical_significance |
| Target Cross-Sectional Distribution | Scope-local daily cross-sectional distribution summary of the realized target y: average mean, standard deviation, skewness, kurtosis, and positive ratio. | target_distribution, dispersion, data_quality |
| Prediction-Target Distribution Comparison | Scope-local comparison of daily cross-sectional yhat and y distributions, including their dispersion ratio, mean difference, skew difference, and positive-ratio difference. | prediction_distribution, target_distribution, calibration |
| Feature Cross-Sectional Dispersion | Scope-local aggregate cross-sectional dispersion of all requested feature columns; feature columns are discovered from the provided dataframe instead of hard-coded. | feature_health, dispersion, data_quality |
| Feature-Target Cross-Sectional Correlation | Scope-local aggregate signal strength of requested feature columns, measured by each feature's mean absolute daily weighted cross-sectional correlation with y. | feature_signal, correlation, input_vs_target |
| Feature Missingness and Degeneracy | Scope-local aggregate missing-value, zero-value, and near-constant rates across all requested numeric feature columns. | feature_health, missingness, data_quality |
| Feature-Prediction Cross-Sectional Correlation | Scope-local model-dependence proxy using each requested feature's mean absolute daily weighted cross-sectional correlation with yhat. | feature_importance, prediction_dependence, correlation |

**Metric request specification.** When $P_i$ does not provide the required evidence, the LLM issues a supplemental metric request with three components:

- *Skill specification.* This specifies which metric to compute through its `name`, `description`, and `tags`, with a registered `skill_id` when a suitable implementation is known.
- *Data specification.* This specifies the data on which to compute the metric through `split`, `date_range`, `universe`, and `columns`.
- *Expectation.* This states the expected metric outcomes that would support or refute the hypothesis. The computed result is compared with these expectations to determine whether the hypothesis should be supported, refuted, or left uncertain.

**Skill retrieval and extension.** The metric library receives the skill specification and uses its `name`, `description`, and `tags` to retrieve a matching metric implementation. If a suitable skill exists, DASH reuses it directly. Otherwise, the LLM generates the requested skill, including its `compute_metric(df)` function. After validation, the new skill is stored in the library for future reuse. Once the skill is available, DASH selects the data according to the data specification and executes the skill on those data. The computed result is then returned for comparison with the request's expectation.

**Token-efficient reuse.** DASH reduces repeated LLM calls by reusing compatible pending refinement plans and diagnostic evidence already available in the current state. Additional reasoning and metric queries are invoked only when the existing plans or evidence are insufficient.

Table 8: Structure of the diagnostic state profile $P_i$. The profile is organized into split-level diagnostics (top), train–validation gap diagnostics, cross-split summaries, and aggregate diagnostic scores (bottom).

| Axis | Scope | Representative metrics |
|---|---|---|
| | *Split-level diagnostics* | |
| Predictive strength | Each split; train/valid | IC mean/std/IR, IC positive rate; Rank IC mean/std/IR, Rank IC positive rate; top–bottom spread mean/std/positive rate |
| Market purity | Each split; train/valid | Signed correlation between daily mean predictions and trailing five-observation market return |
| Temporal stability | Each split; train/valid | Standard deviation of 20-observation rolling mean IC, Rank IC, and spread |
| Prediction-distribution health | Each split; train/valid | Cross-sectional mean and std, std dispersion, skewness, kurtosis, positive-score ratio |
| Regime sensitivity | Each split; train/valid; regime-conditioned subsets | IC, Rank IC, and spread under high-/low-volatility and up-/down-market regimes; high–low volatility and up–down market gaps |
| | *Train–validation gap diagnostics* | |
| Strength gap | Each split; train vs. valid | Relative gaps in IC, Rank IC, ICIR, and spread |
| Purity gap | Each split; train vs. valid | Relative gap in beta exposure / market-alignment proxy |
| Stability gap | Each split; train vs. valid | Relative gaps in IC and Rank IC rolling-mean variability |
| Regime gap | Each split; train vs. valid | Relative train–valid gaps in high–low volatility contrasts (IC, Rank IC, spread) and up–down contrasts (IC, Rank IC) |
| | *Cross-split summaries* | |
| Validation strength | Across validation splits | Mean/std of validation IC, ICIR, Rank IC, and spread; best/worst split IC, ICIR, Rank IC |
| Validation purity | Across validation splits | Signed mean, mean absolute value, maximum, and minimum market-alignment correlation |
| Generalization | Across train–valid gaps | Mean relative gap in IC, Rank IC, ICIR, and spread across splits |
| Robustness | Across validation splits | Mean IC and Rank IC rolling-mean variability across splits |
| | *Aggregate diagnostic scores* | |
| Diagnostic scores | Evaluated split collection | $S_{\mathrm{strength}}, S_{\mathrm{purity}}, S_{\mathrm{robust}}, S_{\mathrm{gen}}, S_{\mathrm{summary}}$ |
| Archive quality | Reference validation only | $q_i = S_{\mathrm{strength}}(\mathcal{V}_{\mathrm{ref}})$; excludes diagnostic splits |

# E Configuration Archive Details

This appendix details the archive structure, admission rule, and parent selection introduced in Section 3.3. All archive scores are computed on the shared reference splits $\mathcal{V}_{\mathrm{ref}}$, providing a common basis for comparing research states.

**Diagnostic cells.** Following the MAP-Elites paradigm (Mouret & Clune, 2015), the archive groups states using discretized behavioral descriptors. In the diagnostic binning scheme described in the main text, these descriptors are

$$\big(S_{\mathrm{gen}}(\mathcal{V}_{\mathrm{ref}}),\ S_{\mathrm{purity}}(\mathcal{V}_{\mathrm{ref}}),\ S_{\mathrm{robust}}(\mathcal{V}_{\mathrm{ref}})\big),$$

with score definitions in Appendix C.4. Bin boundaries are specified in the run configuration and remain fixed during search. Each cell may retain multiple states.

**Fitness score.** Predictive strength uses reference-validation metrics:

$$q_i = \overline{\mathrm{IC}}_i^{\mathrm{ref}} + \overline{\mathrm{Rank\,IC}}_i^{\mathrm{ref}}.$$

Fitness combines this strength with prediction diversity:

$$d_i(\mathcal{A}) = -\max_{s_\ell \in \mathcal{A}\setminus\{s_i\}} \left|\rho_{\mathrm{xs}}\big(\widehat{y}^{(i)}, \widehat{y}^{(\ell)}\big)\right|, \qquad f_i(\mathcal{A}) = w_q q_i + w_d d_i(\mathcal{A}). \tag{4}$$

Here $\rho_{\mathrm{xs}}$ is the mean daily cross-sectional prediction correlation on matched, universe-eligible records from the shared reference splits. The comparison uses stored training and validation predictions, while $q_i$ uses validation metrics only. The weights $w_q$ and $w_d$ are fixed in the run configuration.

**Archive update and selection.** A successfully evaluated successor with valid archive scores must meet the minimum strength threshold $q_i \geq \tau_{\mathrm{admit}}$. For a nonempty archive, admission additionally requires either $d_i(\mathcal{A}) > \tau_{\mathrm{div}}$ or $q_i > q_{\ell^\star}$, where $\ell^\star$ identifies the archived configuration with the largest absolute prediction correlation $|\rho_{\mathrm{xs}}|$ to $C_i$. An admitted state is appended without removing the most similar existing state. Prediction-diversity scores and fitnesses are updated against the enlarged archive. At the start of each iteration, DASH takes up to $d$ states with the highest fitness from each occupied cell, then selects the top $k_{\mathrm{p}}$ candidates globally as parent states. Here $d$ is the per-cell quota and $k_{\mathrm{p}}$ is the parent budget. At an evaluation budget checkpoint or the end of search, DASH selects the eligible archived state with the highest fitness as the final configuration. By default, we set $(w_q, w_d) = (1.0, 0.01)$, $\tau_{\mathrm{admit}} = 0.005$, $\tau_{\mathrm{div}} = -0.8$, $d = 1$ and the parent budget to $k_p = 3$.

# F Experiment Details

## F.1 Baseline Methods

We organize the baselines according to the two search settings defined in Section 3.1. The first group consists of agentic factor-mining methods whose primary search object is the predictive factor. The second group corresponds to the broader joint factor–model setting and contains both standard predictive models and agentic ML research systems. For factor-mining methods, our factor-only method uses a common downstream modeling specification so that differences primarily reflect factor discovery and refinement rather than model choice.

### F.1.1 Factor-Mining Baselines

These methods specialize in iterative discovery and refinement of predictive alpha factors rather than joint optimization of factor construction and downstream predictive models.

- **AlphaAgent.** AlphaAgent (Tang et al., 2025) is an LLM-based framework for alpha factor mining. It iteratively generates and refines candidate factors while using structural and semantic constraints to encourage informative and non-redundant factor proposals.
- **FactorMiner.** FactorMiner (Wang et al., 2026) performs iterative formulaic factor discovery using reusable evaluation capabilities and accumulated experience from previous trials. Its search process uses successful patterns and failure information to guide subsequent factor proposals and reduce redundant exploration.
- **R&D-Agent (Factor).** R&D-Agent (Factor) is our factor-only version of R&D-Agent (Li et al., 2025). The agent retains the original research–development workflow for hypothesis generation, implementation, execution, and feedback, but the editable search space is restricted to factor construction. The downstream model specification is held fixed under the common factor-only benchmark.

### F.1.2 Joint Factor–Model Baselines

The joint setting evaluates methods in the broader forecasting configuration space, where both factor construction and predictive modeling choices may affect performance. We include standard non-agentic predictive models as reference points and agentic systems capable of searching over broader ML configurations.

**(a) Non-Agentic ML Baselines.**

- **XGBoost.** XGBoost (Chen & Guestrin, 2016) is a gradient-boosted tree model that captures nonlinear interactions in tabular financial features.

- **MLP.** The multi-layer perceptron provides a nonlinear feed-forward baseline for learning interactions among input features.
- **GRU.** The gated recurrent unit model (Cho et al., 2014) provides a sequence-modeling baseline capable of capturing temporal dependencies in market data.

These models are evaluated under the feature settings reported in the experimental setup, including raw OHLCV inputs and the Alpha158 feature set (Yang et al., 2020).

**(b) Agentic Joint Factor–Model Baselines.**

- **AutoResearch.** AutoResearch (Karpathy, 2026) automates an iterative code-editing, training, and evaluation loop. Under our forecasting interface, the agent is allowed to modify the editable feature-construction and modeling components and receives validation feedback after each experiment.
- **ML-Master 2.0.** ML-Master 2.0 (Zhu et al., 2026) is a long-horizon ML research agent that accumulates information from execution traces and previous experiments to guide subsequent research decisions. We expose the same editable forecasting components used by the other joint-setting agents.
- **R&D-Agent.** R&D-Agent (Li et al., 2025) jointly explores factor construction and model optimization through research, development, and feedback stages. The research stage proposes hypotheses and tasks, the development stage implements and executes them, and evaluation feedback informs subsequent search decisions.
- **AIRA.** AIRA (Toledo et al., 2025) formulates automated ML research as iterative search over candidate solutions using code-generation, debugging, and improvement operators. In our joint setting, candidate solutions may modify both feature construction and predictive modeling components.

### F.2 Search and Evaluation Configurations

We use separate search and final-evaluation configurations for the joint factor–model and factor-only settings. In both cases, all search decisions and final configuration selection are made using search-period data only; the OOD period remains unseen until final evaluation.

#### F.2.1 Joint factor–model setting.

**Search.** For the ML and agentic baselines, search is conducted using three fixed walk-forward splits:

1. Train: 2005-01-01 to 2014-12-31, Valid: 2015-01-05 to 2016-12-31;
2. Train: 2007-01-01 to 2016-12-31, Valid: 2017-01-05 to 2018-12-31;
3. Train: 2009-01-01 to 2018-12-31, Valid: 2019-01-05 to 2022-12-31.

For these baselines, the search score is the mean validation IC across the three splits. For agentic baselines, this score guides iterative refinement and is also used to select the final configuration. For non-agentic ML baselines, the same score is used for model and hyperparameter selection. DASH selects the eligible configuration with the highest fitness score from its archive for final OOD evaluation, following the selection rule in Appendix E.

**Final OOD evaluation.** The selected configuration is evaluated using three sequential forward OOD cycles, as illustrated in the right panel of Figure 7. In each cycle, the configuration is retrained on an expanding historical window, validated on the most recent available year, and then evaluated on the subsequent non-overlapping forward period. The three cycles are listed in Table 10. The OOD predictions from the three cycles are concatenated for final reporting. No OOD cycle is used during search or configuration selection.

#### F.2.2 Factor-only setting.

**Search.** Factor-only methods search, refine, and select predictive factors using only the train–search period. Candidate factors are evaluated by their IC on a fixed search-period validation window from

Table 10: Forward OOD evaluation cycles for the joint factor–model setting.

| Cycle | Train Range | Valid Range | OOD Test Range |
|---|---|---|---|
| OOD-2023 | 2005-01-01 – 2021-12-31 | 2022-01-05 – 2022-12-31 | 2023-01-05 – 2023-12-31 |
| OOD-2024 | 2005-01-01 – 2022-12-31 | 2023-01-05 – 2023-12-31 | 2024-01-05 – 2024-12-31 |
| OOD-2025 | 2005-01-01 – 2023-12-31 | 2024-01-05 – 2024-12-31 | 2025-01-05 – 2025-10-31 |

2019-01-01 to 2022-12-31. This validation IC serves as the search score for guiding factor refinement and selecting the final factor configuration.

**Final OOD evaluation.** After search, the selected factor configuration is frozen and evaluated directly on the held-out OOD period. Final predictive metrics are computed from the resulting OOD predictions; no OOD information is used to generate, refine, or select factors.

### F.3 Dataset and Prediction Target

We construct daily OHLCV panels for the S&P 500 and NASDAQ-100 universes using a common target-construction procedure. The prediction target is the five-day forward return adjusted for benchmark exposure:

$$r_{t,i}^{(5)} = \frac{P_{t+5,i}}{P_{t,i}} - 1, \qquad y_{t,i} = \mathrm{clip}_{[-0.3,\,0.3]}\left(r_{t,i}^{(5)} - \hat{\beta}_{t,i}\,\tilde{r}_{t,m}^{(5)}\right), \tag{5}$$

where $P_{t,i}$ is the closing price of asset $i$, and $\tilde{r}_{t,m}^{(5)}$ denotes the corresponding five-day benchmark return. We use ETFs tracking the S&P 500 and NASDAQ-100 as benchmarks for their respective universes. The exposure coefficient $\hat{\beta}_{t,i}$ is estimated by OLS as the slope from regressing five-day stock returns on five-day benchmark returns over the preceding 252 daily observations. The estimate is updated monthly.

### F.4 Evaluation Metrics

The main results report two groups of metrics: factor predictive power and portfolio performance.

**Predictive Power Metrics.** Let $\hat{y}_{i,t}$ denote the predicted score and $y_{i,t}$ the realized return for stock $i$ on date $t$, and let $\mathcal{U}_t$ be the cross-section on date $t$. The information coefficient (IC) on date $t$ is the cross-sectional Pearson correlation:

$$\mathrm{IC}_t = \mathrm{corr}_{\mathrm{Pearson}}\big(\{\hat{y}_{i,t}\}_{i\in\mathcal{U}_t},\, \{y_{i,t}\}_{i\in\mathcal{U}_t}\big).$$

The reported IC is its time-average $\overline{\mathrm{IC}} = \frac{1}{T}\sum_{t=1}^{T}\mathrm{IC}_t$, and the IC information ratio is $\mathrm{ICIR} = \overline{\mathrm{IC}}/\mathrm{std}(\mathrm{IC}_t)$, capturing both strength and temporal stability. Rank IC and Rank ICIR replace the Pearson correlation with Spearman rank correlation.

**Portfolio Performance Metrics.** For excess-return portfolio evaluation, metrics are computed from daily *excess returns* relative to the market benchmark. Let $r_t^{\mathrm{p}}$ and $r_t^{\mathrm{b}}$ denote the portfolio and benchmark returns on date $t$, respectively, and define $r_t^{\mathrm{ex}} = r_t^{\mathrm{p}} - r_t^{\mathrm{b}}$. The benchmarks are the corresponding market ETFs for S&P 500 and NASDAQ-100. Over $T$ evaluation days, the mean excess return and its sample standard deviation are

$$\bar{r}^{\mathrm{ex}} = \frac{1}{T}\sum_{t=1}^{T} r_t^{\mathrm{ex}}, \qquad s_{\mathrm{ex}} = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}\big(r_t^{\mathrm{ex}} - \bar{r}^{\mathrm{ex}}\big)^2}.$$

The annualized excess return (ARR) and information ratio (IR) are

$$\mathrm{ARR} = A\bar{r}^{\mathrm{ex}}, \qquad \mathrm{IR} = \sqrt{A}\,\frac{\bar{r}^{\mathrm{ex}}}{s_{\mathrm{ex}}},$$

where $A = 238$ for the equity markets. Thus, ARR uses arithmetic annualization. Define the cumulative summed excess return as $R_t^{\mathrm{ex}} = \sum_{u=1}^{t} r_u^{\mathrm{ex}}$. Maximum drawdown (MDD) is reported as the signed decline from its running peak:

$$\mathrm{MDD} = \min_{1\le t\le T}\left(R_t^{\mathrm{ex}} - \max_{1\le u\le t} R_u^{\mathrm{ex}}\right).$$

Higher ARR and IR are better; MDD is nonpositive, with values closer to zero indicating smaller drawdowns. These metrics use excess returns before transaction costs. Backtest parameters are summarized in Table 11. Absolute NAV for the PIT comparison is defined in Appendix G.4.

Table 11: Backtest configuration for S&P 500 and NASDAQ-100.

| Parameter | S&P 500 | NASDAQ-100 |
|---|---|---|
| Benchmark | S&P 500 | NASDAQ-100 |
| Account size | \$10,000,000 | \$10,000,000 |
| Top-$k$ holdings | 50 | 30 |
| Drop-$n$ | 5 | 3 |
| Limit threshold | 0.0950 | 0.0950 |
| Deal price | Close | Close |
| Open cost | 0.0100% | 0.0100% |
| Close cost | 0.0100% | 0.0100% |

### F.5 Agentic Baseline Initialization

For joint factor–model search, the coding baselines are initialized from a shared XGBoost configuration trained on raw OHLCV sequences. The initial codebase includes the data loading pipeline, an XGBoost training routine, and an evaluation script. Each method can edit the permitted feature construction and modeling components, while the evaluation pipeline and dataset partitioning remain fixed.

### F.6 Sensitivity to LLM Backbones

Table 12: Performance of our method across different LLM backbones on NASDAQ-100 and S&P 500. **Best** results per block are highlighted in bold and second-best are underlined; ties share the same rank.

| LLM Backbone | | Predictive Power Metrics | | | | Performance Metrics | | |
|---|---|---|---|---|---|---|---|---|
| | | **IC** | **ICIR** | **Rank IC** | **Rank ICIR** | **ARR** | **IR** | **MDD** |
| **NASDAQ-100** | GPT-4o | 0.0217 | 0.1402 | 0.0062 | 0.0428 | 0.0146 | 0.1361 | -0.1287 |
| | GPT-4.1 | **0.0319** | **0.2294** | **0.0291** | **0.2207** | 0.1046 | **1.5581** | **-0.0474** |
| | GPT-5-mini | 0.0244 | 0.1849 | 0.0109 | 0.0782 | **0.1685** | 1.0252 | -0.1244 |
| | GPT-5.4 | 0.0271 | 0.1497 | 0.0134 | 0.0920 | 0.0348 | 0.2718 | -0.1377 |
| | GPT-5.6 | 0.0273 | 0.1673 | 0.0180 | 0.1070 | 0.0050 | 0.0457 | -0.1851 |
| | DeepSeek-V4-Pro | 0.0210 | 0.1510 | 0.0170 | 0.1150 | 0.0612 | 0.5563 | -0.1418 |
| **S&P 500** | GPT-4o | 0.0136 | 0.1080 | 0.0061 | 0.0449 | -0.0069 | -0.0601 | -0.1460 |
| | GPT-4.1 | 0.0180 | 0.1373 | 0.0109 | 0.0767 | 0.0520 | 0.4505 | **-0.0982** |
| | GPT-5-mini | 0.0200 | 0.1331 | 0.0093 | 0.0617 | **0.0692** | **0.9892** | -0.1009 |
| | GPT-5.4 | **0.0208** | 0.1497 | **0.0134** | **0.0920** | 0.0538 | 0.8072 | -0.0993 |
| | GPT-5.6 | 0.0206 | **0.1537** | 0.0093 | 0.0663 | 0.0554 | 0.4429 | -0.1010 |
| | DeepSeek-V4-Pro | 0.0145 | 0.1221 | 0.0084 | 0.0611 | -0.0068 | -0.0694 | -0.1091 |

Table 12 examines how DASH's OOD performance varies across six LLM backbones. On NASDAQ-100, GPT-4.1 leads all four predictive metrics, whereas GPT-5-mini achieves the highest ARR. On S&P 500, GPT-5.4 leads IC, Rank IC, and Rank ICIR, GPT-5.6 achieves the highest ICIR, and GPT-5-mini leads ARR and portfolio IR. Predictive and portfolio rankings also differ: on S&P 500, GPT-5.6 has a higher ICIR than GPT-5-mini (0.1537 versus 0.1331) but a lower portfolio IR (0.4429 versus 0.9892). These results show that the observed backbone rankings depend on both the equity universe and the evaluation metric, with no single backbone dominating across all settings.

# G Experiment Results

## G.1 Factor-Only Results

### G.1.1 Token-Matched Comparison

Tables 13–14 and Tables 15–16 report the token-matched factor-only results on NASDAQ-100 and S&P 500 at the 200K- and 400K-token budgets, respectively. Each table evaluates the best individual factor together with Top-5, Top-20, and Top-40 equal-weight and rolling-linear ensembles.

**Token-matched evaluation protocol.** For each method, when cumulative LLM token usage first exceeds a specified budget, we record a snapshot of previously completed valid factors, excluding outputs from the proposal that crosses the threshold. Search continues, but evaluation at that budget uses only this frozen snapshot. We then rank these factors by absolute IC on the common validation split. The best-factor result uses the highest-ranked factor; each Top-$K$ ensemble uses up to $K$ available factors, with $K \in \{5, 20, 40\}$. Equal-weight ensembles average the selected factors after daily cross-sectional z-score normalization, while rolling-linear ensembles fit a linear model using the same selected factors under the prescribed rolling evaluation protocol. Finally, the best factor and ensembles are evaluated on the held-out OOD period. OOD results are not used for factor selection, direction assignment, or further search.

**NASDAQ-100.** At 200K tokens, DASH leads all four metrics under every aggregation strategy (Table 13). The best factor reaches IC 0.0307 and ICIR 0.1650 and remains unchanged at 400K tokens. Additional search improves the Top-5 equal-weight ensemble's IC from 0.0269 to 0.0353. At 400K, DASH retains the highest IC under every aggregation strategy, while the stability advantage varies: R&D-Agent achieves a higher best-factor ICIR of 0.1853 (Table 15).

**S&P 500.** DASH's best factor leads all four metrics at both budgets, with IC increasing from 0.0159 to 0.0170 and ICIR from 0.1262 to 0.1297. At 400K, DASH also leads all four metrics under every ensemble strategy (Table 16). At 200K, the advantage depends on aggregation: FactorMiner's Top-40 rolling-linear ensemble achieves higher IC and ICIR, at 0.0219 and 0.1964, respectively (Table 14).

**Effect of aggregation.** Combining factors can improve IC over the best individual factor, but the gain depends on the weighting scheme and evaluation metric. For example, on S&P 500 at 400K, DASH's Top-20 rolling-linear ensemble raises IC from 0.0170 to 0.0208, while Rank IC decreases from 0.0156 to 0.0095. Thus, larger ensembles do not uniformly improve all aspects of predictive performance.

Table 13: Factor-only out-of-distribution predictive performance on NASDAQ-100 at the 200K-token budget under seven factor aggregation strategies. Higher is better. **Bold**: best; underline: second best within each aggregation strategy.

| Aggregation | Method | IC | Rank IC | ICIR | Rank ICIR |
|---|---|---|---|---|---|
| Best factor | **Ours** | **0.0307** | **0.0311** | **0.1650** | **0.1675** |
| | FactorMiner | 0.0064 | 0.0091 | 0.0432 | 0.0570 |
| | AlphaAgent | 0.0144 | 0.0231 | 0.1156 | 0.1260 |
| | R&D-Agent | 0.0131 | 0.0141 | 0.0954 | 0.0911 |
| Top-5 equal weight | **Ours** | **0.0269** | **0.0303** | **0.1440** | **0.1630** |
| | FactorMiner | 0.0200 | 0.0144 | 0.1163 | 0.0776 |
| | AlphaAgent | 0.0131 | 0.0159 | 0.1008 | 0.0858 |
| | R&D-Agent | 0.0055 | 0.0040 | 0.0412 | 0.0292 |
| Top-5 rolling linear | **Ours** | **0.0363** | **0.0251** | **0.1960** | **0.1364** |
| | FactorMiner | 0.0213 | 0.0183 | 0.1290 | 0.1088 |
| | AlphaAgent | 0.0138 | 0.0164 | 0.1055 | 0.0885 |
| | R&D-Agent | 0.0094 | 0.0065 | 0.0661 | 0.0423 |

| Aggregation | Method | IC | Rank IC | ICIR | Rank ICIR |
|---|---|---|---|---|---|
| Top-20 equal weight | **Ours** | **0.0269** | **0.0303** | **0.1440** | **0.1630** |
| | FactorMiner | 0.0198 | 0.0113 | 0.1112 | 0.0591 |
| | AlphaAgent | 0.0131 | 0.0159 | 0.1008 | 0.0858 |
| | R&D-Agent | 0.0183 | 0.0180 | 0.1386 | 0.1304 |
| Top-20 rolling linear | **Ours** | **0.0363** | **0.0251** | **0.1960** | **0.1364** |
| | FactorMiner | 0.0197 | 0.0155 | 0.1271 | 0.0954 |
| | AlphaAgent | 0.0138 | 0.0164 | 0.1055 | 0.0885 |
| | R&D-Agent | 0.0177 | 0.0159 | 0.1249 | 0.1074 |
| Top-40 equal weight | **Ours** | **0.0269** | **0.0303** | **0.1440** | **0.1630** |
| | FactorMiner | 0.0209 | 0.0124 | 0.1155 | 0.0644 |
| | AlphaAgent | 0.0131 | 0.0159 | 0.1008 | 0.0858 |
| | R&D-Agent | 0.0177 | 0.0168 | 0.1333 | 0.1250 |
| Top-40 rolling linear | **Ours** | **0.0363** | **0.0251** | **0.1960** | **0.1364** |
| | FactorMiner | 0.0206 | 0.0180 | 0.1378 | 0.1111 |
| | AlphaAgent | 0.0138 | 0.0164 | 0.1055 | 0.0885 |
| | R&D-Agent | 0.0289 | 0.0138 | 0.1899 | 0.0983 |

Table 14: Factor-only out-of-distribution predictive performance on S&P 500 at the 200K-token budget under seven factor aggregation strategies. **Bold**: best; underline: second best.

| Aggregation | Method | IC | Rank IC | ICIR | Rank ICIR |
|---|---|---|---|---|---|
| Best factor | **Ours** | **0.0159** | **0.0141** | **0.1262** | **0.1016** |
| | FactorMiner | 0.0096 | 0.0081 | 0.0818 | 0.0636 |
| | AlphaAgent | −0.0063 | −0.0070 | −0.0465 | −0.0466 |
| | R&D-Agent | −0.0001 | −0.0011 | −0.0013 | −0.0130 |
| Top-5 equal weight | **Ours** | **0.0151** | **0.0133** | **0.1105** | **0.0898** |
| | FactorMiner | 0.0061 | 0.0029 | 0.0385 | 0.0170 |
| | AlphaAgent | −0.0063 | −0.0070 | −0.0465 | −0.0466 |
| | R&D-Agent | −0.0043 | −0.0038 | −0.0433 | −0.0361 |
| Top-5 rolling linear | **Ours** | **0.0145** | **0.0129** | **0.1051** | **0.0856** |
| | FactorMiner | 0.0069 | 0.0068 | 0.0517 | 0.0471 |
| | AlphaAgent | −0.0063 | −0.0070 | −0.0465 | −0.0466 |
| | R&D-Agent | −0.0012 | −0.0037 | −0.0132 | −0.0352 |
| Top-20 equal weight | **Ours** | **0.0191** | **0.0129** | **0.1352** | 0.0858 |
| | FactorMiner | 0.0067 | 0.0029 | 0.0418 | 0.0165 |
| | AlphaAgent | −0.0063 | −0.0070 | −0.0465 | −0.0466 |
| | R&D-Agent | 0.0021 | 0.0110 | 0.0192 | **0.0931** |
| Top-20 rolling linear | **Ours** | **0.0202** | 0.0074 | **0.1376** | 0.0493 |
| | FactorMiner | 0.0103 | **0.0088** | 0.0831 | **0.0663** |
| | AlphaAgent | −0.0063 | −0.0070 | −0.0465 | −0.0466 |
| | R&D-Agent | 0.0025 | −0.0031 | 0.0202 | −0.0220 |
| Top-40 equal weight | **Ours** | **0.0191** | **0.0129** | **0.1352** | **0.0858** |
| | FactorMiner | 0.0060 | 0.0032 | 0.0385 | 0.0188 |
| | AlphaAgent | −0.0063 | −0.0070 | −0.0465 | −0.0466 |
| | R&D-Agent | 0.0003 | 0.0095 | 0.0030 | 0.0798 |
| Top-40 rolling linear | **Ours** | 0.0202 | 0.0074 | 0.1376 | 0.0493 |
| | FactorMiner | **0.0219** | **0.0137** | **0.1964** | **0.1114** |
| | AlphaAgent | −0.0063 | −0.0070 | −0.0465 | −0.0466 |
| | R&D-Agent | 0.0060 | −0.0005 | 0.0575 | −0.0044 |

Table 15: Factor-only out-of-distribution predictive performance on NASDAQ-100 at the 400K-token budget under seven factor aggregation strategies. Higher is better. **Bold**: best; underline: second best within each aggregation strategy.

| Aggregation | Method | IC | Rank IC | ICIR | Rank ICIR |
|---|---|---|---|---|---|
| Best factor | **Ours** | **0.0307** | **0.0311** | 0.1650 | **0.1675** |
| | FactorMiner | 0.0064 | 0.0091 | 0.0432 | 0.0570 |
| | AlphaAgent | 0.0144 | 0.0231 | 0.1156 | 0.1260 |
| | R&D-Agent | 0.0263 | 0.0240 | **0.1853** | 0.1613 |
| Top-5 equal weight | **Ours** | **0.0353** | **0.0338** | 0.1879 | **0.1793** |
| | FactorMiner | 0.0103 | 0.0061 | 0.0710 | 0.0366 |
| | AlphaAgent | 0.0202 | 0.0299 | 0.1491 | 0.1662 |
| | R&D-Agent | 0.0293 | 0.0295 | **0.1889** | 0.1755 |
| Top-5 rolling linear | **Ours** | **0.0339** | **0.0317** | **0.1844** | 0.1692 |
| | FactorMiner | 0.0081 | 0.0127 | 0.0565 | 0.0768 |
| | AlphaAgent | 0.0206 | 0.0317 | 0.1506 | **0.1818** |
| | R&D-Agent | 0.0265 | 0.0285 | 0.1758 | 0.1715 |
| Top-20 equal weight | **Ours** | **0.0279** | **0.0301** | 0.1522 | 0.1641 |
| | FactorMiner | 0.0182 | 0.0108 | 0.1096 | 0.0574 |
| | AlphaAgent | 0.0202 | 0.0299 | 0.1491 | **0.1662** |
| | R&D-Agent | 0.0279 | 0.0195 | **0.2078** | 0.1402 |
| Top-20 rolling linear | **Ours** | **0.0347** | 0.0240 | **0.1874** | 0.1301 |
| | FactorMiner | 0.0205 | 0.0185 | 0.1352 | 0.1140 |
| | AlphaAgent | 0.0206 | **0.0317** | 0.1506 | **0.1818** |
| | R&D-Agent | 0.0083 | 0.0166 | 0.0536 | 0.1048 |
| Top-40 equal weight | **Ours** | **0.0279** | **0.0301** | 0.1522 | 0.1641 |
| | FactorMiner | 0.0192 | 0.0109 | 0.1113 | 0.0576 |
| | AlphaAgent | 0.0202 | 0.0299 | 0.1491 | 0.1662 |
| | R&D-Agent | 0.0264 | 0.0250 | **0.1948** | **0.1778** |
| Top-40 rolling linear | **Ours** | **0.0347** | 0.0240 | **0.1874** | 0.1301 |
| | FactorMiner | 0.0234 | 0.0207 | 0.1547 | 0.1306 |
| | AlphaAgent | 0.0206 | **0.0317** | 0.1506 | **0.1818** |
| | R&D-Agent | 0.0097 | 0.0140 | 0.0674 | 0.0932 |

Table 16: Factor-only out-of-distribution predictive performance on S&P 500 at the 400K-token budget under seven factor aggregation strategies. Higher is better. **Bold**: best; underline: second best within each aggregation strategy.

| Aggregation | Method | IC | Rank IC | ICIR | Rank ICIR |
|---|---|---|---|---|---|
| Best factor | **Ours** | **0.0170** | **0.0156** | **0.1297** | **0.1110** |
| | FactorMiner | 0.0096 | 0.0081 | 0.0818 | 0.0636 |
| | AlphaAgent | −0.0063 | −0.0070 | −0.0465 | −0.0466 |
| | R&D-Agent | −0.0001 | −0.0011 | −0.0013 | −0.0130 |
| Top-5 equal weight | **Ours** | **0.0158** | **0.0138** | **0.1084** | **0.0916** |
| | FactorMiner | 0.0093 | 0.0073 | 0.0703 | 0.0515 |
| | AlphaAgent | −0.0063 | −0.0070 | −0.0465 | −0.0466 |
| | R&D-Agent | −0.0043 | −0.0038 | −0.0433 | −0.0361 |
| Top-5 rolling linear | **Ours** | **0.0167** | **0.0146** | **0.1231** | **0.1023** |
| | FactorMiner | 0.0068 | 0.0059 | 0.0496 | 0.0393 |
| | AlphaAgent | −0.0063 | −0.0070 | −0.0465 | −0.0466 |
| | R&D-Agent | −0.0012 | −0.0037 | −0.0132 | −0.0352 |

| Aggregation | Method | IC | Rank IC | ICIR | Rank ICIR |
|---|---|---|---|---|---|
| Top-20 equal weight | **Ours** | **0.0180** | **0.0147** | **0.1267** | **0.0983** |
| | FactorMiner | 0.0069 | 0.0036 | 0.0445 | 0.0220 |
| | AlphaAgent | −0.0063 | −0.0070 | −0.0465 | −0.0466 |
| | R&D-Agent | 0.0011 | 0.0108 | 0.0100 | 0.0897 |
| Top-20 rolling linear | **Ours** | **0.0208** | **0.0095** | **0.1396** | **0.0632** |
| | FactorMiner | 0.0049 | 0.0054 | 0.0353 | 0.0356 |
| | AlphaAgent | −0.0063 | −0.0070 | −0.0465 | −0.0466 |
| | R&D-Agent | 0.0021 | −0.0037 | 0.0175 | −0.0266 |
| Top-40 equal weight | **Ours** | **0.0192** | **0.0149** | **0.1336** | **0.0985** |
| | FactorMiner | 0.0058 | 0.0021 | 0.0365 | 0.0126 |
| | AlphaAgent | −0.0063 | −0.0070 | −0.0465 | −0.0466 |
| | R&D-Agent | −0.0015 | 0.0076 | −0.0131 | 0.0640 |
| Top-40 rolling linear | **Ours** | **0.0204** | **0.0070** | **0.1393** | **0.0472** |
| | FactorMiner | 0.0034 | 0.0016 | 0.0287 | 0.0125 |
| | AlphaAgent | −0.0063 | −0.0070 | −0.0465 | −0.0466 |
| | R&D-Agent | 0.0050 | −0.0008 | 0.0481 | −0.0065 |

### G.1.2 Search Efficiency

Figures 8 and 9 report the fraction of valid generated factors whose OOD IC exceeds each specific threshold. Fractions are computed over factors with sufficient evaluation coverage, with factor directions fixed using validation data. OOD measurements are used only for this post-hoc analysis.

**S&P 500.** DASH achieves the highest fraction across all displayed budgets and thresholds. At 400K tokens, 43.1% of its factors exceed OOD IC 0.01, compared with 16.3% for FactorMiner. At the stricter 0.015 threshold, the corresponding fractions are 29.3% and 4.9%. This indicates a higher concentration of strong OOD factors among the valid candidates produced under the same token budget.

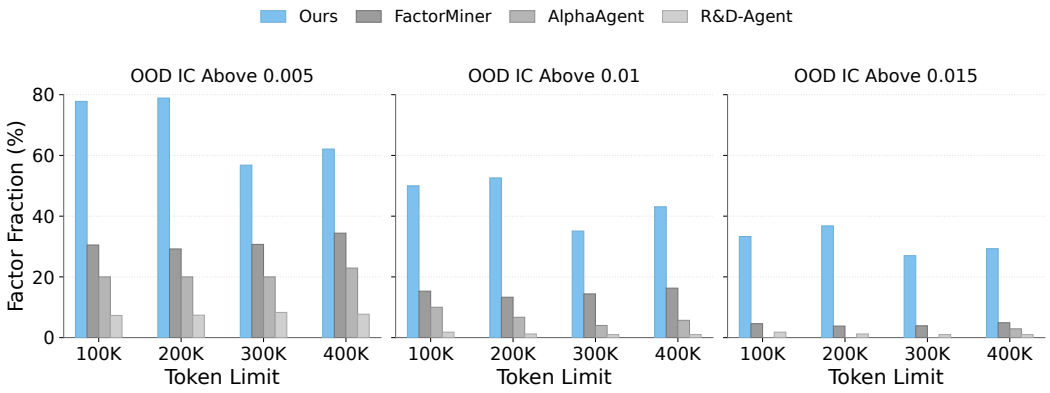


Figure 8: Percentage of OOD-evaluable valid factors with OOD IC above 0.005, 0.01, and 0.015 under different token budgets for S&P 500.

**NASDAQ-100.** DASH's advantage is clearest at the higher thresholds: it leads at OOD IC above 0.02 throughout the displayed budgets. At 400K, 26.5% of its factors exceed 0.02 and 23.5% exceed 0.03, compared with the strongest baseline fractions of 10.1% and 7.1%, respectively. At the looser 0.01 threshold, FactorMiner leads from 100K to 300K, while DASH takes the lead at 400K. The results therefore support DASH's ability to discover a larger proportion of factors with particularly strong OOD predictive power.

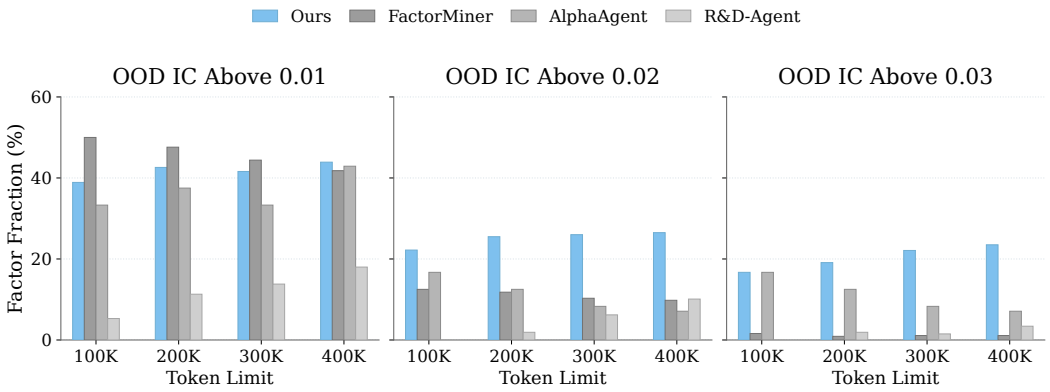


Figure 9: Percentage of OOD-evaluable valid factors with OOD IC above 0.01, 0.02, and 0.03 under different token budgets for NASDAQ-100.

These results demonstrate DASH's search efficiency: under a fixed token budget, it discovers a higher proportion of high-quality factors with strong OOD predictive performance.

### G.1.3 Diversity Analysis

We examine whether the discovered factors provide distinct signals and how strongly they correlate with the Alpha158 feature library (Tables 17 and 18).

**Metrics.** *Diversity* is one minus the mean absolute correlation across distinct generated-factor pairs. *Redundancy* is the percentage of these pairs with absolute correlation greater than 0.8. *Alpha158 Correlation* averages the absolute correlations between generated factors and evaluable features in the Alpha158 library. For each pair, correlation is averaged across daily cross-sections before taking its absolute value. Higher diversity and lower redundancy or library correlation are preferred. These metrics use the 2023–2025 OOD period for post-hoc analysis only.

**Results.** DASH maintains a diverse factor pool on both markets, with diversity scores of 0.8403 on NASDAQ-100 and 0.8388 on S&P 500. Only 2.50% and 2.40% of factor pairs, respectively, exceed the 0.8 correlation threshold, indicating little duplication through highly correlated signals. Its average correlations with Alpha158 are also low, at 0.1325 and 0.1384, indicating limited linear overlap with the existing feature library. Together with the preceding predictive and search-efficiency results, these findings show that DASH discovers high-quality factors while maintaining signal diversity and low redundancy.

Table 17: Factor diversity on NASDAQ-100 at the 400K-token budget.

| **Method** | **Diversity** ↑ | **Redundancy** $> 0.8$ ↓ | **Alpha158 Correlation** ↓ |
|---|---|---|---|
| **Ours** | 0.8403 | 2.50% | 0.1325 |
| FactorMiner | 0.8741 | 1.00% | 0.1266 |
| AlphaAgent | 0.7867 | 12.10% | 0.1344 |
| R&D-Agent | 0.9041 | 2.70% | 0.0898 |

Table 18: Factor diversity on S&P 500 at the 400K-token budget.

| **Method** | **Diversity** ↑ | **Redundancy** $> 0.8$ ↓ | **Alpha158 Correlation** ↓ |
|---|---|---|---|
| **Ours** | 0.8388 | 2.40% | 0.1384 |
| FactorMiner | 0.8635 | 1.40% | 0.1264 |
| AlphaAgent | 0.6005 | 39.00% | 0.0638 |
| R&D-Agent | 0.9236 | 0.70% | 0.0552 |

## G.2 Factor–Model Coevolution Results

### G.2.1 Results Across Multiple Runs

For each agentic method and LLM backbone, we conduct three independent runs. Each run consists of eight search iterations, with at most five configuration proposals allowed per iteration. Tables 19 and 20 summarize the means and standard deviations across the three runs, with standard deviations in parentheses below the means.

Table 19: Complete out-of-distribution predictive and portfolio results on NASDAQ-100 across multiple runs. Means are shown above standard deviations in parentheses. **Bold** and underline mark the best and second-best means. Higher is better; MDD values closer to zero indicate smaller drawdown.

| | Models | Prediction Metrics | | | | Portfolio Metrics | | |
|---|---|---|---|---|---|---|---|---|
| | | IC | ICIR | Rank IC | Rank ICIR | ARR | IR | MDD |
| ML Baselines | $\text{MLP}_{\text{OHLCV}}$ | 0.0122 (0.0059) | 0.0907 (0.0456) | 0.0105 (0.0079) | 0.0773 (0.0588) | −0.0162 (0.0164) | −0.1455 (0.1391) | −0.1795 (0.0221) |
| | $\text{MLP}_{\text{Alpha158}}$ | 0.0109 (0.0021) | 0.0755 (0.0122) | 0.0040 (0.0047) | 0.0283 (0.0333) | 0.0167 (0.0164) | 0.1768 (0.1732) | −0.1400 (0.0089) |
| | $\text{XGBoost}_{\text{OHLCV}}$ | 0.0062 (0.0026) | 0.0448 (0.0188) | 0.0011 (0.0028) | 0.0076 (0.0187) | 0.0257 (0.0053) | 0.2217 (0.0434) | −0.1692 (0.0103) |
| | $\text{XGBoost}_{\text{Alpha158}}$ | **0.0323** (0.0042) | 0.2071 (0.0235) | 0.0230 (0.0056) | 0.1578 (0.0367) | **0.1068** (0.0051) | 0.8299 (0.6723) | −0.0957 (0.0197) |
| | $\text{GRU}_{\text{OHLCV}}$ | 0.0143 (0.0120) | 0.1141 (0.0946) | 0.0113 (0.0024) | 0.0893 (0.0154) | −0.0008 (0.0877) | 0.0040 (0.8751) | −0.1844 (0.0802) |
| | $\text{GRU}_{\text{Alpha158}}$ | 0.0044 (0.0189) | 0.0375 (0.1567) | 0.0066 (0.0167) | 0.0557 (0.1417) | −0.0459 (0.0649) | −0.5428 (0.7631) | −0.2470 (0.1110) |
| Agentic Frameworks | $\text{AutoResearch}_{\text{GPT-4.1}}$ | 0.0234 (0.0007) | 0.1450 (0.0055) | 0.0181 (0.0009) | 0.1172 (0.0059) | −0.0925 (0.1168) | −0.6417 (0.7035) | −0.4058 (0.3005) |
| | $\text{AutoResearch}_{\text{GPT-5-mini}}$ | 0.0176 (0.0042) | 0.1254 (0.0415) | 0.0148 (0.0014) | 0.0995 (0.0115) | −0.0316 (0.0156) | −0.2929 (0.1464) | −0.2382 (0.0317) |
| | $\text{R\&D-Agent}_{\text{GPT-4.1}}$ | 0.0287 (0.0006) | 0.1833 (0.0095) | 0.0186 (0.0023) | 0.1224 (0.0143) | 0.0904 (0.0368) | 0.8325 (0.3454) | −0.1452 (0.0154) |
| | $\text{R\&D-Agent}_{\text{GPT-5-mini}}$ | 0.0251 (0.0027) | 0.1691 (0.0204) | 0.0217 (0.0010) | 0.1414 (0.0052) | 0.0639 (0.0030) | 0.5903 (0.0390) | −0.1545 (0.0148) |
| | $\text{AIRA}_{\text{GPT-4.1}}$ | 0.0169 (0.0031) | 0.1171 (0.0190) | 0.0115 (0.0072) | 0.0756 (0.0455) | −0.0033 (0.0612) | −0.0265 (0.6028) | −0.1778 (0.0903) |
| | $\text{AIRA}_{\text{GPT-5-mini}}$ | 0.0165 (0.0012) | 0.1209 (0.0131) | 0.0097 (0.0047) | 0.0665 (0.0300) | 0.0134 (0.0248) | 0.1392 (0.2452) | −0.1461 (0.0394) |
| | $\text{ML-Master2.0}_{\text{GPT-4.1}}$ | 0.0272 (0.0014) | 0.1634 (0.0042) | 0.0179 (0.0008) | 0.1151 (0.0045) | 0.0553 (0.0328) | 0.5287 (0.2585) | −0.1321 (0.0158) |
| | $\text{ML-Master2.0}_{\text{GPT-5-mini}}$ | 0.0234 (0.0039) | 0.1425 (0.0164) | 0.0113 (0.0086) | 0.0730 (0.0556) | 0.0301 (0.0969) | 0.2266 (0.8699) | −0.1866 (0.0827) |
| | **Ours**$_{\text{GPT-4.1}}$ | 0.0312 (0.0013) | **0.2243** (0.0088) | **0.0275** (0.0028) | **0.2089** (0.0204) | 0.0569 (0.0441) | **1.0428** (0.0417) | **−0.0887** (0.0470) |
| | **Ours**$_{\text{GPT-5-mini}}$ | 0.0301 (0.0051) | 0.2116 (0.0236) | 0.0155 (0.0082) | 0.1056 (0.0503) | 0.0899 (0.0699) | 0.9801 (0.5754) | −0.1010 (0.0331) |

**NASDAQ-100.** DASH with GPT-4.1 leads ICIR, Rank IC, Rank ICIR, and IR, and achieves the smallest drawdown (MDD $-0.0887$). Its IR of 1.0428 has a standard deviation of 0.0417 across runs, compared with $0.8325 \pm 0.3454$ for R&D-Agent with GPT-4.1. $\text{XGBoost}_{\text{Alpha158}}$ has higher IC and ARR, so DASH's advantage is clearest in predictive stability and risk-adjusted portfolio performance.

**S&P 500.** DASH with GPT-5-mini leads all seven metrics, combining the strongest predictive performance with ARR 0.0893, IR 1.1214, and the smallest drawdown (MDD $-0.0871$). Its GPT-4.1 variant achieves the second-smallest drawdown ($-0.0994$), although its predictive and return metrics are weaker. Thus, the best-performing backbone differs across markets, while DASH achieves the highest IR and smallest drawdown on both.

Table 20: Complete out-of-distribution predictive and portfolio results on S&P 500 across multiple runs. Means are shown above standard deviations in parentheses. **Bold** and underline mark the best and second-best means. Higher is better; MDD values closer to zero indicate smaller drawdown.

| | Models | Prediction Metrics | | | | Portfolio Metrics | | |
|---|---|---|---|---|---|---|---|---|
| | | IC | ICIR | Rank IC | Rank ICIR | ARR | IR | MDD |
| ML Baselines | $\text{MLP}_{\text{OHLCV}}$ | 0.0118 (0.0004) | 0.1301 (0.0097) | <u>0.0120</u> (0.0002) | <u>0.1133</u> (0.0176) | −0.0104 (0.0223) | −0.1043 (0.2461) | −0.1031 (0.0382) |
| | $\text{MLP}_{\text{Alpha158}}$ | 0.0066 (0.0011) | 0.0613 (0.0120) | 0.0068 (0.0043) | 0.0593 (0.0374) | −0.0568 (0.0143) | −0.6105 (0.1364) | −0.1845 (0.0411) |
| | $\text{XGBoost}_{\text{OHLCV}}$ | 0.0036 (0.0014) | 0.0379 (0.0150) | −0.0021 (0.0014) | −0.0199 (0.0136) | −0.0210 (0.0155) | −0.2270 (0.1683) | −0.1292 (0.0233) |
| | $\text{XGBoost}_{\text{Alpha158}}$ | 0.0108 (0.0011) | 0.0943 (0.0078) | 0.0047 (0.0030) | 0.0414 (0.0259) | 0.0260 (0.0170) | 0.2494 (0.1475) | −0.1249 (0.0164) |
| | $\text{GRU}_{\text{OHLCV}}$ | 0.0101 (0.0030) | 0.1128 (0.0364) | 0.0099 (0.0035) | 0.0968 (0.0314) | −0.0133 (0.0183) | −0.1927 (0.2446) | −0.1492 (0.0444) |
| | $\text{GRU}_{\text{Alpha158}}$ | 0.0111 (0.0055) | 0.1383 (0.0744) | 0.0055 (0.0028) | 0.0671 (0.0380) | −0.0003 (0.0235) | −0.0023 (0.2545) | −0.1308 (0.0502) |
| Agentic Frameworks | $\text{AutoResearch}_{\text{GPT-4.1}}$ | 0.0146 (0.0004) | 0.1162 (0.0019) | 0.0064 (0.0004) | 0.0477 (0.0024) | <u>0.0657</u> (0.0079) | <u>0.5180</u> (0.0624) | −0.1068 (0.0165) |
| | $\text{AutoResearch}_{\text{GPT-5-mini}}$ | 0.0123 (0.0054) | 0.0933 (0.0411) | 0.0016 (0.0095) | 0.0078 (0.0727) | 0.0393 (0.0205) | 0.3035 (0.1424) | −0.1546 (0.0360) |
| | $\text{R\&D-Agent}_{\text{GPT-4.1}}$ | <u>0.0178</u> (0.0016) | <u>0.1424</u> (0.0090) | 0.0105 (0.0002) | 0.0782 (0.0011) | 0.0213 (0.0293) | 0.1954 (0.2714) | −0.1167 (0.0118) |
| | $\text{R\&D-Agent}_{\text{GPT-5-mini}}$ | 0.0163 (0.0015) | 0.1254 (0.0160) | 0.0065 (0.0008) | 0.0468 (0.0077) | 0.0235 (0.0208) | 0.2004 (0.1692) | −0.1177 (0.0292) |
| | $\text{AIRA}_{\text{GPT-4.1}}$ | 0.0119 (0.0034) | 0.1081 (0.0256) | 0.0039 (0.0039) | 0.0303 (0.0299) | −0.0404 (0.0061) | −0.3977 (0.0833) | −0.1945 (0.0169) |
| | $\text{AIRA}_{\text{GPT-5-mini}}$ | 0.0118 (0.0026) | 0.1038 (0.0206) | 0.0041 (0.0002) | 0.0323 (0.0009) | −0.0324 (0.0380) | −0.3175 (0.3672) | −0.1844 (0.0603) |
| | $\text{ML-Master2.0}_{\text{GPT-4.1}}$ | 0.0172 (0.0031) | 0.1401 (0.0278) | 0.0098 (0.0021) | 0.0750 (0.0191) | 0.0054 (0.0393) | 0.0399 (0.3596) | −0.1373 (0.0344) |
| | $\text{ML-Master2.0}_{\text{GPT-5-mini}}$ | 0.0139 (0.0052) | 0.1009 (0.0390) | 0.0060 (0.0029) | 0.0422 (0.0211) | −0.0052 (0.0619) | −0.0801 (0.5159) | −0.1736 (0.0523) |
| | **Ours**$_{\text{GPT-4.1}}$ | 0.0175 (0.0005) | 0.1377 (0.0010) | 0.0101 (0.0008) | 0.0731 (0.0034) | 0.0361 (0.0267) | 0.3150 (0.2342) | −0.0994 (0.0131) |
| | **Ours**$_{\text{GPT-5-mini}}$ | **0.0246** (0.0057) | **0.1996** (0.0381) | **0.0185** (0.0027) | **0.1478** (0.0579) | **0.0893** (0.0260) | **1.1214** (0.1819) | **−0.0871** (0.0180) |

### G.2.2 Token-Matched Comparison

Table 21 compares joint factor–model search at 100K, 200K, and 400K cumulative LLM tokens. Matching token budgets accounts for the additional LLM work used for diagnosis and metric-grounded assessment, complementing the iteration-matched comparison in the main text.

DASH achieves the highest IR and ARR in all six market–budget comparisons, with a substantial advantage in OOD portfolio performance.

**NASDAQ-100.** At 100K tokens, DASH achieves IR 0.1907 and ARR 0.0211, compared with 0.0215 and 0.0023 for R&D-Agent, the strongest baseline on both metrics. At 200K and 400K, DASH is the only method with positive IR and ARR; all baselines have negative excess returns. Thus, the portfolio advantage holds across all three token limits.

**S&P 500.** At both 100K and 200K tokens, DASH's IR of 0.4128 is approximately 2.1 times the strongest baseline's 0.1933, while its ARR of 0.0491 is nearly twice the best baseline ARR of 0.0250. At 400K, DASH further improves to IR 0.6836 and ARR 0.0945, compared with 0.3938 and 0.0489 for R&D-Agent. The substantial advantage therefore persists as the token budget increases.

On both markets, DASH at 100K tokens already surpasses every baseline at 400K tokens in both IR and ARR. This indicates that DASH reaches strong OOD portfolio performance substantially earlier in the search, despite spending additional tokens on diagnosis and hypothesis assessment. The result

Table 21: Token-matched factor–model coevolution results at the 100K-, 200K-, and 400K-token budgets. **Bold**: best; underline: second best within each market and budget.

| | | NASDAQ-100 | | | | S&P 500 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Method** | **IC** | **ICIR** | **IR** | **ARR** | **IC** | **ICIR** | **IR** | **ARR** |
| 100K tokens | AutoResearch | 0.0111 | 0.0677 | −0.7090 | −0.0734 | 0.0115 | 0.0880 | 0.1770 | 0.0207 |
| | R&D-Agent | 0.0203 | 0.1407 | 0.0215 | 0.0023 | 0.0102 | 0.0926 | −0.2477 | −0.0246 |
| | AIRA | 0.0172 | 0.1170 | −0.0154 | −0.0017 | **0.0192** | 0.1444 | 0.1933 | 0.0250 |
| | ML-Master 2.0 | 0.0151 | 0.1055 | −0.1328 | −0.0136 | 0.0102 | 0.0926 | −0.2477 | −0.0246 |
| | **Ours** | **0.0259** | **0.1721** | **0.1907** | **0.0211** | 0.0188 | **0.1496** | **0.4128** | **0.0491** |
| 200K tokens | AutoResearch | 0.0111 | 0.0677 | −0.7090 | −0.0734 | 0.0115 | 0.0880 | 0.1770 | 0.0207 |
| | R&D-Agent | 0.0169 | 0.1064 | −0.3223 | −0.0342 | **0.0195** | 0.1441 | 0.0326 | 0.0037 |
| | AIRA | 0.0172 | 0.1170 | −0.0154 | −0.0017 | 0.0192 | 0.1444 | 0.1933 | 0.0250 |
| | ML-Master 2.0 | 0.0151 | 0.1055 | −0.1328 | −0.0136 | 0.0102 | 0.0926 | −0.2477 | −0.0246 |
| | **Ours** | **0.0259** | **0.1721** | **0.1907** | **0.0211** | 0.0188 | **0.1496** | **0.4128** | **0.0491** |
| 400K tokens | AutoResearch | 0.0102 | 0.0618 | −0.4345 | −0.0428 | 0.0115 | 0.0880 | 0.1770 | 0.0207 |
| | R&D-Agent | 0.0182 | 0.1142 | −0.1995 | −0.0211 | 0.0212 | 0.1557 | 0.3938 | 0.0489 |
| | AIRA | 0.0172 | 0.1170 | −0.0154 | −0.0017 | 0.0192 | 0.1444 | 0.1933 | 0.0250 |
| | ML-Master 2.0 | 0.0232 | **0.1710** | −0.0180 | −0.0018 | 0.0102 | 0.0926 | −0.2477 | −0.0246 |
| | **Ours** | **0.0244** | 0.1291 | **0.0837** | **0.0090** | **0.0219** | **0.1646** | **0.6836** | **0.0945** |

therefore suggests that the diagnostic-guided search uses the available token budget more effectively than the baseline search strategies.

### G.2.3 Search Efficiency

Figure 3 in the main text reports parent–child OOD IC improvements on NASDAQ-100, while Figure 10 below presents the corresponding analysis on S&P 500. Both figures cover the 100K-, 200K-, and 400K-token budgets. The positive transition rate is the fraction of evaluable pairs whose child improves OOD IC over its parent. The average positive gain is the mean relative IC change, $(\mathrm{IC}_{\mathrm{child}} - \mathrm{IC}_{\mathrm{parent}})/|\mathrm{IC}_{\mathrm{parent}}|$, among these improving pairs. Both are reported as percentages.

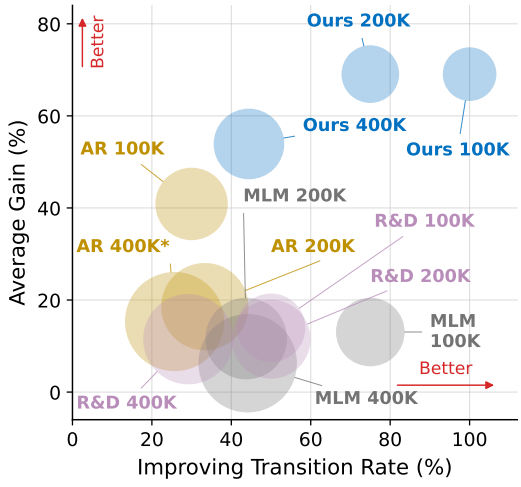


Figure 10: Search efficiency under different token budgets. A state transition is positive when the child improves IC over its parent. X-axis: fraction of positive transition pairs; y-axis: average IC Gain among positive transitions. Bubble size shows the number of state transitions.

### G.3 Additional Results on Crypto Market

To assess DASH's performance across different markets and temporal settings, we additionally evaluate it on the cryptocurrency market.

#### G.3.1 Market, Universe, and Evaluation Protocol

**Market and universe.** We extend the evaluation to a daily cryptocurrency panel (Crypto) spanning 2020-01-01 to 2025-10-20. The universe comprises a static set of 110 crypto assets, including Bitcoin (BTC) and Ethereum (ETH). Observations follow a daily calendar, including weekends. Each asset is represented by open, high, low, close, and volume (OHLCV), with BTC providing the market benchmark. The prediction target is the five-day forward return adjusted for exposure to BTC.

**Train–search and OOD periods.** The train–search period runs from 2020-01-01 to 2024-12-31, and the held-out OOD evaluation window runs from 2025-01-05 to 2025-10-20. Table 22 specifies the training and validation windows. Factor-only methods rank factors by absolute IC on the common validation period, 2024-01-05 to 2024-12-31, and freeze their signs before OOD evaluation. The coevolution baselines use mean validation IC over the three search splits.

Table 22: Temporal splits for the Crypto experiments. The 2024 window is the common validation period for factor selection.

| Split | Training | Validation | OOD test |
|---|---|---|---|
| Search 1 | 2020-01-01–2021-12-31 | 2022-01-05–2022-12-31 | — |
| Search 2 | 2020-01-01–2022-12-31 | 2023-01-05–2023-12-31 | — |
| Search 3 | 2020-01-01–2023-12-31 | 2024-01-05–2024-12-31 | — |
| Final OOD | 2020-01-01–2023-12-31 | 2024-01-05–2024-12-31 | 2025-01-05–2025-10-20 |

The final model evaluation uses a single forward OOD cycle, with the training and validation windows shown in the last row of Table 22. The coevolution methods start from raw asset OHLCV and a ridge predictor with regularization coefficient $\alpha = 1.0$. The crypto experiments use GPT-4.1 and report results at the token checkpoints specified below.

**Random diagnostic splits.** For DASH, validation start dates for random diagnostic splits are sampled from January 1, 2021 to December 31, 2023. Each training period extends one year backward from the sampled date and ends on the preceding day, while the validation period extends one year forward from that date, ending one day before its first anniversary.

#### G.3.2 Factor-Only Results

**Token-matched predictive performance.** Tables 23 and 24 report OOD results at 200K and 400K tokens, respectively. DASH achieves the best individual-factor performance on all four metrics at both budgets, with IC $0.0514$ and ICIR $0.3548$ already at 200K. These results remain unchanged at 400K, indicating that a strong factor is found within the smaller budget. The advantage also extends to factor aggregation: DASH achieves the highest Rank IC across all seven settings at both budgets and leads all four metrics with Top-5 rolling-linear aggregation. Overall, these results demonstrate strong OOD predictive performance for both individual factors and factor combinations under matched token budgets.

Table 23: Factor-only out-of-distribution predictive performance on Crypto at the 200K-token budget under seven factor aggregation strategies. Higher is better. **Bold**: best; underline: second best within each aggregation strategy.

| Aggregation | Method | IC | Rank IC | ICIR | Rank ICIR |
|---|---|---|---|---|---|
| Best factor | **Ours** | **0.0514** | **0.0873** | **0.3548** | **0.4099** |
| | FactorMiner | −0.0118 | −0.0101 | −0.0782 | −0.0704 |
| | AlphaAgent | 0.0024 | −0.0200 | 0.0110 | −0.1046 |
| | R&D-Agent | 0.0063 | −0.0037 | 0.0589 | −0.0337 |

| Aggregation | Method | IC | Rank IC | ICIR | Rank ICIR |
|---|---|---|---|---|---|
| Top-5 equal weight | **Ours** | 0.0416 | **0.0709** | 0.2174 | **0.3080** |
| | FactorMiner | **0.0437** | 0.0426 | **0.2680** | 0.2539 |
| | AlphaAgent | 0.0261 | 0.0069 | 0.1603 | 0.0410 |
| | R&D-Agent | 0.0238 | 0.0292 | 0.1634 | 0.1980 |
| Top-5 rolling linear | **Ours** | **0.0314** | **0.0547** | **0.1503** | **0.2601** |
| | FactorMiner | 0.0212 | −0.0044 | 0.1295 | −0.0252 |
| | AlphaAgent | 0.0019 | −0.0169 | 0.0091 | −0.0922 |
| | R&D-Agent | 0.0001 | 0.0057 | 0.0005 | 0.0339 |
| Top-20 equal weight | **Ours** | **0.0502** | **0.0920** | **0.2433** | **0.3968** |
| | FactorMiner | 0.0360 | 0.0462 | 0.2303 | 0.2984 |
| | AlphaAgent | 0.0261 | 0.0069 | 0.1603 | 0.0410 |
| | R&D-Agent | 0.0291 | 0.0377 | 0.1945 | 0.2724 |
| Top-20 rolling linear | **Ours** | 0.0189 | **0.0343** | 0.0947 | **0.1820** |
| | FactorMiner | **0.0199** | 0.0030 | **0.1386** | 0.0203 |
| | AlphaAgent | 0.0019 | −0.0169 | 0.0091 | −0.0922 |
| | R&D-Agent | −0.0226 | −0.0232 | −0.1613 | −0.1633 |
| Top-40 equal weight | **Ours** | **0.0502** | **0.0920** | 0.2433 | **0.3968** |
| | FactorMiner | 0.0353 | 0.0623 | 0.2051 | 0.3352 |
| | AlphaAgent | 0.0261 | 0.0069 | 0.1603 | 0.0410 |
| | R&D-Agent | 0.0392 | 0.0433 | **0.2706** | 0.2919 |
| Top-40 rolling linear | **Ours** | 0.0189 | **0.0343** | 0.0947 | **0.1820** |
| | FactorMiner | **0.0339** | 0.0209 | **0.2495** | 0.1446 |
| | AlphaAgent | 0.0019 | −0.0169 | 0.0091 | −0.0922 |
| | R&D-Agent | −0.0125 | −0.0074 | −0.0930 | −0.0584 |

Table 24: Factor-only out-of-distribution predictive performance on Crypto at the 400K-token budget under seven factor aggregation strategies. Higher is better. **Bold**: best; underline: second best within each aggregation strategy.

| Aggregation | Method | IC | Rank IC | ICIR | Rank ICIR |
|---|---|---|---|---|---|
| Best factor | **Ours** | **0.0514** | **0.0873** | **0.3548** | **0.4099** |
| | FactorMiner | 0.0303 | 0.0474 | 0.2058 | 0.2967 |
| | AlphaAgent | 0.0024 | −0.0200 | 0.0110 | −0.1046 |
| | R&D-Agent | 0.0063 | −0.0037 | 0.0589 | −0.0337 |
| Top-5 equal weight | **Ours** | **0.0416** | **0.0709** | 0.2174 | 0.3080 |
| | FactorMiner | 0.0400 | 0.0508 | **0.2812** | **0.3671** |
| | AlphaAgent | 0.0269 | 0.0037 | 0.1746 | 0.0231 |
| | R&D-Agent | 0.0238 | 0.0292 | 0.1634 | 0.1980 |
| Top-5 rolling linear | **Ours** | **0.0314** | **0.0547** | **0.1503** | **0.2601** |
| | FactorMiner | −0.0225 | −0.0375 | −0.2013 | −0.3141 |
| | AlphaAgent | −0.0003 | −0.0171 | −0.0017 | −0.0952 |
| | R&D-Agent | 0.0001 | 0.0057 | 0.0005 | 0.0339 |
| Top-20 equal weight | **Ours** | **0.0459** | **0.0915** | 0.2213 | **0.3995** |
| | FactorMiner | 0.0334 | 0.0588 | 0.1984 | 0.3611 |
| | AlphaAgent | 0.0269 | 0.0037 | 0.1746 | 0.0231 |
| | R&D-Agent | 0.0404 | 0.0523 | **0.2573** | 0.3319 |
| Top-20 rolling linear | **Ours** | **0.0191** | **0.0342** | **0.0958** | **0.1811** |
| | FactorMiner | 0.0117 | −0.0081 | 0.0792 | −0.0521 |
| | AlphaAgent | −0.0003 | −0.0171 | −0.0017 | −0.0952 |
| | R&D-Agent | −0.0151 | −0.0138 | −0.0979 | −0.0933 |

| Aggregation | Method | IC | Rank IC | ICIR | Rank ICIR |
|---|---|---|---|---|---|
| Top-40 equal weight | **Ours** | **0.0459** | **0.0915** | **0.2213** | **0.3995** |
| | FactorMiner | 0.0322 | 0.0578 | 0.1873 | 0.3334 |
| | AlphaAgent | 0.0269 | 0.0037 | 0.1746 | 0.0231 |
| | R&D-Agent | 0.0283 | 0.0287 | 0.1998 | 0.2079 |
| Top-40 rolling linear | **Ours** | 0.0191 | **0.0342** | 0.0958 | **0.1811** |
| | FactorMiner | **0.0321** | 0.0095 | **0.2277** | 0.0674 |
| | AlphaAgent | −0.0003 | −0.0171 | −0.0017 | −0.0952 |
| | R&D-Agent | −0.0263 | −0.0232 | −0.1808 | −0.1623 |

**Search efficiency.** Figure 11 reports the fraction of OOD-evaluable valid factors whose IC exceeds 0.01, 0.02, and 0.03 at the 100K-, 200K-, 300K-, and 400K-token budgets. DASH achieves the highest fraction at the stricter 0.02 and 0.03 thresholds across all displayed budgets. At 400K tokens, 71.6% of its factors exceed an OOD IC of 0.02, compared with 28.0% for AlphaAgent, while 47.3% exceed 0.03, compared with 16.2% for FactorMiner, the strongest baseline at this threshold. These results indicate that, under the same token budget, DASH discovers a higher proportion of high-quality factors with strong OOD predictive performance.

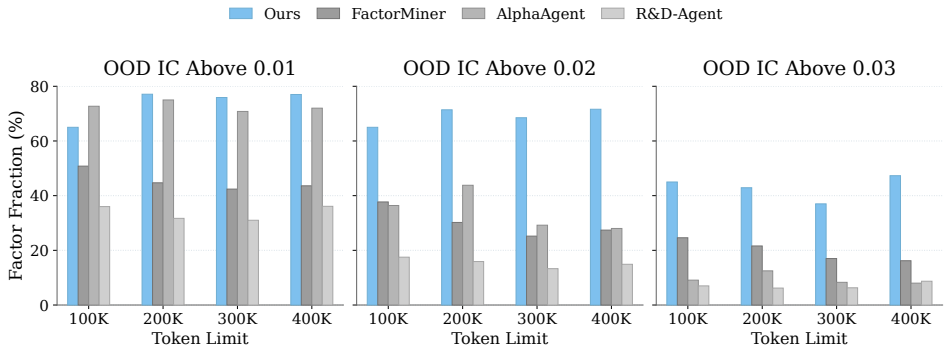


Figure 11: Percentage of OOD-evaluable valid factors with OOD IC above 0.01, 0.02, and 0.03 under different token budgets for Crypto.

### G.3.3 Factor–Model Coevolution Results

**Token-matched evaluation and backtest setup.** Table 25 compares factor–model coevolution at nominal budgets of 200K and 400K LLM tokens. For each method and budget, results are reported as mean ± sample standard deviation over three repeated experiments. All methods use the same daily TopK-Dropout strategy with seven target holdings (`topk`=7), up to three replacements per rebalance (`n_drop`=3), a minimum holding period of one daily bar (`hold_thresh`=1), and a target invested fraction of 0.95 (`risk_degree`=0.95). The initial account value is 100,000,000, and buy and sell transaction-cost rates are each 0.1%. The realized backtest period is January 5–October 19, 2025.

**Portfolio metric.** The reported IR is computed from daily portfolio returns before transaction costs. Let $r_t^{\mathrm{p}}$ and $r_t^{\mathrm{f}}$ denote the daily portfolio and risk-free returns, respectively, and define $e_t = r_t^{\mathrm{p}} - r_t^{\mathrm{f}}$, with $r_t^{\mathrm{f}} = 0$. We use 365-day annualization to reflect daily cryptocurrency trading:

$$\mathrm{IR} = \sqrt{365}\,\frac{\bar{e}}{s(e)}, \qquad \bar{e} = \frac{1}{T}\sum_{t=1}^{T} e_t, \qquad s(e) = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}(e_t - \bar{e})^2}.$$

Here $s(e)$ measures daily return variability within each backtest; the standard deviations reported in the table measure variation across the three repeated runs.

DASH achieves the highest mean IC, Rank IC, Rank ICIR, and portfolio IR at both 200K and 400K tokens. Its largest relative advantage is in Rank IC, exceeding the strongest baseline by 51.9% and

Table 25: Factor–model coevolution results on Crypto at 200K and 400K tokens, reported as mean ± sample standard deviation. **Bold**: best mean; underline: second best within each budget. Higher is better for all metrics.

| | | Predictive Power | | | Portfolio Metrics |
|---|---|---|---|---|---|
| | **Method** | **IC** | **Rank IC** | **Rank ICIR** | **IR** |
| 200K tokens | AutoResearch | $0.0215 \pm 0.0063$ | $\underline{0.0472} \pm 0.0248$ | $\underline{0.2962} \pm 0.1359$ | $0.1984 \pm 0.7649$ |
| | R&D-Agent | $\underline{0.0285} \pm 0.0068$ | $0.0425 \pm 0.0184$ | $0.2569 \pm 0.0932$ | $0.4893 \pm 0.4175$ |
| | AIRA | $0.0215 \pm 0.0097$ | $0.0335 \pm 0.0263$ | $0.1797 \pm 0.1366$ | $0.0585 \pm 0.4701$ |
| | ML-Master 2.0 | $0.0261 \pm 0.0085$ | $0.0308 \pm 0.0181$ | $0.1869 \pm 0.0740$ | $\underline{0.5595} \pm 0.5593$ |
| | **Ours** | $\mathbf{0.0334} \pm 0.0062$ | $\mathbf{0.0717} \pm 0.0039$ | $\mathbf{0.3575} \pm 0.0273$ | $\mathbf{0.6190} \pm 0.3574$ |
| 400K tokens | AutoResearch | $\underline{0.0318} \pm 0.0225$ | $\underline{0.0549} \pm 0.0358$ | $\underline{0.3261} \pm 0.1728$ | $0.1962 \pm 0.7856$ |
| | R&D-Agent | $0.0287 \pm 0.0068$ | $0.0448 \pm 0.0199$ | $0.2683 \pm 0.1028$ | $\underline{0.5847} \pm 0.2633$ |
| | AIRA | $0.0232 \pm 0.0112$ | $0.0348 \pm 0.0274$ | $0.1908 \pm 0.1473$ | $0.3786 \pm 0.3381$ |
| | ML-Master 2.0 | $0.0261 \pm 0.0085$ | $0.0308 \pm 0.0181$ | $0.1869 \pm 0.0740$ | $0.5595 \pm 0.5593$ |
| | **Ours** | $\mathbf{0.0351} \pm 0.0034$ | $\mathbf{0.0713} \pm 0.0033$ | $\mathbf{0.3584} \pm 0.0258$ | $\mathbf{0.5958} \pm 0.3205$ |

29.9%, respectively. Notably, DASH at 200K tokens already outperforms every baseline at 400K tokens on all four metrics, demonstrating efficient search under the evaluated budgets. It also has the lowest reported standard deviations across all three predictive metrics at both budgets. Increasing the budget further improves IC, while the ranking metrics remain nearly unchanged and IR declines slightly. These results extend DASH's predictive and portfolio advantages beyond the primary equity benchmarks.

### G.4 Point-in-Time Universe Sensitivity Analysis

To assess sensitivity to universe construction, we use point-in-time (PIT) versions of the NASDAQ-100 and S&P 500 universes covering the same sample period as the main equity experiments. Tables 26 and 27 report joint factor–model search on PIT S&P 500 and NASDAQ-100 universes at 100K and 400K tokens. Each cell reports the mean ± sample standard deviation across three repeated runs. Portfolio performance is measured by terminal net asset value (NAV). With $\mathrm{NAV}_0 = 1$, we compute $\mathrm{NAV}_t = \prod_{u=1}^{t}(1 + r_u)$, where $r_u$ is the daily portfolio return. NAV uses the backtest window from January 5, 2023 to September 25, 2025.

**Results on S&P 500.** Table 26 shows that DASH achieves the highest mean IC, ICIR, and NAV at both budgets. At 400K tokens, its IC and ICIR reach $0.0136$ and $0.1212$, while mean NAV reaches $1.3513$, compared with $1.2944$ for the strongest baseline. Its Rank ICIR also improves from $0.0307$ at 100K to $0.0668$ at 400K, approaching the best baseline value of $0.0682$. These results show that DASH's predictive advantage is accompanied by stronger wealth accumulation on the PIT universe.

**Results on NASDAQ-100.** In Table 27, DASH leads all four metrics at both budgets. At 400K, it achieves IC $0.0224$ and Rank ICIR $0.1694$, with mean NAV $1.8382$ versus $1.6290$ for the strongest baseline. Strong portfolio performance is already obtained at 100K, where mean NAV reaches $2.0103$. Although the larger budget improves the predictive metrics, it does not increase NAV relative to 100K. DASH nevertheless retains the highest mean NAV at both budgets.

**NAV trajectories.** Figures 12 and 13 show NAV trajectories at the 400K-token budget. DASH finishes with the highest NAV in both markets: $1.4772$ on S&P 500 and $1.8187$ on NASDAQ-100, versus $1.3814$ and $1.4614$ for the strongest plotted baseline, respectively. The NASDAQ-100 curve separates from the baselines by late 2023, while the S&P 500 advantage becomes clearer during 2024. Both curves show temporary declines followed by recovery. These trajectories illustrate how the wealth gap develops over time.

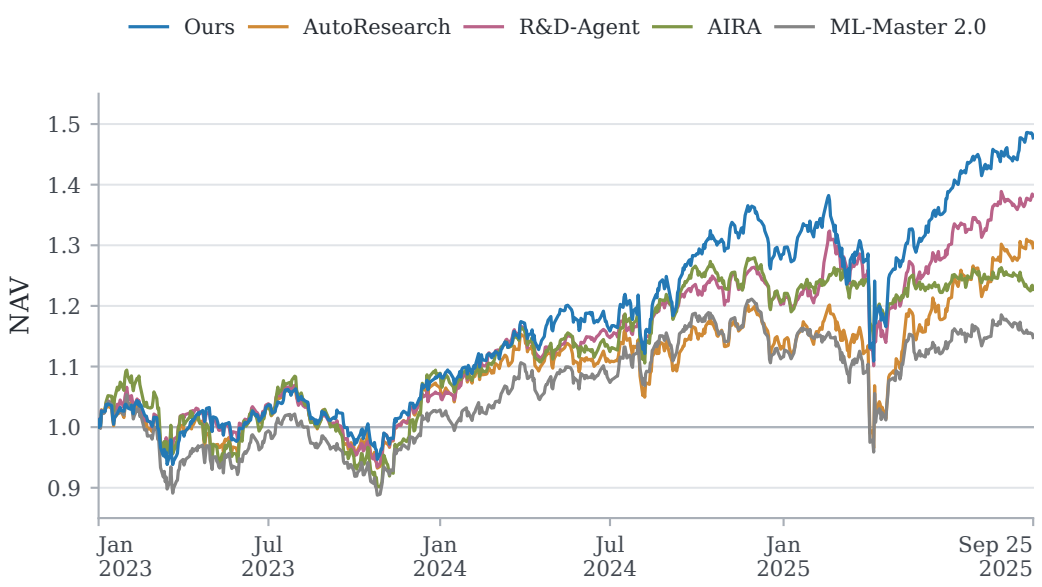


Figure 12: OOD NAV trajectories at 400K tokens on the PIT S&P 500 universe.

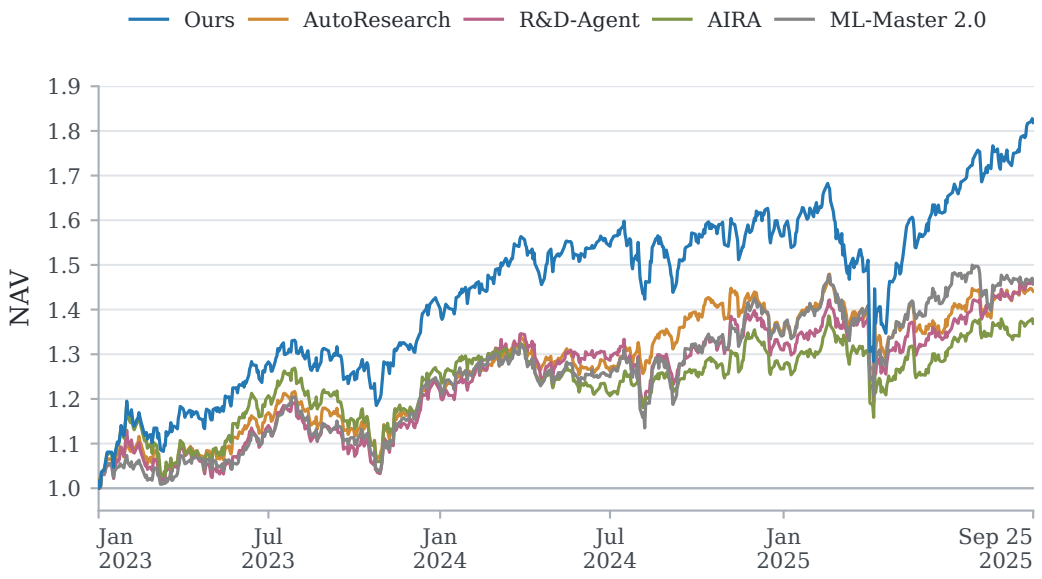


Figure 13: OOD NAV trajectories at 400K tokens on the PIT NASDAQ-100 universe.

Table 26: PIT factor–model search results on S&P 500 at 100K and 400K tokens, reported as mean ± sample standard deviation across three runs. **Bold**: best mean; underline: second best within each budget. Higher is better for all metrics.

| | | Predictive Power | | | Portfolio Metrics |
|---|---|---|---|---|---|
| | **Method** | **IC** | **ICIR** | **Rank ICIR** | **NAV** |
| 100K tokens | AutoResearch | $0.0082 \pm 0.0030$ | $0.0759 \pm 0.0259$ | $\mathbf{0.0469 \pm 0.0168}$ | $1.2866 \pm 0.0894$ |
| | R&D-Agent | $0.0027 \pm 0.0041$ | $0.0262 \pm 0.0396$ | $-0.0058 \pm 0.0449$ | $1.2745 \pm 0.2195$ |
| | AIRA | $0.0079 \pm 0.0013$ | $0.0712 \pm 0.0117$ | $0.0341 \pm 0.0166$ | $1.2149 \pm 0.1061$ |
| | ML-Master 2.0 | $0.0087 \pm 0.0020$ | $0.0748 \pm 0.0190$ | $0.0453 \pm 0.0269$ | $1.2371 \pm 0.0969$ |
| | **Ours** | $\mathbf{0.0097 \pm 0.0025}$ | $\mathbf{0.0915 \pm 0.0253}$ | $0.0307 \pm 0.0150$ | $\mathbf{1.3229 \pm 0.0528}$ |
| 400K tokens | AutoResearch | $0.0100 \pm 0.0020$ | $0.0941 \pm 0.0254$ | $\mathbf{0.0682 \pm 0.0290}$ | $1.2944 \pm 0.1383$ |
| | R&D-Agent | $0.0027 \pm 0.0039$ | $0.0286 \pm 0.0402$ | $0.0172 \pm 0.0667$ | $1.2428 \pm 0.1308$ |
| | AIRA | $0.0073 \pm 0.0007$ | $0.0656 \pm 0.0062$ | $0.0329 \pm 0.0202$ | $1.2550 \pm 0.0392$ |
| | ML-Master 2.0 | $0.0087 \pm 0.0020$ | $0.0748 \pm 0.0190$ | $0.0453 \pm 0.0269$ | $1.2371 \pm 0.0969$ |
| | **Ours** | $\mathbf{0.0136 \pm 0.0010}$ | $\mathbf{0.1212 \pm 0.0072}$ | $0.0668 \pm 0.0045$ | $\mathbf{1.3513 \pm 0.1451}$ |

Table 27: PIT factor–model search results on NASDAQ-100 at 100K and 400K tokens, reported as mean ± sample standard deviation across three runs. **Bold**: best mean; underline: second best within each budget. Higher is better for all metrics.

| | | Predictive Power | | | Portfolio Metrics |
|---|---|---|---|---|---|
| | **Method** | **IC** | **ICIR** | **Rank ICIR** | **NAV** |
| 100K tokens | AutoResearch | $0.0109 \pm 0.0056$ | $0.0736 \pm 0.0343$ | $0.0602 \pm 0.0471$ | $1.4974 \pm 0.1968$ |
| | R&D-Agent | $0.0105 \pm 0.0071$ | $0.0800 \pm 0.0589$ | $0.0782 \pm 0.0364$ | $1.6189 \pm 0.3545$ |
| | AIRA | $0.0145 \pm 0.0057$ | $0.1014 \pm 0.0424$ | $0.0887 \pm 0.0478$ | $1.5567 \pm 0.3550$ |
| | ML-Master 2.0 | $0.0127 \pm 0.0050$ | $0.0851 \pm 0.0313$ | $0.1033 \pm 0.0123$ | $1.4287 \pm 0.0258$ |
| | **Ours** | $\mathbf{0.0160 \pm 0.0099}$ | $\mathbf{0.1140 \pm 0.0697}$ | $\mathbf{0.1114 \pm 0.0675}$ | $\mathbf{2.0103 \pm 0.1252}$ |
| 400K tokens | AutoResearch | $0.0127 \pm 0.0079$ | $0.0843 \pm 0.0517$ | $0.0496 \pm 0.0304$ | $1.3683 \pm 0.1028$ |
| | R&D-Agent | $0.0101 \pm 0.0076$ | $0.0743 \pm 0.0589$ | $0.0661 \pm 0.0249$ | $1.6290 \pm 0.3577$ |
| | AIRA | $0.0121 \pm 0.0046$ | $0.0833 \pm 0.0336$ | $0.0769 \pm 0.0320$ | $1.5893 \pm 0.3869$ |
| | ML-Master 2.0 | $0.0101 \pm 0.0106$ | $0.0695 \pm 0.0723$ | $0.0664 \pm 0.0555$ | $1.4498 \pm 0.0111$ |
| | **Ours** | $\mathbf{0.0224 \pm 0.0013}$ | $\mathbf{0.1605 \pm 0.0187}$ | $\mathbf{0.1694 \pm 0.0147}$ | $\mathbf{1.8382 \pm 0.1129}$ |

## H Action Space Safeguards

Since the LLM generates executable code for both feature construction and model specification, preventing look-ahead leakage and preserving the evaluation protocol is critical. DASH treats this as an action-space constraint: the agent may propose refinements, but each refinement must pass a sequence of protocol, static, and dynamic checks before it can be evaluated and admitted to the archive.

**Fixed evaluation infrastructure.** The data construction pipeline, target definition, temporal splits, OOD protocol, and evaluation harness are fixed and excluded from the LLM's modification scope. Runtime execution restores the controlled dataset protocol before evaluating a candidate, so a proposed configuration cannot change the train–validation split, prediction target, OOD windows, or metric computation. The OOD period is used only for final reporting and post-search analysis; diagnostic profiles, hypothesis verification, archive insertion, parent selection, and solution ranking are all computed from search-period evidence.

**Factor-level constraints.** Generated factor code must satisfy per-asset isolation for time-series operations and per-date grouping for cross-sectional operations. Static screening rejects forbidden

patterns such as forward-looking window arguments, negative shifts, centered rolling windows, backfilling, ungrouped global reductions, direct access to target or weight columns, and file or network I/O. The factor output must align exactly with the input panel index, be numeric, contain no infinite values, and remain within a bounded missing-value rate. Candidate factors that fail any of these checks are discarded before they can enter FactorBase.

**Dynamic point-in-time validation.** Beyond static screening, each generated factor is evaluated through a causal truncation test. The factor function is executed on both the full panel and a truncated panel that removes all rows after a cutoff date. Historical factor values on the shared index must remain unchanged up to a tight numerical tolerance ($\epsilon = 10^{-6}$). This test is repeated over sampled cutoffs and assets, and is designed to catch subtler forms of look-ahead bias that static pattern matching cannot detect.

**Codebase edit scope.** Model-side refinements are restricted to an allowlist of editable model specification files, such as model definitions, losses, and configuration entries. Data loading, preprocessing, split construction, training orchestration, evaluation scripts, and post-search OOD reporting are protected from LLM edits. Proposed edits are parsed as structured file operations and rejected if they target an unapproved path, escape the candidate workspace, or fail syntax and import checks. Candidates are executed in an isolated runtime with read-only data mounts and bounded compute resources. Thus, a model refinement can alter the forecasting configuration, but not the evaluation protocol that judges it.

**Search-time evidence separation.** The diagnostic loop only exposes search-period metric evidence to the LLM. In particular, OOD IC, OOD Rank IC, OOD portfolio statistics, and OOD execution outcomes are not included in observation, hypothesis, metric-planning, verdict, or solution-generation prompts.

# I Additional Related Work

## I.1 Multi-Agent ML Pipelines

AutoML-Agent (Trirat et al., 2024) decomposes the ML pipeline into specialized agents coordinated by a retrieval-augmented planner. MLZero (Fang et al., 2025) combines coding, execution, and error analysis with semantic and episodic memory. DS-Agent (Guo et al., 2024) retrieves expert solutions and revises them through execution feedback. These approaches organize research through agent specialization, memory, and experience reuse. DASH focuses on the diagnostic process connecting an observed weakness in the current configuration to a proposed refinement.

## I.2 Feedback Enrichment in ML Agents

AIRA (Toledo et al., 2025) studies the generalization gap between validation-based selection and held-out performance. AIRA-2 (Hambardzumyan et al., 2026) controls label access through fixed evaluation splits, R&D-Agent (Yang et al., 2025) aggregates feedback across search branches, SEA-TS (Xu et al., 2026) audits generated code for leakage, and SAGA (Du et al., 2025) evolves the optimization objective. AlphaEvolve already supports multiple evaluation metrics, rich feedback, and staged evaluation of newly generated programs (Novikov et al., 2025, Section 2.4).

DASH makes the transition from an observed weakness to an edit explicit. The agent generates hypotheses about the current parent, specifying requested measurements, their data scopes, and expected outcomes before computation. An external skill library computes the numerical evidence without LLM inference. The LLM then compares the evidence with the fixed expectations to assign a supported, refuted, or uncertain verdict. Refuted hypotheses are excluded from refinement, while supported and uncertain hypotheses remain eligible with uncertainty retained. These diagnostic checks concern the current parent before editing; AlphaEvolve's published evaluation cascade evaluates generated programs. The diagnostic protocol could also be incorporated into evolutionary systems such as AlphaEvolve.

### I.3 LLM Agents for Quantitative Finance

AlphaAgent (Tang et al., 2025) uses an Idea–Factor–Eval loop with structural regularization. QuantaAlpha (Han et al., 2026) refines factor-search trajectories through mutation and crossover. R&D-Agent (Li et al., 2025) jointly explores factor construction and predictive models. AlphaBench (Luo et al., 2026) further documents limitations of LLMs as zero-shot judges of factor quality, motivating the use of computed evidence in financial research.

DASH evaluates configurations on shared reference splits and sampled diagnostic windows. Its diagnostic profile preserves temporal and regime-specific evidence, enabling hypotheses about where predictive performance breaks down. Requested measurements assess these hypotheses before refinement, while prediction redundancy informs which evaluated alternatives are retained and revisited during search.

### I.4 Quality-Diversity Search

Quality-diversity (QD) algorithms seek collections of diverse, high-performing solutions (Pugh et al., 2016). Classical MAP-Elites (Mouret & Clune, 2015) retains one elite per behavioral cell, supporting both solution quality and behavioral coverage (Cully et al., 2015). FunSearch (Romera-Paredes et al., 2024) and EoH (Liu et al., 2024) combine LLM-generated proposals with evolutionary search. AlphaEvolve maintains search diversity through a program database inspired by MAP-Elites and island populations (Novikov et al., 2025, Section 2.5). DASH connects this evolutionary search paradigm with diagnostic-guided refinement. Its diagnostic profiles provide behavioral descriptors for archive organization and guide refinement through metric-grounded hypotheses. The archive allows multiple configurations per cell and separately accounts for prediction redundancy alongside predictive strength when retaining and selecting candidates for refinement. This keeps alternative starting points available for subsequent diagnostic-guided search.