FACTORRSI: LLM-GUIDED TEMPORAL FACTOR MINING THROUGH RECURSIVE SELF-IMPROVEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Long behavioral histories can contain predictive signals that differ in temporal scope, sample applicability, and modeling requirements. Manually identifying and combining these signals is costly. LLM-guided factor proposals reduce this effort, but a fixed search policy can continue to allocate resources to unproductive directions as evidence accumulates. We formulate LLM-Guided Factor Evolution as joint search over executable sample conditions, temporal windows, and encoder families, followed by fusion of applicable experts. We then introduce RSI (Recursive Self-Improvement)-Guided Factor Evolution, which adapts the policy directing this search. An outer controller uses search outcomes and VALID-monitor feedback to revise search allocation and proposal preferences, retaining or rolling back each update after validation. Across five datasets and three LLM backbones, RSI-on improves 46 of 48 reported endpoint metrics over RSI-off; the five Q3.5-397B comparisons use 8.2–77.8% fewer input tokens. Factor analyses find useful window–encoder combinations, strong descendants especially from crossover, and gains from combining applicable experts.

1 INTRODUCTION

Time-series prediction (Hall & Rasheed, 2025; Jiang et al., 2026; Zhang et al., 2025a; Liu et al., 2025a)supports tasks ranging from equity-return and retail-demand forecasting to traffic-speed prediction (Zhang et al., 2025b; Priya & Francis, 2025). Although their targets and horizons differ, a prediction from a long behavioral history may depend on several signals rather than one fixed interval. A market forecast can reflect both a recent volatility burst and a persistent liquidity pattern (Rohan et al., 2025; Kiruba et al., 2025; Thota et al., 2025); traffic demand can reflect a recurring cycle and a local disruption. Such signals can differ in when they occur, which samples they describe, and which model captures them effectively. Recent forecasting studies find that predictive variables can matter at different time lags and that models have sample-dependent strengths (Ma et al., 2025; Liu et al., 2025b). The task is to discover these distinct temporal behaviors and combine their evidence in a shared prediction.

Traditional feature engineering leaves much of this work to domain experts. They specify a behavioral hypothesis, select samples and a historical window, choose a model, and evaluate the resulting signal. A persistent trend and a short local disturbance, for example, may warrant different temporal windows and encoders. Testing the possible combinations requires substantial manual effort, while committing to one window or architecture can miss useful signals.

We represent each candidate as an executable temporal factor comprising a sample-applicability condition, a time anchor and range, an encoder family, and an admissible training configuration. Its explicit condition and temporal description make the proposed signal inspectable, even when the fitted encoder is not itself transparent. We use an LLM to propose plausible relationships among behaviors, windows, and encoders, then test those proposals by fitting and evaluating the corresponding experts. Prior work has studied LLM-based feature construction (Hollmann et al., 2023; Han et al., 2024; Abhyankar et al., 2025) and iterative financial factor mining (Tang et al., 2025; Wang et al., 2026; Han et al., 2026). Our formulation, LLM-Guided Factor Evolution (Data/Model-Hybrid)–Fixed Policy, searches over data–model factors while retaining prescribed rules for generating and evaluating them. Section 2 formalizes the factor and prediction setting.

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

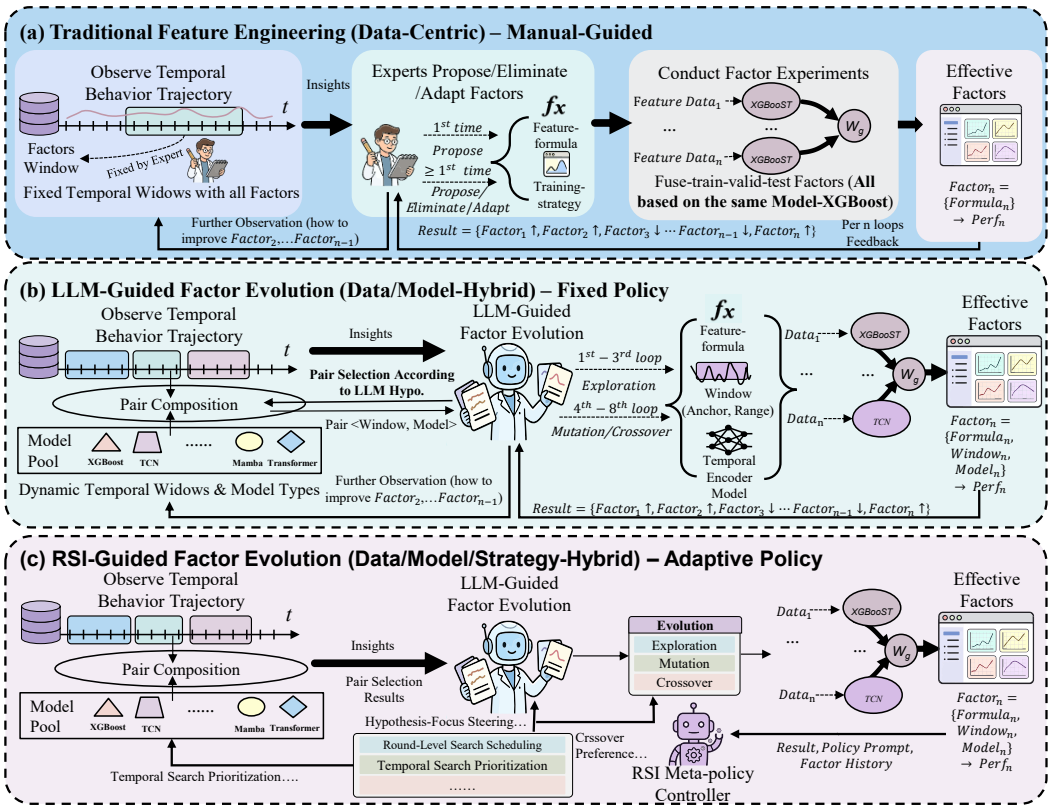

Figure 1: Three levels of temporal-factor discovery. (a) Traditional Feature Engineering (Data-Centric)–Manual-Guided: experts specify and test data-derived signals. (b) LLM-Guided Factor Evolution (Data/Model-Hybrid)–Fixed Policy: an LLM proposes executable data–model factors under prescribed decision rules. (c) RSI-Guided Factor Evolution (Data/Model/Strategy-Hybrid)–Adaptive Policy: feedback updates the policy that governs subsequent discovery. Data selection, encoder choice, training strategy, and hyperparameter selection are all policy decisions; the implementation determines which policy components can be revised.

Discovery alone is insufficient when several factors apply to the same sample. One expert may capture a signal on a narrow subset, while another contributes different information on overlapping samples. Specialized predictors and subgroup discovery offer related perspectives (Ismail et al., 2023; Eyuboglu et al., 2022). Here, factor conditions determine applicability, validation performance determines eligibility and reliability, and fusion combines the predictions of applicable experts with the configured calibration and global predictor. A prediction can therefore use multiple temporal signals without requiring one encoder to represent every behavior.

Automated proposals still leave the search policy fixed. As the factor population grows, a prescribed allocation of exploration, mutation, and crossover may repeatedly generate low-value hypotheses instead of developing useful ones. We introduce RSI-Guided Factor Evolution (Data/Model/Strategy-Hybrid)–Adaptive Policy to improve the decisions governing subsequent search. An outer controller reviews search outcomes and VALID-monitor feedback, proposes bounded changes to operator allocation and preferences over conditions, windows, encoders, and hypotheses, and retains or rolls back those changes after validation. Data selection, model choice, training strategy, and hyperparameter rules are all policy decisions; the implemented controller adapts a permitted subset while keeping the expert-fitting and evaluation rules fixed. Figure 1 summarizes the progression from manual feature engineering to fixed-policy factor evolution and then adaptive-policy search. The same feedback structure could respond to changing behavior, although production shifts are not evaluated here.

Across five datasets and three LLM backbones, RSI-on improves 46 of 48 reported endpoint metrics over RSI-off, with one tie and one regression. In the five Q3.5-397B endpoint comparisons, cumulative input-token use falls by 8.2–77.8%. Factor-level analyses identify useful signals across temporal windows and encoders, while strong later descendants arise especially through crossover. Fusion also improves predictions over most individual factors on their matched samples. Together, these results support the two parts of the approach: discovering complementary data–model factors and adapting the policy that directs their discovery.

2 Problem Formulation and Preliminaries

2.1 Prediction from multiple temporal signals

Let $\mathbf{x}_{i,t}$ denote the feature vector observed for entity i at time t. Using the history $X_{i,\le t} = \{\mathbf{x}_{i,u} : u \le t\}$, we predict a target $y_{i,t+h} \in \mathbb{R}$ realized at the future time $t+h$, where $h > 0$ is the task-specific forecast horizon. Thus, t is the input cutoff and $t+h$ is the target time. For a target defined over a future interval, such as a return, $t+h$ denotes the end of the target interval. Observation indices follow the signal periods of each dataset, so their calendar duration can differ. A task specification $\mathcal{T}$ defines the feature schema, target and realization horizon, chronological partitions, admissible encoder families, and evaluation metrics. Daily equity prediction, weekly signal-era prediction, and traffic forecasting use the same search interface, but each retains its own data and evaluation protocol. We seek a predictor $\hat{y}_{i,t+h} = G(X_{i,\le t}; \mathcal{T})$ that combines **multiple signals from the observed history**. **A temporal factor specifies both the samples to which it applies and a model of a particular historical interval.** Factors may overlap: the same sample can exhibit a short-term disturbance and a persistent trend. The task is to discover useful data–model combinations and integrate their predictions, rather than associate each sample with a single temporal pattern.

2.2 Executable data–model factors

We write a candidate factor as

$$z_j = (h_j, c_j, k_j, L_j, e_j, \theta_j), \tag{1}$$

where h_j is a natural-language hypothesis, c_j is an executable sample-selection condition, k_j and L_j specify temporal placement and width, e_j is an encoder family, and θ_j is its effective training configuration. The hypothesis explains the proposal, while the executable fields determine its behavior. Conditions combine endpoint-feature comparisons using Boolean operators and absolute or within-period percentile thresholds. They select samples while preserving all input feature channels.

For ordinary sequence encoders, the implementation parameter $k_j \ge 1$ specifies the lag of the right endpoint of the input window. We define the right endpoint $b_j(t)$ and starting anchor $a_j(t)$ as

$$b_j(t) = t + 1 - k_j, \qquad a_j(t) = b_j(t) - L_j + 1 = t + 2 - k_j - L_j, \tag{2}$$

so that the input is $W_j(X_{i,\le t}) = (\mathbf{x}_{i,a_j(t)}, \ldots, \mathbf{x}_{i,b_j(t)})$. Setting $k_j = 1$ includes the latest available observation, whereas increasing k_j shifts the window backward while leaving the prediction target unchanged. The current sequence implementation constrains the width to $1 \le L_j \le 60$. The starting anchor depends on both the endpoint lag and the window width. Sample membership is evaluated at the input cutoff t, independently of the historical window:

$$m_j(i,t) = \mathbb{I}[c_j(\mathbf{x}_{i,t})]\, \mathbb{I}[W_j(X_{i,\le t}) \text{ is available}]. \tag{3}$$

Historical rows inside a valid window need not themselves satisfy c_j.

Temporal specifications are normalized to the actual input requirements of each encoder. Tree encoders operate on the flat endpoint row and therefore have effective $(k_j, L_j) = (1, 1)$; proposed temporal parameters do not alter this input.

The training subset and fitted expert are

$$\mathcal{D}_j^{\text{train}} = \{(W_j(X_{i,\le t}), y_{i,t+h}) : (i,t) \in \mathcal{D}^{\text{train}},\ m_j(i,t) = 1\}, \quad f_j = \text{Train}_{e_j,\theta_j}(\mathcal{D}_j^{\text{train}}). \tag{4}$$

For non-sequence families, W_j denotes the corresponding normalized input. The pipeline constrains the training configuration: model size, dropout, and batch size for ordinary sequence models scale with the number of matched training samples, while tree and external models follow family-specific rules. These rules are part of the training policy.

2.3 LLM-GUIDED FACTOR EVOLUTION: DATA/MODEL-HYBRID WITH FIXED POLICY

A policy governs factor discovery and training through four components: data selection, model selection, training, and search allocation,

$$\pi = (\pi^{\text{data}}, \pi^{\text{model}}, \pi^{\text{train}}, \pi^{\text{search}}). \tag{5}$$

These components govern sample conditions and temporal windows, encoder choice, training strategies and hyperparameters, and the allocation of exploration and variation, respectively. They specify the rules and preferences behind factor proposals; different proposals can arise from the same policy as search feedback changes.

Let $\mathcal{P}_r$ denote the retained checkpoint pool after round r, $\mathcal{A}_r$ the archive used for parent selection, and $\mathcal{M}_r$ the recorded plans, lineage, metrics, and execution outcomes. Validation-based inclusion rules determine the subset $\mathcal{S}_r \subseteq \mathcal{P}_r$ eligible for the ensemble. Under search policy π_r, factor proposals follow

$$(o_r, P_r) \sim \pi_r(\cdot \mid \mathcal{A}_r, \mathcal{M}_r), \qquad z'_r \sim q_\psi(\cdot \mid \mathcal{T}, o_r, P_r, \mathcal{M}_r, \pi_r), \tag{6}$$

Here, P_r denotes the selected parents; a proposal may contain multiple factors. In fixed-policy evolution, $\pi_r = \pi_0$ even as $\mathcal{A}_r$, $\mathcal{M}_r$, and $\mathcal{P}_r$ change. The task therefore searches jointly over sample conditions, temporal windows, and encoder families without adapting the rules that guide the search. Section 3 describes how RSI relaxes this restriction.

The objective is to improve ensemble utility within a specified resource budget,

$$\max \ \mathbb{E}[Q(G_{\mathcal{S}_R})] \quad \text{subject to} \quad \sum_{r=1}^{R} \mathbf{c}_r \preceq \mathcal{B}, \qquad z'_r \in \mathcal{Z}_\mathcal{T}, \tag{7}$$

where $\mathbf{c}_r$ records search and training costs, and $\mathcal{Z}_\mathcal{T}$ is the set of executable factors allowed by the task specification. The experiments report performance at dataset-specific endpoints, along with LLM calls and input-token costs. Some analyses compare matched rounds within a dataset; endpoint comparisons use different completed search horizons. The resource-constrained objective does not imply that all reported runs have equal computational or token budgets.

The task also fixes the data used for fitting and evaluation. Experts are fitted on TRAIN, while disjoint chronological VALID subsets support checkpoint selection, calibration, ensemble selection, and monitoring. These roles remain unchanged when RSI adapts the search policy in Section 3, using VALID-monitor results as evidence. Section 4.1 and Table 2 specify the dataset partitions and boundary purging;

3 LLM-GUIDED FACTOR EVOLUTION WITH RSI POLICY ADAPTATION

The fixed-policy task in Section 2.3 specifies the factor search space and its governing policy. We first describe the shared LLM-guided evolution loop and prediction rule, then the RSI controller that adapts the policy. Figure 2 summarizes the two levels.

3.1 FACTOR DISCOVERY AND EVOLUTIONARY VARIATION

Given the task definition and prior outcomes, the LLM proposes an executable factor in the form of Equation 1: **a condition specifying where the factor applies, a historical window specifying where to look, and an encoder specifying how to model that history.** Each feasible factor trains an expert on the TRAIN samples selected by its condition. Exploration proposes new factor hypotheses; mutation revises the hypothesis, condition, temporal window, or encoder of one parent; and crossover recombines two parent specifications into new factors. RSI-on and RSI-off use these same operators, while RSI-on adapts their relative use and the proposal preferences supplied to the LLM. Appendix L details the prompts and parent selection; Appendix Figure 8 illustrates the three operators.

3.2 COMBINING COMPLEMENTARY FACTORS

A factor contributes only to predictions for which its condition and temporal input are available. Let f_0 be the unconditional global predictor and $\mathcal{S}_r$ the factors eligible for the ensemble after validation

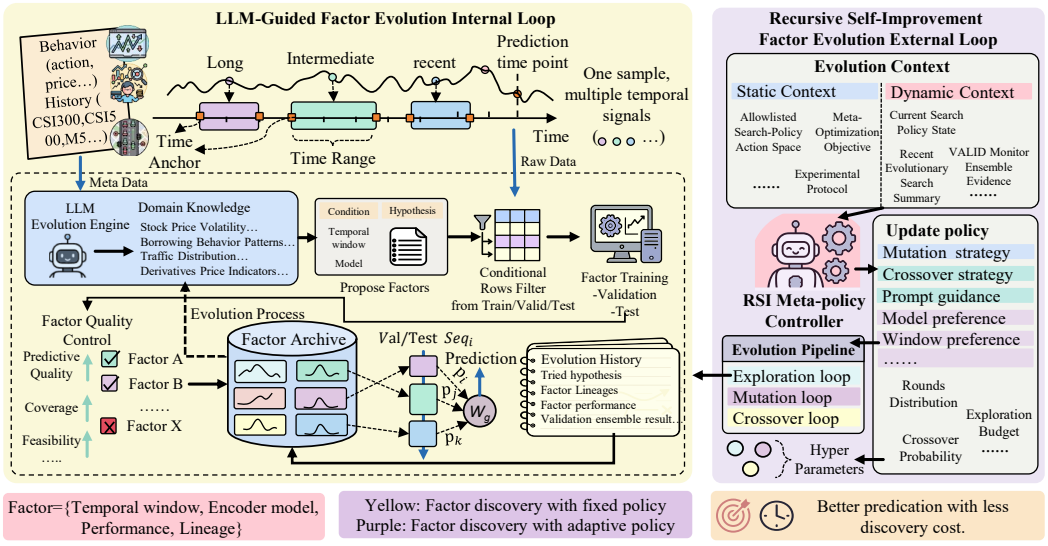

Figure 2: Overview of fixed-policy factor evolution and RSI-guided policy adaptation.

selection. For an eligible expert j, the primary validation metric q_j determines its reliability weight, $w_j = \left[\max(q_j, 10^{-4})\right]^2$. Among experts applicable to a sample, a larger q_j yields a larger fusion weight. Experts that do not apply to that sample are excluded from its local prediction. Eligibility requires sufficient validation support and positive IC, ICIR, and RankIC; the exact selection rule is given in Appendix L.

Where configured, the calibration map g_j is fitted on applicable VALID-calibration samples; otherwise it is the identity. With $\tilde{f}_j = g_j \circ f_j$, the applicable experts form a local prediction,

$$s_{i,t+h} = \begin{cases} \dfrac{\sum_{j \in \mathcal{S}_r} m_j(i,t) w_j \tilde{f}_j(W_j(X_{i,\le t}))}{\sum_{j \in \mathcal{S}_r} m_j(i,t) w_j}, & \sum_{j \in \mathcal{S}_r} m_j(i,t) w_j > 0, \\ f_0(X_{i,\le t}), & \text{otherwise.} \end{cases} \tag{8}$$

If none applies, the global predictor supplies the prediction. A dataset-fixed blend combines the local and global predictions,

$$\hat{y}_{i,t+h} = \alpha f_0(X_{i,\le t}) + (1-\alpha) s_{i,t+h}. \tag{9}$$

Calibration follows the dataset profile, while $\alpha = 0.05$ for all five datasets and both RSI settings. Appendix K reports a sweep of factor-pool/global weights from 0.90/0.10 to 0.99/0.01. The 0.95/0.05 pair gives the strongest overall balance across the reported metrics, supporting this fixed setting. The resulting prediction can draw on several behavioral signals when their factors apply to the same sample, while the global predictor remains available where local evidence is absent.

3.3 From fixed-policy evolution to adaptive search

RSI extends the shared factor-evolution and fusion pipeline by adapting admissible components of the policy in Equation 5. RSI-off retains the prescribed policy; RSI-on uses completed search summaries and VALID-monitor performance to revise it as the factor population develops. Both settings retain the same factor space, training protocol, and prediction rule. At each reflection, the controller proposes a bounded change, tests the provisional policy through subsequent search, and uses validation feedback to retain or roll it back. Appendix Figure 9 shows one recorded reflection.

3.4 State-conditioned policy reflection

The controller summarizes what the current policy has produced rather than inspecting individual TEST examples. Its state contains the policy version π_v, recent search summaries $\mathcal{H}_v$, the latest

successful VALID-monitor triplet $\mathbf{M}_v = (M_{v,\mathrm{IC}}, M_{v,\mathrm{ICIR}}, M_{v,\mathrm{RankIC}})$, and the preceding policy decision d_{v-1}. A meta-level LLM uses this evidence to propose a change,

$$\delta_v \sim q_{\mathrm{meta}}(\cdot \mid \pi_v, \mathcal{H}_v, \mathbf{M}_v, d_{v-1}, \mathcal{T}), \qquad \pi'_v = \mathrm{ValidateAndMerge}(\pi_v, \delta_v), \tag{10}$$

Here, $\mathcal{T}$ denotes fixed task constraints, and $\mathrm{ValidateAndMerge}$ admits only valid policy changes. A valid patch may change two kinds of decisions. Search-allocation parameters govern exploration, mutation, crossover, and parent selection; proposal preferences guide the LLM toward selected coverage levels, temporal regions, encoders, and hypotheses. The latter are preferences rather than hard filters on feasible factors. The rationale states why the controller favors a direction, whereas only the validated policy patch changes subsequent search. The action space, prompt, and mutation taxonomy are specified in Appendix L.

RSI adjusts the number of initial exploration rounds per session, the probability of crossover versus mutation, parent-selection settings, mutation-operator weights, and soft proposal preferences for encoders, sample coverage, temporal windows, and hypotheses (Appendix Table 6).

3.5 Validation challenge and rollback

RSI assesses each provisional policy at the next eligible VALID-monitor ensemble evaluation. For $m \in \{\mathrm{IC}, \mathrm{ICIR}, \mathrm{RankIC}\}$, let $b_m > 0$ be a fixed dataset-specific reference and define $J(\mathbf{M}) = \frac{1}{3}\sum_m M_m/b_m$. The update is retained only if $J(\mathbf{M}^{\mathrm{after}}) > J(\mathbf{M}^{\mathrm{before}})$ and $\min_m \frac{M_m^{\mathrm{after}} - M_m^{\mathrm{before}}}{b_m} \geq -0.05$; otherwise, RSI restores the previous policy while retaining generated factors. Appendix L.1 gives the full rule and its interpretation.

4 Experiments

We evaluate fixed-policy LLM-guided factor evolution and adaptive-policy RSI-guided factor evolution through predictive performance, temporal-factor structure, endpoint resource use, and factor-lineage analysis. The study covers CSI300, CSI500, Numerai, M5 Walmart, and PEMS-BAY. Section 4.1 defines these dataset-specific interpretations.

4.1 Datasets and evaluation protocol

We evaluate equity returns on CSI300 and CSI500 with Qlib Alpha158 panels (Yang et al., 2020), weekly equity signals on Numerai Signals v2.1 (Numerai, 2025), 28-day retail demand on M5 (Makridakis et al., 2022), and one-hour-ahead traffic speed on PEMS-BAY (Li et al., 2018). For every task, the TRAIN, VALID, and TEST partitions are disjoint and strictly ordered in time: TRAIN precedes VALID, which precedes TEST. Dataset protocols are detailed in Appendix F. We use task-specific variants of IC, ICIR, and RankIC when the target or signal period differs from daily equity prediction. Definitions are given in Appendix F.

4.2 Standalone models and evolutionary ensembles

Table 1 compares FactorRSI with fixed-policy evolution, other LLM-guided methods, XGBoost, and LightGBM across five datasets and three LLM backbones. Additional baselines appear in Appendix Tables 4 and 3. RSI-on improves 46 of 48 unrounded endpoint metrics over RSI-off (one tie and one regression), while FactorRSI compares favorably with competing methods. These results are consistent with benefits from both complementary factor discovery and policy adaptation, but do not establish equal-budget or seed-averaged superiority.

4.3 Search cost

Across five datasets, RSI-on achieves comparable or better endpoint TEST performance with 8.2–77.8% fewer input tokens, improving IC, ICIR, and RankIC per million tokens in every dataset (Figure 3(a); Appendix Table 5). This pattern is consistent with policy adaptation directing search toward productive proposals and rolling back ineffective updates. The controller ablation supports the performance benefits of adaptation and rollback (Figure 6), but different endpoint rounds prevent attributing token savings to individual components.

Table 1: TEST performance with Qwen3.5-397B-A17B-FP8 (Q), Gemini 3.7 Flash (G), and GPT-5.6 Luna (L). Cells report IC/ICIR/RankIC, except M5, which reports WRMSSE/IC/ICIR/RankIC. Results are reported at performance extrema; lower WRMSSE and higher values of the other metrics are better.

Model / method	CSI300	CSI500	Numerai	M5	PEMS-BAY
Our methods: fixed and adaptive policy					
FactorRSI-off (Q)	.044/.281/.035	.030/.267/.030	.053/2.961/.045	1.661/.374/.526/.590	.728/6.222/.740
FactorRSI-on (Q)	.045/.302/.036	.031/.269/.032	.053/2.988/.046	1.658/.376/.536/.600	.731/6.347/.746
FactorRSI-off (G)	.040/.264/.031	.028/.237/.031	.051/2.876/.043	1.718/.377/.520/.564	.706/5.901/.713
FactorRSI-on (G)	.041/.271/.032	.030/.254/.031	.053/3.064/.046	1.694/.371/.526/.586	.717/6.031/.727
FactorRSI-off (L)	.044/.295/.034	.029/.269/.031	.053/3.214/.047	1.654/.378/.539/.601	.732/6.419/.742
FactorRSI-on (L)	.046/.317/.034	.031/.274/.032	.054/3.426/.047	1.633/.381/.539/.608	.743/6.571/.749
LLM-guided factor evolution					
FactorMiner (Q)	.026/.227/.031	.022/.237/.028	.048/2.535/.044	1.756/.359/.516/.563	.701/5.440/.729
QuantaAlpha (Q)	.042/.252/.031	.029/.261/.032	.051/2.494/.044	1.724/.350/.514/.579	.699/5.882/.714
AlphaAgent (Q)	.023/.206/.027	.020/.243/.030	.049/2.516/.041	1.987/.350/.510/.557	.700/5.861/.707
FactorMiner (G)	.024/.217/.030	.021/.224/.028	.047/2.705/.044	1.768/.356/.513/.561	.691/5.412/.716
AlphaAgent (G)	.021/.202/.025	.018/.231/.028	.049/2.361/.040	2.004/.335/.521/.567	.694/5.794/.701
QuantaAlpha (G)	.040/.264/.030	.028/.245/.029	.051/2.482/.043	1.784/.342/.511/.561	.681/5.210/.703
FactorMiner (L)	.027/.239/.030	.023/.257/.027	.049/2.616/.048	1.744/.362/.519/.576	.713/5.562/.732
QuantaAlpha (L)	.042/.271/.032	.030/.275/.032	.052/2.515/.043	1.701/.362/.518/.584	.713/5.834/.724
AlphaAgent (L)	.023/.215/.028	.020/.246/.030	.050/2.537/.043	1.984/.353/.521/.562	.717/5.913/.717
Complex models					
XGBoost	.031/.262/.027	.018/.167/.020	.051/2.089/.042	1.962/.338/.520/.583	.699/5.415/.721
LightGBM	.011/.067/.003	.010/.089/.008	.047/2.780/.037	1.929/.341/.523/.523	.701/5.463/.722

4.4 What do targeted factor adjustments change?

Targeted edits have task- and metric-dependent effects (Figure 3(b); Appendix Figure 17): range resizing improves all three metrics on CSI500; encoder switching improves all three on CSI300 but lowers IC and ICIR on CSI500. Condition tightening improves IC and RankIC but reduces ICIR on PEMS-BAY. These differences motivate the RSI policy to use validation feedback when allocating operators and steering proposals, rather than assigning fixed preferences to temporal, encoder, and sample-selection changes. Fixed-policy evolution can propose these data–model edits; RSI further uses validation feedback to adapt operator allocation and proposal preferences. Heterogeneous outcomes show that access to the same operators does not ensure equally effective search policies.

4.5 Temporal structure of the discovered factors

The main comparison shows broad predictive gains from the evolved ensemble. We next examine whether the factors discovered by LLM-guided search draw on a range of temporal structures. Across observed CSI300 input-range lengths, median IC, ICIR, and RankIC remain positive, but performance is not monotonic in length: median IC is 0.0261 at $L = 10$, 0.0173 at $L = 20$, and 0.0152 at $L = 40$ (Figure 3). Factor quality also varies within the same length. The most frequent temporal conditions contain three to seven encoders with different compositions, showing that evolution is not confined to a single window–encoder configuration (Figure 4). Crucially, encoder performance changes across temporal conditions; even XGBoost does not dominate every region (Figure 4). These results support joint window–encoder search for factors suited to distinct temporal signals.

4.6 RSI, factor concentration, and fusion

The endpoint cost results motivate a closer look at the factor populations produced by each policy.

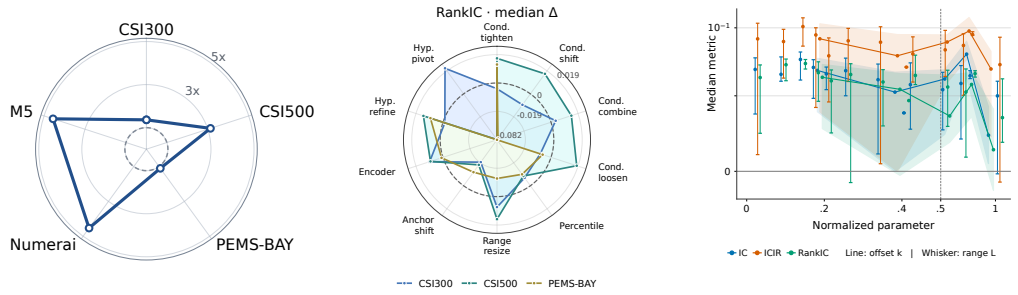

Figure 3: Search efficiency, targeted adjustments, and temporal-factor quality. (a) Geometric mean of the RSI-on/RSI-off TEST IC, ICIR, and RankIC performance-per-million-token ratios across datasets. (b) Median child-minus-parent RankIC by mutation type and dataset. (c) CSI300 factor IC, ICIR, and RankIC (vertical axis) across normalized window-end lag $k/5$ and input-range length $L/60$ (horizontal axis); bands and whiskers show interquartile ranges. Detailed endpoint results and IC/ICIR adjustment views appear in Appendix Table 5 and Figure 17.

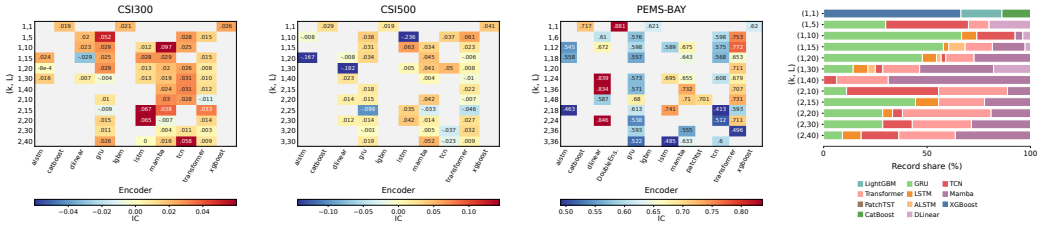

Figure 4: Temporal conditions and encoder choices. Panels (a)–(c) show median factor IC by encoder (horizontal axis) and the 12 most frequent (k, L) conditions (vertical axis) on CSI300, CSI500, and PEMS-BAY; gray cells have no eligible factor. Panel (d) shows encoder shares (horizontal axis) within the 12 most frequent CSI300 temporal conditions (vertical axis).



Figure 5: Panels 1–3 show RSI-on at round 80 and RSI-off at rounds 80 and 250: horizontal and vertical axes give single-factor and ensemble TEST IC on matched participating samples, color denotes TEST coverage, and the dashed line denotes equal IC. Panel 4 shows participating-slice count over TEST samples for RSI-on and RSI-off across five datasets.

At round 80, RSI-on has higher median single-factor TEST IC (0.0211 versus 0.0170), a narrower interquartile range (0.0137 versus 0.0176), and more positive-IC factors (95.2% versus 88.5%) than RSI-off, despite retaining more factors (270 versus 217). Even after 250 rounds and 677 factors, RSI-off reaches median IC 0.0194 and interquartile range 0.0161. Longer fixed-policy search thus expands the pool without matching the earlier concentration of predictive factors under RSI.

This pattern is consistent with RSI directing mutation and crossover toward promising lineages rather than expanding the archive indiscriminately. The scatter shows concentration in predictive quality, not proximity in factor-specification space or cessation of exploration; Appendix N examines the retained PEMS-BAY descendants.

Fusion provides a second, complementary gain. On matched samples, the ensemble exceeds the individual factor's IC for 87.4%, 81.1%, and 83.0% of factors in the three snapshots. Thus, the benefit of multi-signal prediction is present under both policies and is not limited to the strongest isolated factor. Calibration, reliability weighting, and global-model supplementation allow applicable signals to contribute to a common prediction.

4.6.1 WHERE DO STRONGER FACTORS EMERGE?

Exploration produces nearly overlapping RSI-on and RSI-off RankIC distributions on both datasets (Appendix Figure 19). On CSI300, narrower-coverage mutation and crossover factors under RSI-on lead clearly in the high-performance tail: at TEST RankIC ≥ 0.08, mutation yields four versus zero factors and crossover eight versus one. These strong descendants help explain the final ensemble gain. On CSI500, RSI-on leads only in crossover; its mutation factors trail RSI-off, yet the RSI-on ensemble still performs better. This pattern suggests that crossover produces factors that participate more effectively in the ensemble and add greater marginal predictive value, offsetting weaker mutation outcomes. Endpoint comparisons do not isolate the causal effect of an operator.

5 ABLATION STUDIES

We ablate RSI state construction, LLM reflection, policy execution, and validation-based rollback through five restricted variants. Full RSI achieves the highest reported TEST IC, ICIR, and RankIC in all 15 dataset–metric comparisons; reflection without structured updates, no adaptation, updates without rollback, partial state masking, and random updates each fall short. This pattern supports the joint role of state-informed policy updates and validation-based acceptance, but separate runs cannot isolate their causal effects. Appendix J defines the variants and reports their results (Figure 6).

6 RELATED WORK

LLMs have been used directly for time-series forecasting. LLMTime serializes numerical histories as text for zero-shot prediction (Gruver et al., 2023), while Time-LLM reprograms temporal patches and uses a frozen LLM for forecasting (Jin et al., 2024). These methods use the LLM within the predictive path. Here, the LLM instead proposes explicit sample conditions, temporal windows, and encoder families; the resulting experts are trained and combined to make predictions.

LLM-guided financial factor mining provides a closer precedent for iterative discovery. AlphaAgent regularizes generated factor expressions for originality and complexity (Tang et al., 2025); FactorMiner uses experience from earlier trials to guide subsequent proposals (Wang et al., 2026); and QuantaAlpha refines mining trajectories through mutation and crossover (Han et al., 2026). Our method focuses on a further level of adaptation: RSI revises the policy that allocates search and steers factor proposals, then accepts or rolls back the revision using validation feedback. This separates evolution of the factors from adaptation of the policy that discovers them.

7 CONCLUSION

We replace manual feature design with LLM-guided search over sample conditions, windows, and encoders, then fuse applicable experts. RSI adds bounded, validation-guided adaptation of search allocation and proposal preferences. Across five datasets and three LLM backbones, RSI improves 46 of 48 reported metrics over RSI-off; the five Q3.5-397B endpoints use fewer input tokens. Heterogeneous experts and strong crossover descendants help explain these gains. Equal-budget, repeated-run evaluation and production-shift adaptation remain open.

AI USE STATEMENT

Generative AI tools assisted with manuscript organization, drafting, code inspection, and interpretation of author-supplied experimental outputs. Reported measurements originate from the experimental pipeline rather than language-model estimates. The authors are responsible for verifying the results, citations, implementation descriptions, and final claims before submission.

ETHICS STATEMENT

The experiments concern financial prediction, retail demand, and traffic forecasting. Explicit factor specifications support inspection but do not establish causal validity, fairness, or deployment readiness. Use in consequential decisions would require separate assessment of data permissions, privacy, robustness, and effects on the affected population.

REPRODUCIBILITY STATEMENT

Sections 2 and 3 specify the executable factor representation, chronological validation roles, fusion rules, and bounded RSI updates. The appendix documents the condition grammar, RSI prompts and operator taxonomy, and diagnostic measurement provenance. An anonymized project page is linked at the start of the appendix. Reproduction requires the dataset-specific run configuration, policy history, checkpoint identities, and evaluation receipts; historical runs should not be relabeled using a later implementation profile.

REFERENCES

Nikhil Abhyankar, Parshin Shojaee, and Chandan K. Reddy. Llm-fe: Automated feature engineering for tabular data with llms as evolutionary optimizers. *arXiv preprint arXiv:2503.14434*, 2025. URL https://arxiv.org/abs/2503.14434.

Sabri Eyuboglu, Maya Varma, Khaled Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Ré. Domino: Discovering systematic errors with cross-modal embeddings. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2203.14960.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pp. 2242–2251, 2019. URL https://proceedings.mlr.press/v97/ghorbani19c.html.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/3eb7ca52e8207697361b2c0fb3926511-Abstract-Conference.html.

Timothy Hall and Khaled Rasheed. A survey of machine learning methods for time series prediction. *Applied Sciences*, 15(11):5957, 2025.

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

Sungwon Han, Jinsung Yoon, Sercan O. Arik, and Tomas Pfister. Large language models can automatically engineer features for few-shot tabular learning. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2404.09491.

Noah Hollmann, Samuel Müller, and Frank Hutter. Large language models for automated data science: Introducing caafe for context-aware automated feature engineering. *arXiv preprint arXiv:2305.03403*, 2023. URL https://arxiv.org/abs/2305.03403.

Aya Abdelsalam Ismail, Sercan O. Arik, Jinsung Yoon, Ankur Taly, Soheil Feizi, and Tomas Pfister. Interpretable mixture of experts. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=DdZoPUPm0a.

Yushan Jiang, Wenchao Yu, Geon Lee, Dongjin Song, Kijung Shin, Wei Cheng, Yanchi Liu, and Haifeng Chen. Timexl: Explainable multi-modal time series prediction with llm-in-the-loop. *Advances in Neural Information Processing Systems*, 38:135431–135467, 2026.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-llm: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/680b2a8135b9c71278a09cafb605869e-Abstract-Conference.html.

Angelin S Kiruba, D Jyothi Sri, V Meenakshi, S Jeyaraman, R Gokul, and S Kumaran. Risk prediction in financial markets using hybrid ai and time series forecasting models. In *2025 International Conference on Sensors and Related Networks (SENNET) Special Focus on Digital Healthcare (64220)*, pp. 1–7. IEEE, 2025.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1707.01926.

Hang Liu, Gaowei Yan, Lifeng Cao, Suxia Ma, Guanjia Zhao, and Zhongyuan Liu. Physics-informed koopman networks for industrial process time-series prediction. *Industrial & Engineering Chemistry Research*, 64(37):18328–18346, 2025a.

Shengcai Liu, Caishun Chen, Xinghua Qu, Ke Tang, and Yew-Soon Ong. Large language models as evolutionary optimizers. In *IEEE Congress on Evolutionary Computation*, pp. 1–8, 2024. doi: 10.1109/CEC60901.2024.10611913.

Zhining Liu, Ze Yang, Xiao Lin, Ruizhong Qiu, Tianxin Wei, Yada Zhu, Hendrik Hamann, Jingrui He, and Hanghang Tong. Breaking silos: Adaptive model fusion unlocks better time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 40022–40042, 2025b. URL https://proceedings.mlr.press/v267/liu25cm.html.

Xiaowen Ma, Zhen-Liang Ni, Shuai Xiao, and Xinghao Chen. Timepro: Efficient multivariate long-term time series forecasting with variable- and time-aware hyper-state. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 42096–42111, 2025. URL https://proceedings.mlr.press/v267/ma25p.html.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The m5 competition: Background, organization, and implementation. *International Journal of Forecasting*, 38(4): 1325–1336, 2022. doi: 10.1016/j.ijforecast.2021.07.007.

Numerai. Numerai signals dataset v2.1 (alpha). https://signals.numer.ai/data/v2.1, 2025.

C Swetha Priya and F Sagayaraj Francis. Optimizing traffic speed prediction using a multi-objective genetic algorithm-enhanced rnn for intelligent transportation systems. *IEEE Access*, 13:35688–35706, 2025.

Arafat Rohan, Md Deluar Hossen, Md Nuruzzaman Pranto, Balayet Hossain, Areyfin Mohammed Yoshi, and Rakibul Islam. Artificial intelligence in financial market prediction: advancements in machine learning for stock price forecasting. *Frontiers in Artificial Intelligence*, 8:1696423, 2025.

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.

Raghupathi Thota, Sri Meghana Potluri, Bharathi Kaki, Haayder M Abbas, et al. Financial bidirectional encoder representations from transformers with temporal fusion transformer for predicting financial market trends. In *2025 International Conference on Intelligent Computing and Knowledge Extraction (ICICKE)*, pp. 1–5. IEEE, 2025.

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.

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. URL https://arxiv.org/abs/2009.11189.

Xuan Zhang, Meiqi Song, Xiao Xiao, and Xiaojing Liu. Exp-transformer time series prediction model for accident scenarios in high-reliability energy systems: Nuclear power plants case. *Energy*, 322:135481, 2025a.

Yifan Zhang, Qishen Zhou, Jianping Wang, Anastasios Kouvelas, and Michail A Makridis. Casaformer: Congestion-aware sparse attention transformer for traffic speed prediction. *Communications in Transportation Research*, 5:100174, 2025b.

Project page. An anonymized project page is available at `https://anonymous.4open.science/w/factorrsi-project-page-8B92/`.

A Detailed boundary with adjacent paradigms

Our progression separates the object being constructed from the policy constructing it. Traditional Feature Engineering (Data-Centric)–Manual-Guided places hypothesis design and experimental choices primarily with domain experts. LLM-Guided Factor Evolution (Data/Model-Hybrid)–Fixed Policy automates executable data–model proposals. RSI-Guided Factor Evolution (Data/Model/Strategy-Hybrid)–Adaptive Policy additionally optimizes permitted decision rules using feedback. Data selection, encoder selection, training strategies, and hyperparameter selection are components of policy at all three levels; the levels differ in how that policy is specified and improved.

Generic LLM evolutionary optimizers mutate or recombine candidate solutions under performance feedback (Liu et al., 2024). Our candidates are explicit temporal data–model factors, and the RSI outer loop changes the policy governing their generation and evaluation within a fixed information contract. A context-sensitive proposal from an unchanged policy is distinct from a policy update.

LLM-based automated feature engineering proposes transformations for a downstream model (Hollmann et al., 2023; Han et al., 2024; Abhyankar et al., 2025). Temporal-factor evolution jointly specifies applicable samples, a temporal view, and an encoder. Several resulting experts can contribute to one prediction through validation-based reliability weighting.

Boosting fits learners to a prescribed residual or gradient (Friedman, 2001). Here, the LLM generates structured factor hypotheses under the current policy, and execution supplies empirical feedback. RSI adapts selected policy controls rather than replacing expert training with language-model judgments.

Mixtures of experts provide model specialization, while our factor conditions give explicit sample applicability. The eligible experts can vary across examples; their calibrated predictions are combined with a global predictor using the fixed dataset-specific fusion rule. Policy adaptation changes the search for these experts and does not change that fusion rule in the present implementation.

Feature attribution explains a predictor, and Data Shapley values training examples relative to a learning algorithm (Ghorbani & Zou, 2019). RSI instead optimizes the policy for discovering factors. The optional add-one marginal gate is a separately configured population-selection mechanism; neither its use nor Shapley estimation follows from enabling RSI.

B Discussion and Limitations

The three-level progression distinguishes manual signal discovery, automated data–model factor evolution under a fixed policy, and adaptive optimization of that policy. Its usefulness depends on whether a history contains complementary signals that can be represented by the factor specification and learned from sufficiently large local populations. Training and combining several experts introduces costs that their predictive value must justify. The current comparisons assess predictive and recorded search efficiency, not a direct measurement of saved human labor.

Policy includes data-selection rules, encoder-selection rules, training strategies, hyperparameter-selection rules, and search allocation. A fixed policy can already be context-dependent, as when model size is determined by the number of matched samples. RSI adds the ability to revise selected policy components from feedback. The present controller exposes bounded search controls and proposal preferences; several training and execution rules remain fixed. Broader training-policy adaptation is a possible extension, not an evaluated capability implied by the Data/Model/Strategy-Hybrid designation.

The factor representation makes sample applicability, temporal scope, and encoder choice inspectable, but an interpretable factor is not necessarily a causal mechanism. Boolean conditions may miss useful latent structure, and local validation quality can be unstable on small supports. Reliability weighting and optional calibration do not guarantee that every additional factor is complementary. These limitations motivate separate examination of coverage, redundancy, and predictive performance.

RSI acceptance compares validation outcomes before and after a policy challenge while the factor population also changes. It is a practical control rule rather than a causal estimate of an isolated update. The reported endpoint gains extend across several datasets and backbones, but do not establish seed-averaged improvements or an equal-budget advantage. Factor-level distributions describe retained candidates, whose supports and score availability differ; they do not decompose the ensemble gain into causal contributions from individual operators. Moreover, aggregate causal TEST feedback reaches the inner proposal loop, so the complete search should not be described as evaluation on an untouched final holdout. Production-shift adaptation likewise remains a design motivation rather than a demonstrated deployment result.

LLM-generated hypotheses can inherit domain stereotypes, and financial predictions can have disparate effects. Explicit factor specifications aid inspection but do not establish fairness or causal validity. Any deployment requires separate checks appropriate to its population, information constraints, and decision context.

C TEMPORAL-FACTOR GRAMMAR

The executable condition parser operates on available endpoint features. Its Boolean expressions select samples; the temporal window and encoder are separate fields of the factor specification. A compact grammar for a nontrivial condition is:

```
condition   := predicate | NOT condition
             | (condition AND condition)
             | (condition OR condition)
predicate   := feature comparator threshold
threshold   := numeric_constant | Qpercentile
comparator  := < | <= | > | >= | == | !=
```

Legal feature names come from the dataset schema. A percentile threshold is computed within the relevant signal period; an absolute threshold uses the feature's supplied units. An empty condition denotes an unconditional population. The parser does not create arbitrary temporal aggregation programs: any rolling statistic used in a condition must already be an available feature. Sequence construction then applies the normalized temporal plan in Equation 2, while membership is evaluated at the prediction endpoint. All input channels remain available to the trained expert.

Stored factor identities combine the normalized executable condition, effective temporal specification, and encoder. Checkpoint retention compares validation quality within this identity; it does not establish equivalence between every pair of conditions that happens to select the same samples.

D EVOLUTION-STATE INFORMATION

The inner proposal loop receives the task specification, admissible factor fields, selected parents, archive summaries, and permitted evaluation feedback. Under RSI-off these inputs evolve while the discovery and training decision rules remain fixed. RSI-on additionally supplies the currently accepted or provisional policy preferences.

The outer controller receives the current policy, the VALID-monitor metric triplet, recent operation and execution counts, and the preceding policy decision. It updates permitted policy components without changing the task's information-access contract. The meta-controller's VALID-only evidence must be distinguished from aggregate causal TEST feedback released to the inner loop, as described in Section **??**.

E OPTIONAL MARGINAL EVALUATION

The implementation provides a separately configured VALID-only add-one gate for candidate factors. When enabled, it compares candidate and incumbent ensemble predictions using the same chronological validation roles, with checks for gain, blockwise stability, and duplication. The default setting disables this gate. Neither RSI-on nor RSI-off therefore implies coalition-refitted Factor Shapley

Table 2: Temporal prediction tasks and chronological data splits: daily equity-return prediction (CSI300 and CSI500), weekly equity-signal prediction (Numerai), 28-day retail-demand forecasting (M5), and one-hour-ahead traffic-speed forecasting (PEMS-BAY).

Dataset	Frequency	Target	TRAIN	VALID	TEST
CSI300	Daily	Return	08.01.01–14.12.31	15.01.01–16.12.31	17.01.01–20.08.01
CSI500	Daily	Return	08.01.01–14.12.31	15.01.01–16.12.31	17.01.01–20.08.01
Numerai	Weekly	20D2L	05.01.07–19.11.29	19.12.06–22.01.14	22.01.21–26.04.24
M5	Daily	28-day sales	14.11.10–15.12.06	15.12.07–16.03.27	16.04.25–16.05.22
PEMS-BAY	5 min	1-h speed	17.01.01 08:00 17.05.08 00:00	17.05.08 00:05 17.05.26 02:15	17.05.26 02:20 17.07.01 06:55

Dates use YY.MM.DD (2000s). PEMS-BAY adds UTC HH:MM, with each interval's start above its end.

estimation, and no Shapley result should be inferred from the RSI comparison. The implemented fusion and the policy-level acceptance test are given in Equations 9 and 12, respectively.

F Datasets and task-specific metrics

CSI300 and CSI500 use daily Alpha158 equity panels (Yang et al., 2020) to predict a future close-to-close return from information available after the current close. Numerai uses the labelled Signals v2.1 panel (Numerai, 2025) and its official factor- and feature-neutralized 20D2L target at weekly signal eras. M5 Walmart (Makridakis et al., 2022) uses daily item–store sales for direct 28-day demand forecasting. PEMS-BAY (Li et al., 2018) uses five-minute sensor readings to predict traffic speed one hour ahead; targets are standardized using TRAIN statistics. Table 2 gives the configured chronological bounds.

All TRAIN, VALID, and TEST partitions are ordered and disjoint. CSI300 and CSI500 use the configured Qlib date ranges. Numerai and PEMS-BAY divide ordered unique timestamps in a 70/10/20 ratio. For these four tasks, VALID is further partitioned into early-stopping, calibration, selection, and monitoring periods with nominal proportions of 30%, 20%, 25%, and 25%. Boundaries are purged according to target realization: two trading days for CSI300 and CSI500, five weekly signal periods for Numerai, and twelve five-minute intervals for PEMS-BAY. M5 uses four consecutive, complete 28-day VALID blocks assigned to the same four roles. The interval from 2016-03-28 to 2016-04-24 supplies input history for the subsequent TEST forecast. Ensemble membership, reliability weights, and calibration are fixed from VALID before this forecast, and TEST results are not returned to the LLM. The other four datasets use causal walk-forward evaluation and aggregate feedback. Both RSI arms share these task-specific rules.

For CSI300, CSI500, Numerai, and PEMS-BAY, let c_t and r_t be the Pearson and Spearman correlations between predictions and targets across entities at signal period t. We report $\mathrm{IC} = \mathrm{mean}_t(c_t)$, $\mathrm{ICIR} = \mathrm{mean}_t(c_t)/\,\mathrm{std}_t(c_t)$, and $\mathrm{RankIC} = \mathrm{mean}_t(r_t)$, with a sample standard deviation and no annualization. A signal period is a trading day for CSI300 and CSI500, a weekly era for Numerai, and a five-minute timestamp for PEMS-BAY; the corresponding entities are stocks or traffic sensors. These task-specific targets and grouping rules change the interpretation and scale of the scores, but higher values remain better.

M5 uses forecast-error scores in the logged fields named `IC`, `ICIR`, and `RankIC`. They equal $1/(1+\mathrm{WRMSSE})$, $1/(1+\mathrm{RMSSE}_{\mathrm{L12}})$, and $1/(1+\mathrm{MASE}_{\mathrm{L12}})$, respectively. WRMSSE aggregates scaled squared errors over the 12-level sales hierarchy; the latter errors assess item–store forecasts. All three transformed scores increase as forecast errors decrease. The separately reported WRMSSE follows its conventional lower-is-better direction.

G Individual temporal-model baselines

H Additional complex-model baselines

Table 3: TEST performance of individual temporal models at performance extrema. Cells report IC/ICIR/RankIC, except M5, which reports WRMSSE/IC/ICIR/RankIC. Lower WRMSSE and higher values of the other metrics are better.

Model / method	CSI300	CSI500	Numerai	M5	PEMS-BAY
GRU	.022/.115/.016	.023/.225/.028	.047/2.552/.040	2.119/.321/.514/.514	.708/5.991/.743
ALSTM	.021/.134/.016	.016/.173/.017	.047/3.167/.040	2.204/.312/.511/.574	.684/5.643/.702
LSTM	.020/.131/.015	.009/.096/.016	.038/2.303/.033	1.548/.393/.527/.589	.652/5.523/.683
Transformer	.022/.146/.020	.013/.150/.017	.043/2.941/.037	1.558/.391/.524/.585	.701/5.798/.738
TCN	.023/.158/.021	.020/.215/.021	.036/2.542/.032	2.146/.318/.517/.581	.708/5.804/.733
Mamba	.020/.109/.018	.013/.145/.018	.035/1.844/.034	1.766/.362/.524/.589	.699/5.488/.727
PatchTST	.020/.168/.019	.014/.160/.013	.039/2.156/.036	2.325/.301/.510/.575	.702/5.879/.738
iTransformer	.009/.106/.017	.002/.029/.006	.039/2.450/.036	2.170/.315/.514/.578	.698/5.804/.735
DLinear	.010/.009/.015	.007/.076/.010	.047/2.468/.041	2.598/.278/.501/.564	.707/5.832/.738

Table 4: Additional complex-model TEST results at performance extrema. Cells report IC/ICIR/RankIC, except M5, which reports WRMSSE/IC/ICIR/RankIC. Lower WRMSSE and higher values of the other metrics are better.

Model / method	CSI300	CSI500	Numerai	M5	PEMS-BAY
CatBoost	.026/.191/.019	.004/.038/.003	.050/2.048/.041	1.893/.346/.526/.590	.701/5.390/.722
DoubleEnsemble	.017/.125/.016	.014/.129/.006	.047/2.627/.039	1.885/.347/.515/.574	.693/5.299/.716
MASTER	.023/.017/.006	.013/.135/.004	.048/2.694/.039	1.814/.351/.511/.581	.693/5.164/.720
StockMixer	.011/.014/.016	.005/.145/.008	.048/2.717/.039	2.036/.341/.509/.572	.702/5.311/.720
Numerai official LightGBM	.011/.096/.016	.013/.127/.016	.032/2.147/.031	2.156/.303/.511/.561	.693/5.921/.732

I ENDPOINT SEARCH RESOURCE ACCOUNTING

Table 5 reports the endpoint rounds, LLM calls, cumulative input tokens, included slices, latency, elapsed time, and metric-specific TEST values per million input tokens. For metric m and cumulative input-token count T, the latter is $10^6 M_m^{\mathrm{TEST}}/T$. Figure 3(a) shows the geometric mean of the three RSI-on/RSI-off ratios within each dataset. The endpoints differ in round, so these values describe reported performance relative to accumulated context use rather than matched-round gains.

Table 5: Endpoint resource use and TEST performance per million input tokens for FactorRSI with Q3.5-397B. Calls count LLM calls, Slices count terminal included slices, LLM latency is summed in minutes, and elapsed time runs from the first LLM call to TEST.

Dataset	RSI	Round	Calls	Input tokens	Slices	LLM latency (min)	Elapsed (h)	IC /M	ICIR /M	RankIC /M
CSI300	on	210	223	1,941,534	528	30.7	20.3	0.023281	0.155547	0.018285
CSI300	off	290	299	2,537,525	747	40.7	27.8	0.017222	0.110738	0.013714
CSI500	on	80	83	724,733	271	11.6	11.3	0.042636	0.371171	0.044154
CSI500	off	250	251	2,200,366	677	36.6	30.7	0.013680	0.121343	0.013725
PEMS-BAY	on	110	116	473,941	312	15.3	43.7	1.541542	13.391962	1.574036
PEMS-BAY	off	110	128	516,424	389	15.9	43.2	1.409113	12.048239	1.433512
Numerai	on	50	54	280,610	224	7.8	26.9	0.188162	10.648231	0.162859
Numerai	off	250	250	1,264,408	891	33.4	45.8	0.041759	2.341807	0.035827
M5	on	50	52	179,856	79	7.1	7.9	2.091351	2.978994	3.336703
M5	off	240	258	809,608	248	32.9	30.0	0.462567	0.649464	0.728958

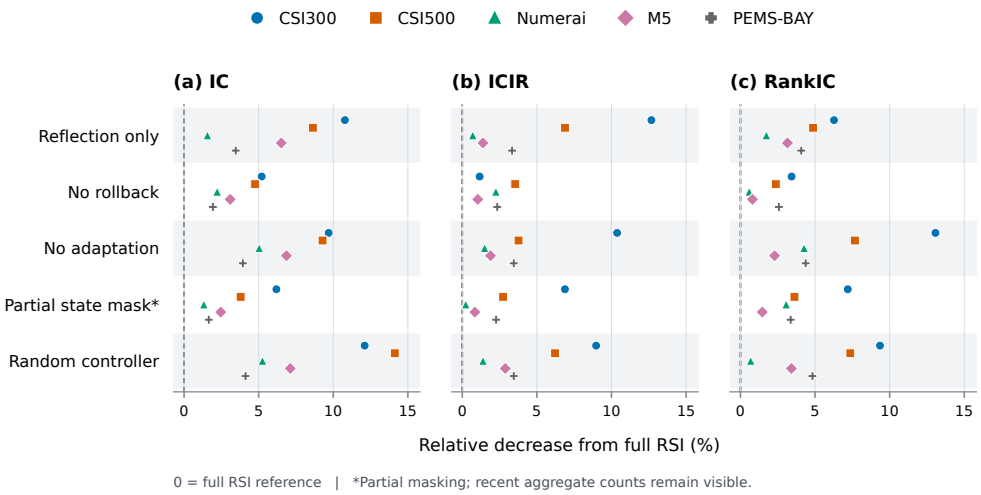

Figure 6: RSI controller ablation on five datasets. Each point shows the relative decrease in a reported best TEST metric from the full RSI reference, $100(M_{\text{full}} - M_{\text{variant}})/M_{\text{full}}$. The dashed line denotes the reference; points farther right indicate larger decreases. Metrics and datasets remain separate. No rollback denotes Adapt-without-rollback; partial state masking denotes the implemented state-blind branch. These are descriptive run summaries, without uncertainty estimates or a verified common evaluation horizon.

J RSI CONTROLLER ABLATION

This ablation examines the four controller components introduced in Sections 3.4 and 3.5: state construction, LLM reflection, policy execution, and validation with rollback. The variants alter the available state, the choice of update, its execution, or the rollback decision. We ask whether adaptive-policy gains require this complete loop, beyond textual reflection under a fixed structured policy.

We compare five variants with the full RSI-on reference. Reflection-only passes the LLM's rationale to subsequent generation prompts but leaves the structured policy fixed. No-adaptation retains the scheduled meta-level calls and the initial RSI policy, but neither applies policy updates nor passes the rationale to generation; it is therefore distinct from RSI-off. Adapt-without-rollback applies valid policy updates regardless of the subsequent acceptance decision. Random controller retains the meta-level calls, policy constraints, and VALID-based acceptance procedure, but replaces the LLM's update with a bounded random policy determined by the seed, dataset, and round. The branch named state-blind masks the previous policy decision and reported recent-round count, while retaining the current policy, VALID metrics, and recent operation, training, skip, and error counts. We label this branch partial state masking because it does not remove all search-state information.

Figure 6 reports the available best TEST summaries for CSI300, CSI500, Numerai, M5, and PEMS-BAY. We normalize each variant by its dataset-specific full reference separately for IC, ICIR, and RankIC, without constructing an aggregate score. The full controller has the highest reported value in all 15 dataset–metric combinations. No-adaptation falls below it by 3.94–9.69% in IC and 2.29–13.07% in RankIC. Reflection-only exceeds No-adaptation in 11 of the 15 comparisons, but remains below the full controller throughout. Thus, textual feedback is useful in several observed comparisons, yet does not recover the performance of the complete adaptive controller.

Applying structured updates without rollback exceeds reflection-only in 13 of the 15 comparisons; Numerai IC and ICIR are the exceptions. Relative to the full controller, removing rollback lowers IC by 1.93–5.20%, ICIR by 1.04–3.55%, and RankIC by 0.59–3.43%. Replacing LLM-selected updates with random valid updates yields an IC decrease of 4.12–14.12%. Together, these observations are consistent with useful contributions from both the direction of policy adaptation and its subsequent validation. Partial state masking also remains below the full reference, although its Numerai ICIR is

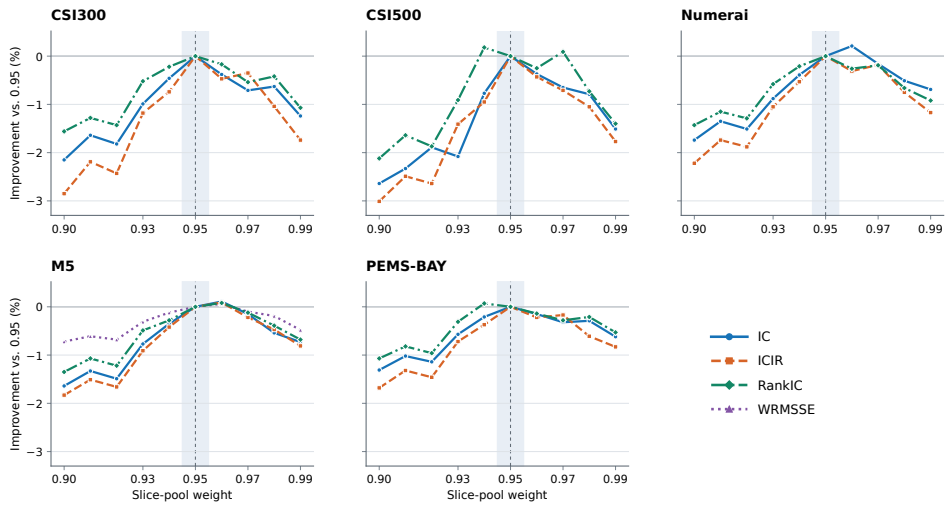

Figure 7: Sensitivity to the factor-pool/global-predictor weights. Each panel corresponds to a dataset. The horizontal axis gives the factor-pool weight; the global weight is its complement. The vertical axis reports the percentage change relative to the 0.95/0.05 setting. Positive values indicate improvement; the direction of WRMSSE is reversed because lower values are better. The dashed vertical line marks the adopted weight.

close (2.981 versus 2.988). Because this variant retains aggregate search statistics, its result cannot identify the effect of removing search state altogether.

These comparisons describe reported performance rather than isolated causal effects. The summaries combine separate run batches, and some documented trajectories terminated before their planned budgets. A common completed horizon and a shared checkpoint for the three reported best metrics have not been established for this collection; we therefore do not interpret the values as a single VALID-selected model per arm. Independent-seed variation is also unavailable. The controller's own acceptance and rollback decisions use VALID rather than TEST, but this fact alone does not establish the absence of TEST feedback elsewhere in the search pipeline. Matched-horizon evaluations, an explicit checkpoint-selection rule, and repeated runs are needed to quantify the reliability of the observed differences.

K FUSION-WEIGHT SENSITIVITY

We vary the factor-pool weight from 0.90 to 0.99, assigning the complementary weight to the global predictor. Figure 7 summarizes the changes in IC, ICIR, RankIC, and, for M5, WRMSSE relative to the 0.95/0.05 setting. The adopted blend provides the strongest overall balance across the five datasets, although a neighboring weight can improve an individual metric.

L RSI PROMPT PROTOCOL AND OPERATOR TAXONOMY

This appendix specifies the messages and action space underlying the controller in Section 3.4. The deployed prompts are written in Chinese and instantiated with dataset-specific terms. The English templates below preserve their operative fields and constraints; bracketed items are runtime substitutions, not observed model responses.

Figure 8 provides an illustrative CSI300-style operator comparison under fixed and adaptive policies. Its factor specifications and policy values are teaching examples; Figure 9 instead excerpts a saved controller call.

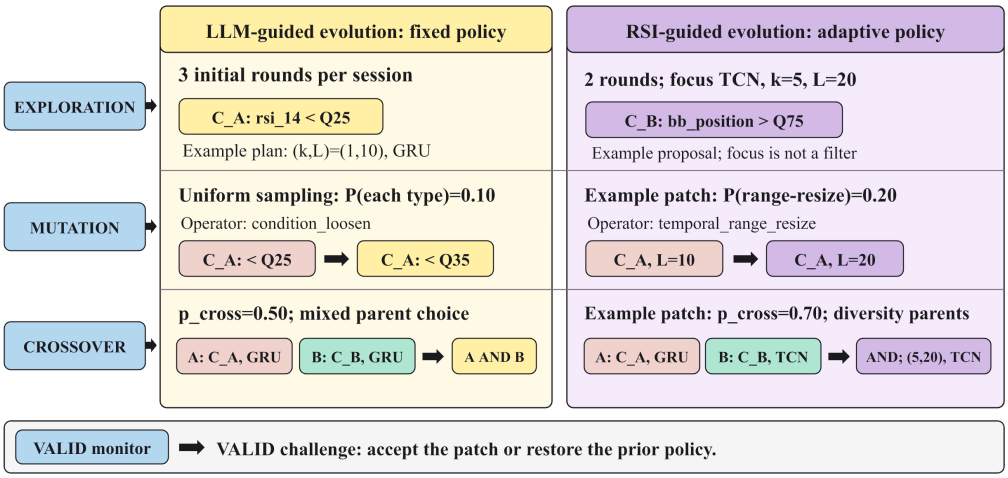

Figure 8: Illustrative operator-level comparison of fixed-policy and RSI-guided factor evolution. In this CSI300-style example, C_A and C_B are percentile-based sample conditions, and (k, L) specifies the window-end lag and input length. The fixed policy retains three exploration rounds, uniform mutation-type weights, and mixed crossover-parent selection. An example RSI patch favors a TCN and $(k, L) = (5, 20)$ exploration region, raises the weight of temporal-range resizing and the crossover probability, and selects diverse parents. The resulting proposals and children show how exploration, mutation, and crossover can change under the same operator set and feasibility checks. All factor specifications and adaptive numerical settings in the diagram are illustrative, not observed run records. A VALID-monitor challenge accepts or restores the patch for the next search interval.

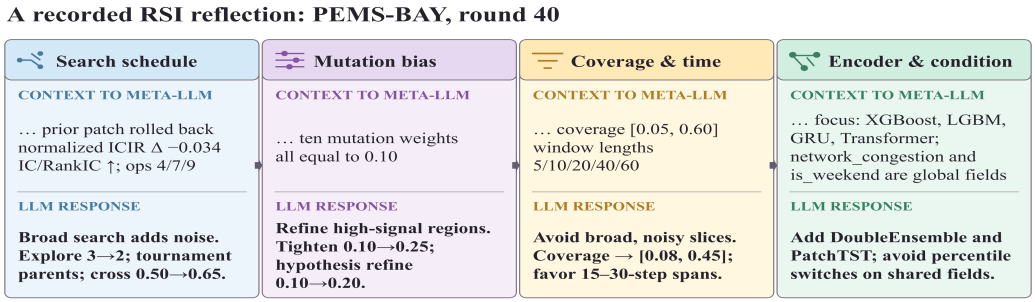

Figure 9: An RSI controller response from PEMS-BAY round 40. After the preceding policy update failed the VALID-monitor acceptance rule, the controller proposed changes to search allocation, mutation weights, coverage and temporal focus, and encoder and condition guidance.

Factor-generation dialogue. The inner-loop system message gives the prediction target, feature descriptions and TRAIN statistics, temporal semantics, condition grammar, eligible encoders, coverage limits, validation objective, and JSON output contract. For a sequence encoder, the condition is evaluated at the signal endpoint t, while `abs_steps_pred` determines the window endpoint and `time_range_steps` determines its length. Tree encoders use the endpoint feature row and have effective temporal fields $(1, 1)$. The model must justify the temporal choice from the hypothesis and condition rather than enumerate lags mechanically. The required plan fields are shown below; `condition_filter` can be empty for an unconditional fallback.

```
System: Propose an executable condition, temporal view,
and encoder for [dataset/task]. Use only [listed features]
```

```
and [allowed encoders]. Return JSON only.
User (exploration): [TRAIN/VALID sizes and feature classes];
[top archived factors and validation summaries];
propose diverse plans, including an unconditional baseline.
Assistant: {"action":"train", "plans":[{
  "name":"...", "hypothesis":"...",
  "condition_filter":{"condition":"..."},
  "abs_steps_pred":k, "time_range_steps":L,
  "encoder":"...", "rationale":"..."}]}
```

After execution, the next user message reports the round outcome, including trained, skipped, and failed counts, selected checkpoint summaries, coverage, and VALID metrics. For mutation, this message additionally supplies the chosen operator, one parent hypothesis, condition, encoder, temporal fields, coverage, and validation scores; it asks for one child plan. For crossover, it supplies the same fields for two parents, suggests conjunction, disjunction, or recombination of their conditions and temporal descriptions, and asks for one or two children. Both return the same `train/plans` JSON schema. The fixed-policy and RSI variants share this dialogue, but RSI appends the current policy's encoder, coverage, temporal, hypothesis, and avoidance preferences. The addendum explicitly states that these are guidance rather than hard filters and cannot alter the data or evaluation protocol.

```
User (mutation): Round [r]; mutation type [operator];
instruction [operator-specific change]; parent
{hypothesis, condition, encoder, lag, range,
 coverage, matched rows, VALID IC/ICIR/RankIC};
archive [summary]; target [primary VALID metric].
Return exactly one child in the train/plans schema.

User (crossover): Round [r]; parent A {...};
parent B {...}; combine their conditions and, for
sequence models, redesign temporal fields if warranted;
archive [summary]; target [primary VALID metric].
Return one or two children in the train/plans schema.
```

Controller dialogue and state packet. At an eligible reflection event, the meta-level system message instructs the controller to improve the next search interval using completed TRAIN/VALID outcomes. It permits only a JSON strategy patch and prohibits changes to code, splits, targets, purge and time rules, VALID roles, fusion, resources, or information permissions. The meta-level user message has the following structure:

```
Review the search policy and propose a bounded patch.
Improve the joint VALID-monitor metrics while retaining
condition, encoder, and temporal diversity.
Evidence JSON: {
  "round": [index],
  "current_strategy": [policy version and values],
  "latest_VALID_monitor_ensemble":
    {"combined_ic":..., "combined_icir":...,
     "combined_rank_ic":...},
  "benchmark_normalized_joint_score":...,
  "recent_round_summary":
    {"rounds":..., "operation_counts":...,
     "trained":..., "skipped":..., "errors":...},
  "previous_patch_decision":...,
  "allowed_mutations":..., "allowed_encoders":...,
  "immutable_constraints":...
}
Assistant: {"action":"update_strategy",
  "rationale":"...", "strategy_patch":{...}}
```

The state packet contains only the latest successful VALID-monitor ensemble triplet, aggregate recent-round counts, the current policy, the preceding acceptance decision, and dataset-specific protocol constraints. Round-level checkpoint records may be persisted for auditing, but they are not copied wholesale into this prompt. No TEST metric, label, prediction, per-sample residual, or coverage distribution enters the RSI controller. This boundary differs from the inner generator, which can receive the released aggregate TEST triplet under the evaluation protocol of Section **??**.

Executable actions and proposal preferences. Table 6 gives the controller's complete allowlisted action schema. A patch may omit fields; omitted fields retain their current values. Invalid fields or values reject the entire patch. Mutation weights are normalized after validation. Focus fields steer subsequent prompts, but do not reject a feasible plan merely because it falls outside the expressed preference.

Table 6: Allowlisted RSI policy fields. Ranges are inclusive. Focus lists contain at most eight entries unless otherwise stated.

Field	Meaning and constraint
`exploration_rounds_per_session`	Initial exploration rounds, integer 1–3
`crossover_probability`	Crossover versus mutation probability, $[0.20, 0.80]$
`parent_selection_mode`	`mixed`, `tournament`, or `diversity`
`diversity_parent_probability`	Diversity choice in mixed mode, $[0, 1]$
`mutation_weights`	Nonnegative weights for the ten operators; positive total
`encoder_focus`	Nonempty list of allowed encoders, at most six
`coverage_target_min/max`	Prompt targets: minimum $[0.02, 0.60]$, maximum $[0.10, 0.80]$, with minimum $<$ maximum
`abs_steps_pred_focus`	Preferred window-end lags, integers 1–60
`time_range_steps_focus`	Preferred input lengths, integers 1–60
`hypothesis_focus`, `avoid_patterns`	Free-text search guidance, at most eight entries each

Mutation classes. The operator names are part of the generator instruction and the controller action schema. `condition_tighten` adds a conjunct or tightens a threshold; `condition_loosen` relaxes or removes one; `condition_shift` changes the condition feature; `condition_combine` joins subconditions with AND or OR; and `percentile_switch` exchanges an absolute threshold with a within-period quantile. `temporal_anchor_shift` changes the window-end lag while preserving the condition, and `temporal_range_resize` changes the input length while preserving the condition and lag. `encoder_switch` changes the encoder under the same condition. `hypothesis_refine` sharpens an existing explanation and its condition, whereas `hypothesis_pivot` proposes a new hypothesis, condition, and encoder. For crossover, the two-parent instruction can combine conditions through AND or OR, recombine subconditions, and redesign the temporal fields when the child uses a sequence encoder.

Fixed execution settings. The default policy uses three exploration rounds per session, $p_{\text{cross}} = 0.5$, equal mixing of diversity and tournament selection for crossover parents, and uniform mutation-type weights. Exploration is also used when fewer than two parents are available. Tournament selection samples up to three archived candidates and chooses the strongest; diversity selection pairs a strong candidate with one differing in condition–time identity or encoder. The parent archive retains the strongest checkpoint within each normalized condition–time–encoder niche. The separate checkpoint pool replaces an identical effective configuration only when validation improves. These rules are shared by RSI-on and RSI-off until an admissible RSI patch changes a controllable search setting. Session length, metric schedules, and stopping limits remain fixed run settings.

Ensemble eligibility and calibration. A conditioned expert enters the ensemble only if its VALID-selection support contains at least 50 samples across two signal periods and its IC, ICIR, and RankIC are finite and strictly positive. An optional VALID-only add-one gate for incremental ensemble value is disabled in the reported default; neither RSI setting performs coalition refitting or Factor Shapley estimation. CSI300, CSI500, and M5 use isotonic calibration for local predictions; Numerai and PEMS-BAY use uncalibrated, reliability-weighted local predictions. All five datasets use the fixed global blend $\alpha = 0.05$ in Equation 9, with the global predictor serving as fallback when no local factor applies. These dataset-profile settings are the same for RSI-on and RSI-off.

Reflection cadence and run-specific protocol. A reflection requires a successful VALID-monitor evaluation and at least 20 completed rounds since the previous eligible event by default; periodic ensemble evaluation may delay it. Missing or invalid patches leave the policy unchanged. Policy versions, parent identities, plans, checkpoint metrics, and update decisions are retained to resume a run consistently. In the inspected CSI500 adapter, the prescribed selection metric is ICIR through round 250; thereafter, a successful VALID-monitor ICIR above 0.4 can trigger alternation between IC and RankIC. This schedule applies to both RSI settings and is not learned by the controller. Historical results follow the configurations of their original runs.

L.1 VALIDATION CHALLENGE AND ROLLBACK

Let $\mathbf{M}$ contain the IC, ICIR, and RankIC of an eligible VALID-monitor ensemble evaluation. Fixed dataset-specific references $b_m > 0$ normalize their different scales. The controller uses the mean normalized score

$$J(\mathbf{M}) = \frac{1}{3} \sum_{m \in \{\mathrm{IC},\mathrm{ICIR},\mathrm{RankIC}\}} \frac{M_m}{b_m}. \tag{11}$$

The next eligible evaluation after a provisional update is compared with the pre-update baseline. The update is retained only if

$$J(\mathbf{M}^{\mathrm{after}}) > J(\mathbf{M}^{\mathrm{before}}) \quad \wedge \quad \min_m \frac{M_m^{\mathrm{after}} - M_m^{\mathrm{before}}}{b_m} \geq -0.05. \tag{12}$$

The second condition prevents an improvement in the joint score from masking a normalized decline greater than 0.05 in any metric. If either condition fails, the previous policy is restored for subsequent search, while factors already generated remain in the pool. Because the factor population changes between evaluations, this challenge guides search rather than identifying the causal effect of an isolated policy update.

M ADDITIONAL TEMPORAL-FACTOR ANALYSES

We report the remaining distribution and performance views from the same RSI-off catalogs used in Section 4.5. Distribution plots count saved non-global records with encoder, window-end lag k, and input-range length L available, without a minimum-frequency or performance filter. These comprise 855 CSI300, 777 CSI500, and 391 PEMS-BAY records. Heatmaps additionally require complete IC, ICIR, and RankIC measurements, leaving 847, 557, and 387 records, respectively. Different eligibility rules mean that distribution and performance plots need not have identical support.

Heatmap statistics use the stored TEST-lockbox measurements on CSI300 and PEMS-BAY and VALID-selection measurements on CSI500. Here, TEST-lockbox is the source-record designation; it does not imply that the complete adaptive search had an untouched final holdout (Section **??**). Each cell is a median over eligible factors, with no minimum cell count. Gray denotes unavailable cells, not zero performance. Color limits are computed separately for each panel from the 5th–95th percentiles of observed cell medians; when this interval crosses zero, the limits are made symmetric around zero. Values outside the limits saturate in color but retain their numerical annotations. The figures describe factor-level heterogeneity and do not provide confidence intervals or independent-run significance tests.

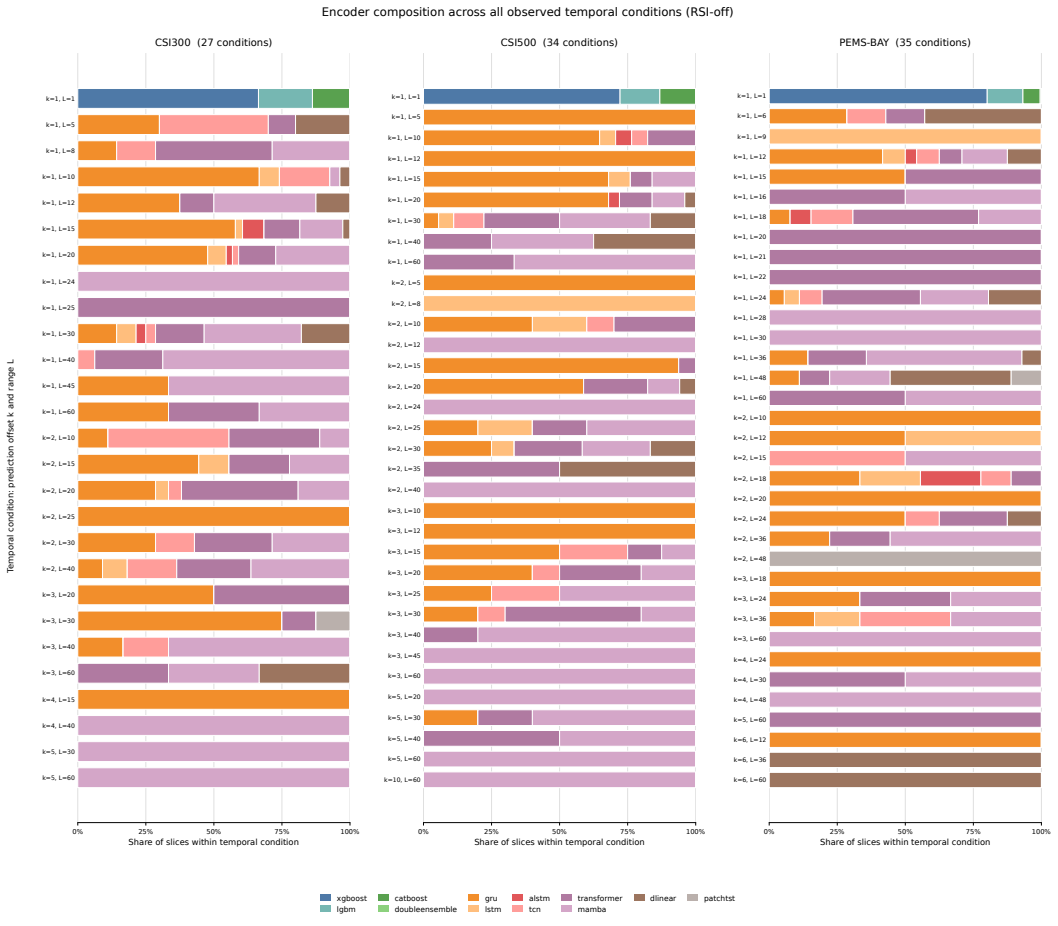

Figure 10: Full encoder composition across temporal conditions. The three panels show CSI300, CSI500, and PEMS-BAY. Horizontal axes give within-condition record percentages; vertical axes list every observed (k, L) pair. Colors identify encoders. Each row sums to 100%; conditions are not weighted by their catalog frequency. Counts describe encoder composition conditional on a temporal pair, not the marginal frequency of that pair.

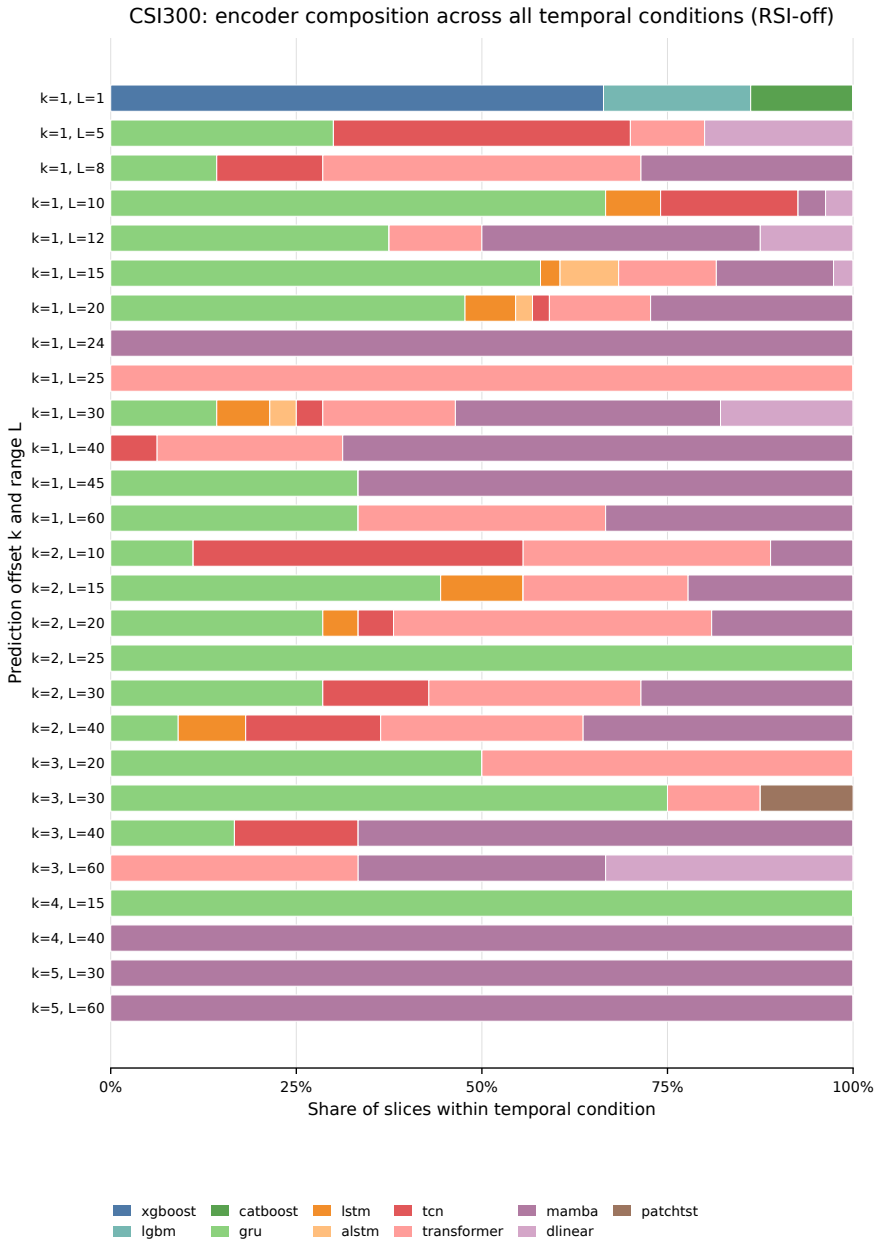

Figure 11: Complete CSI300 encoder distribution. The horizontal axis gives the percentage of records within each condition; the vertical axis lists all 27 observed (k, L) conditions. Colors identify encoders, and each row sums to 100%. Counts describe encoder composition conditional on a temporal pair, not the marginal frequency of that pair.

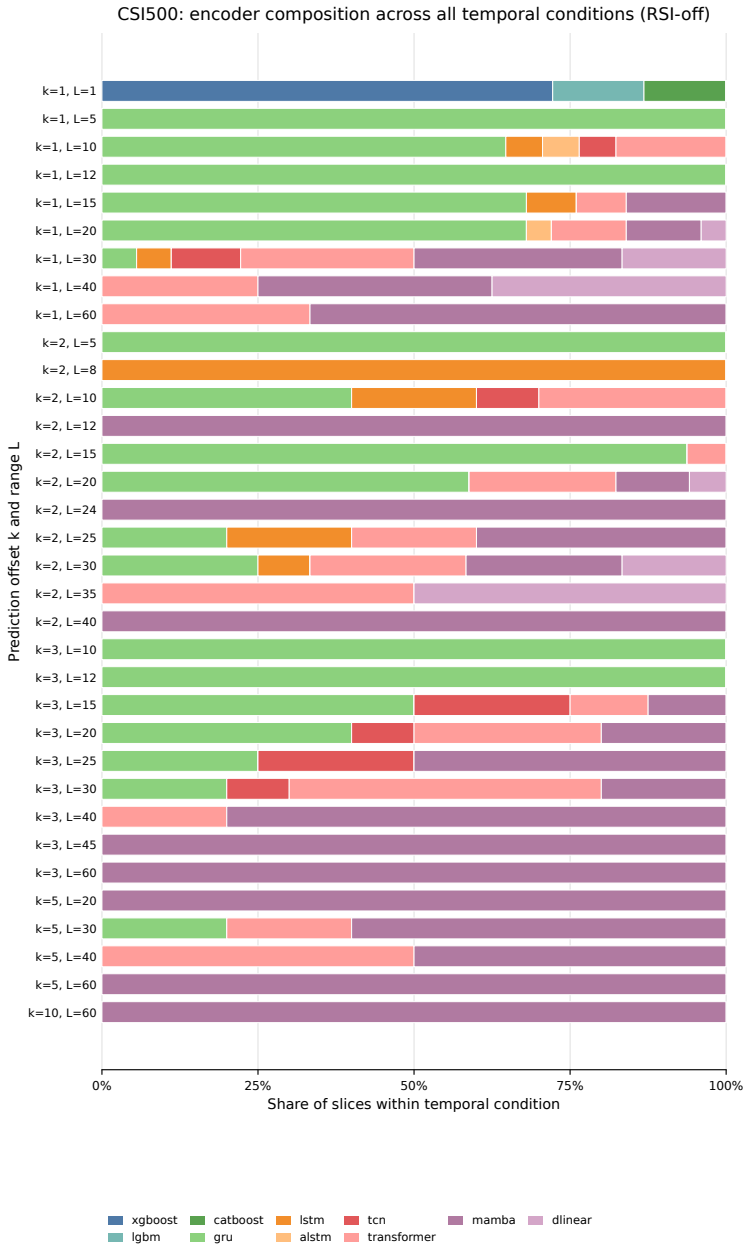

Figure 12: Complete CSI500 encoder distribution. The horizontal axis gives the percentage of records within each condition; the vertical axis lists all 34 observed (k, L) conditions. Colors identify encoders, and each row sums to 100%. Counts describe encoder composition conditional on a temporal pair, not the marginal frequency of that pair.

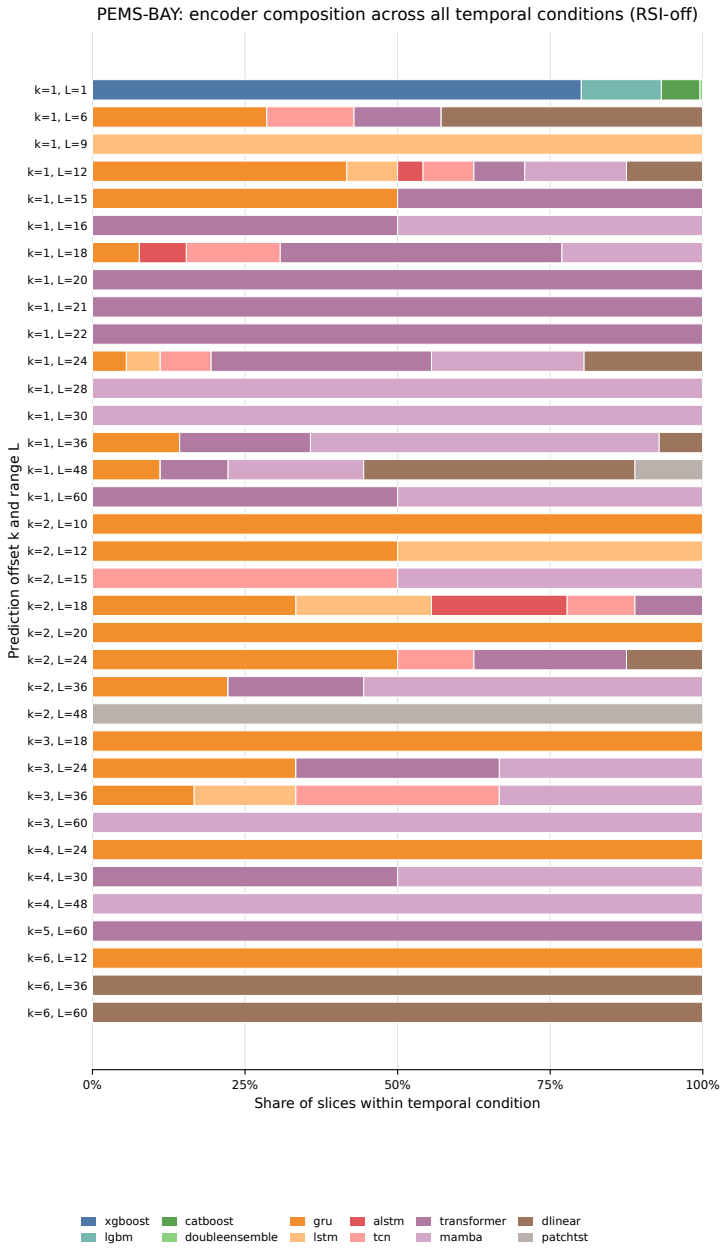

Figure 13: Complete PEMS-BAY encoder distribution. The horizontal axis gives the percentage of records within each condition; the vertical axis lists all 35 observed (k, L) conditions. Colors identify encoders, and each row sums to 100%. Counts describe encoder composition conditional on a temporal pair, not the marginal frequency of that pair.

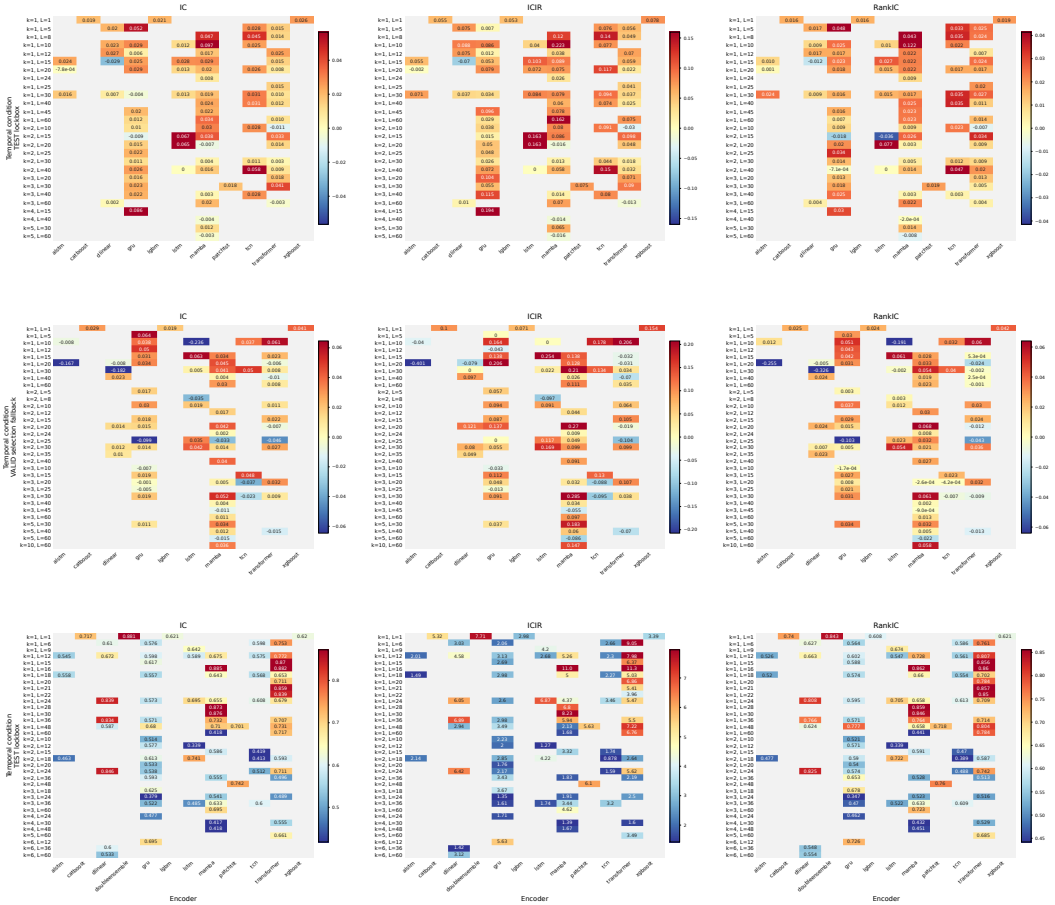

Figure 14: Complete encoder–condition performance audit. Rows correspond to CSI300, CSI500, and PEMS-BAY; columns correspond to IC, ICIR, and RankIC. Each horizontal axis lists encoders, and each vertical axis lists all observed (k, L) conditions. Annotations report the median native metric within a cell. All conditions are retained. CSI300 and PEMS-BAY use TEST lockbox; CSI500 uses VALID selection. Gray cells are unavailable; colors use independent panel scales as defined above. Singleton cells are included.

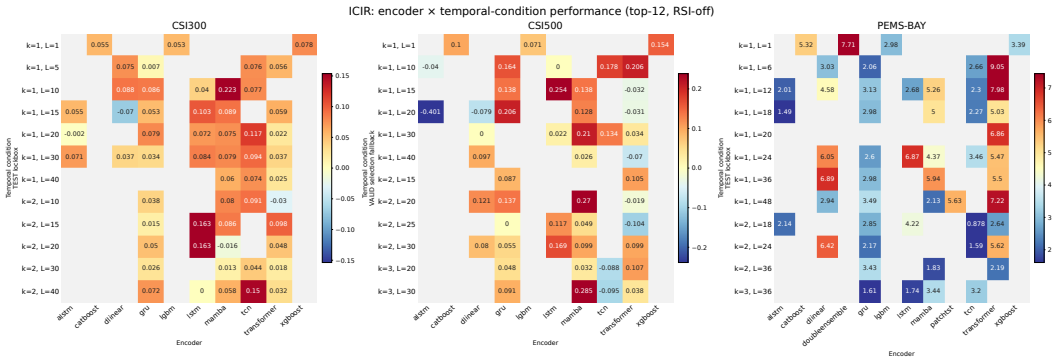

Figure 15: Encoder–condition ICIR. Left to right: CSI300, CSI500, and PEMS-BAY. Horizontal axes list encoders; vertical axes list the 12 conditions with the largest complete-record support in each dataset, ordered by (k, L). Annotations report median native ICIR within a cell. CSI300 and PEMS-BAY use TEST lockbox; CSI500 uses VALID selection. Gray cells are unavailable; colors use independent panel scales as defined above. Singleton cells are included.

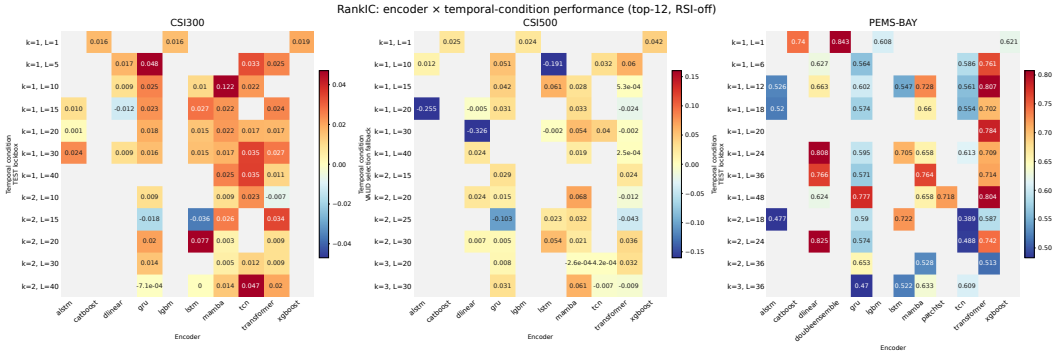

Figure 16: Encoder–condition RankIC. Left to right: CSI300, CSI500, and PEMS-BAY. Horizontal axes list encoders; vertical axes list the 12 conditions with the largest complete-record support in each dataset, ordered by (k, L). Annotations report median native RankIC within a cell. CSI300 and PEMS-BAY use TEST lockbox; CSI500 uses VALID selection. Gray cells are unavailable; colors use independent panel scales as defined above. Singleton cells are included.

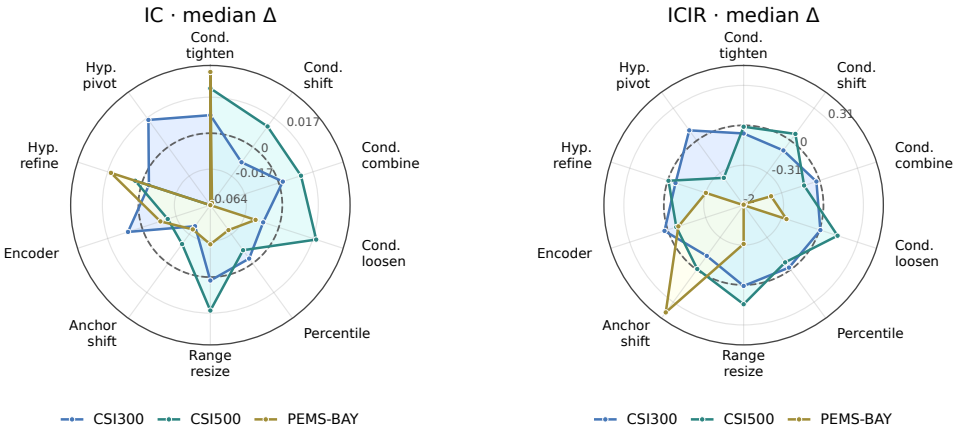

Figure 17: Parent-relative IC and ICIR outcomes of targeted factor adjustments. Panels (a) and (b) report median child-minus-parent IC and ICIR, respectively, across ten requested mutation types (angular labels). Radial positions show signed changes, colors distinguish CSI300, CSI500, and PEMS-BAY, and dashed rings mark zero.

M.1 Targeted-adjustment measurement provenance

Figure 3(b) and Appendix Figure 17 apply a common filtering and aggregation rule to the saved parent–child table: retain the ten requested mutation types, discard missing changes separately for each metric, and take the median child-minus-parent change within each dataset–operator cell. Common filtering does not imply identical evaluation splits. CSI300 and PEMS-BAY use saved single-factor TEST measurements; CSI500 uses VALID-selection history matched by expert identity and the latest available round at or before the expert round, falling back to the nearest recorded round when necessary. The radars use RSI-off cells whenever available. Only CSI500 temporal anchor shifting uses RSI-on measurements, with two pairs, because its RSI-off cell is absent. No other displayed cell uses this fallback. These quantities describe individual-factor outcomes, not changes in fused ensemble performance or a randomized operation comparison. All three metrics have the same observed support per displayed cell, ranging from 1 to 16 pairs.

N REFINING USEFUL LINEAGES ACROSS TEMPORAL ENCODERS

On PEMS-BAY at round 110, RSI-on slightly raises median single-factor TEST IC (0.6368 versus 0.6317). Fusion exceeds individual-factor IC on 99.7% and 100% of matched supports under RSI-on and RSI-off, with median gains of 0.0316 and 0.0299, respectively (Figure 18).

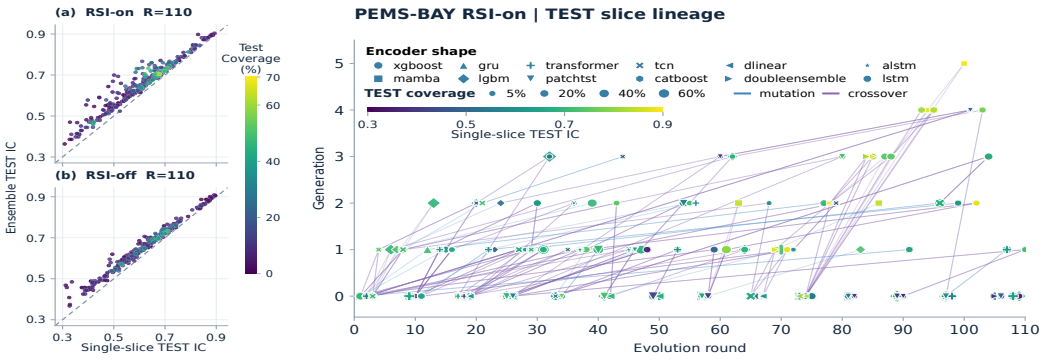

Figure 18: PEMS-BAY factor quality, fusion, and lineage at round 110. Left: single-factor TEST IC (horizontal axis) versus matched-support ensemble TEST IC (vertical axis) for 387 RSI-on and 389 RSI-off factors. Right: creation round (horizontal axis) versus lineage generation (vertical axis) of retained RSI-on factors, with mutation and crossover links.

On PEMS-BAY, RSI-on yields slightly stronger individual factors, while fusion exceeds single-factor TEST IC on at least 99.7% of matched supports under both policies (Figure 18). Of the 39 top-decile RSI-on factors, 36 emerge after round 60 and 26 originate from crossover; the elites span multiple encoder families. These observations are consistent with adaptive search developing strong, diverse descendants and fusion combining their signals.

O FACTOR ORIGINS: RANKIC, IC, AND ICIR

Figure 19 reports the RankIC analysis discussed in Section 4.6.1; Figures 20 and 21 provide the complementary IC and ICIR views. Both use the same endpoint populations, historical-score eligibility, operation labels, and condition-matched coverage definition. The upper distributions distinguish changes in typical factor quality from changes in the high-performance tail; the lower bars show the applicability of factors in each performance interval. Bins are metric-specific, and the coverage bars are medians rather than counts or disjoint portions of the TEST set.

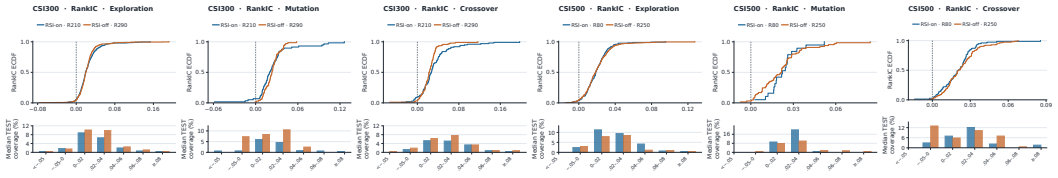

Figure 19: Single-factor TEST RankIC and coverage at endpoint snapshots. Panels (a)–(c) show exploration, mutation, and crossover on CSI300 (RSI-on/off rounds 210/290); panels (d)–(f) show the same on CSI500 (rounds 80/250). Upper plots show RankIC distributions on each factor's applicable TEST samples; lower plots show median coverage within RankIC intervals. Coverage is the fraction of all TEST samples to which a factor applies.

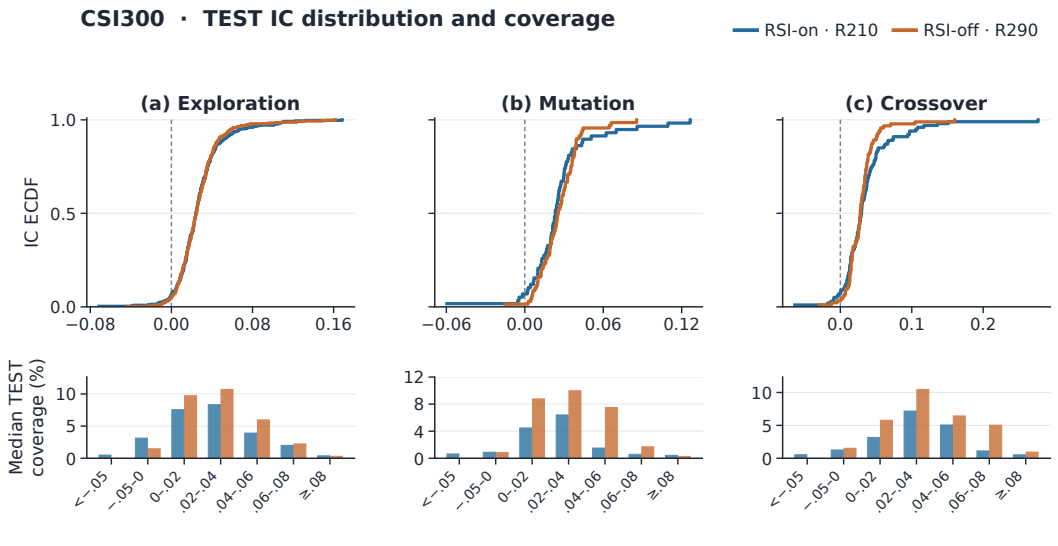

Figure 20: CSI300 TEST IC distributions and coverage for RSI-on at round 210 and RSI-off at round 290. Columns show exploration, mutation, and crossover. Top: single-factor IC on condition-matched TEST samples (horizontal axis) and the within-arm, within-operation empirical cumulative distribution (vertical axis). Bottom: IC intervals (horizontal axis) and median TEST coverage in percent (vertical axis), relative to 197,012 TEST samples. Blue is RSI-on; orange is RSI-off. The common evaluable subsets contain 511 RSI-on and 640 RSI-off factors; eligibility and exclusions match Figure 19.

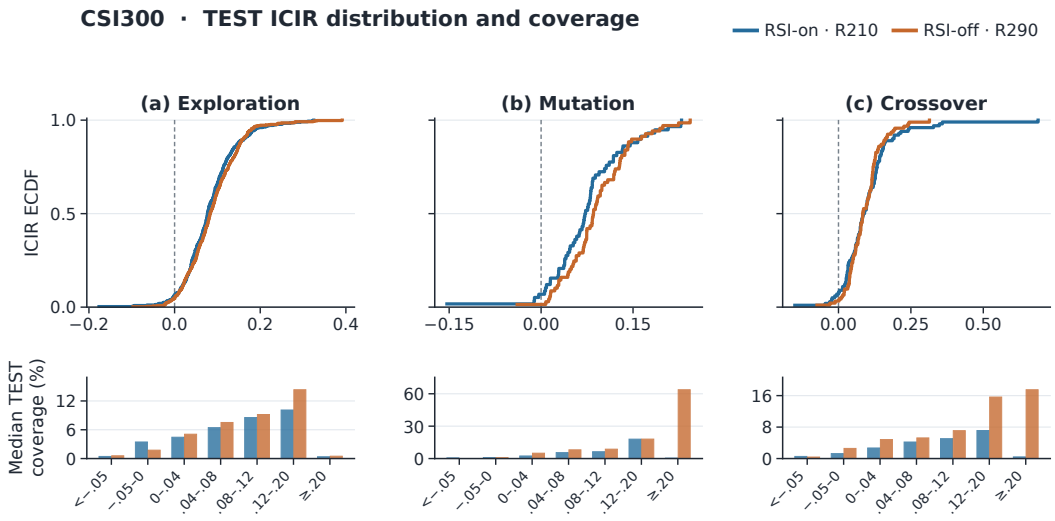

Figure 21: CSI300 TEST ICIR distributions and coverage for the same endpoint subsets. Columns show exploration, mutation, and crossover. Top: single-factor ICIR on condition-matched TEST samples (horizontal axis) and the within-arm, within-operation empirical cumulative distribution (vertical axis). Bottom: ICIR intervals (horizontal axis) and median condition-matched TEST coverage in percent (vertical axis), relative to 197,012 samples. Blue is RSI-on; orange is RSI-off. Intervals follow the ICIR scale and differ from those in the IC and RankIC figures.