Conditional Factors: Discovering When to Use a Formula

Anonymous authors
Paper under double-blind review

Abstract

Formulaic alpha mining produces short, executable predictors, but a formula alone does not say which assets, in which information states, it should rank. We introduce *conditional factors*: programs that pair an applicability predicate with a ranking formula, discover the two jointly, and execute them as a pair. In U.S. equities, a shared symbolic financial state enters both the predicate and the score; the generator is rewarded by rank correlation measured only on the predicate's support, subject to support-validity constraints. An expected-rank analysis explains why the support matters: changing the comparison set can change the optimal order. We use fixed-formula controls to ask three questions. First, conditional evaluation changes the selected formula for 155 of 192 predicates in a frozen candidate library, although the ranking advantage of the new choice changes sign across years. Second, holding formulas fixed, predicates raise executed returns over always-on use in both markets and, in cryptocurrency, above same-count random support; the U.S. matched-support ranking gain, however, is positive in 2024 and negative in 2025. Third, a signed decomposition of executed returns shows two routes by which predicates help: amplifying favorable use and attenuating unfavorable use. On 2024 cryptocurrency spot data, the 20-day spot-state program earns the highest return among the reported GP, PPO, PS-Tree, and AlphaSAGE-8 comparisons; in a preregistered 2025 extension its gain over always-on use persists while its absolute return trails the external methods. Formula-level applicability is thus an executable and testable part of factor discovery whose value varies across periods.

1 Introduction

Formulaic alpha mining converts observable market information into short expressions that rank assets by expected future return. AlphaGen, for example, searches for complementary formulas over six price and volume inputs and feeds the quality of their linear combination back to generation (Yu et al., 2023). A price-change formula found this way leaves one decision open: *which assets should it rank, and at what information stage?* A reversal formula, for instance, might order firms well shortly after a new annual report and poorly otherwise. We study how to discover a formula together with the condition under which it is used.

Characteristic-based asset pricing suggests that this condition is informative. IPCA makes factor exposures depend on firm characteristics (Kelly et al., 2019), nonlinear interactions among predictors improve return forecasts (Gu et al., 2020), and earnings momentum helps explain price momentum (Novy-Marx, 2015). We use such firm information not as another input to one global predictor, but to describe where an individual formula applies. Because accounting reports arrive asynchronously and are later revised, every U.S. input is reconstructed from the records visible at the decision time.

Figure 3 shows a program found by our method. When a new annual report first becomes visible, the program keeps firms whose reported profitability changed in a particular direction, then ranks only those firms by a price formula that contains the same financial state. The state therefore decides both *where* the program applies and *how* it scores. We call the complete tuple of predicate, score, direction, and forecast horizon a *conditional factor*. Each factor is activated independently at execution, may overlap with other factors, and contributes to a fixed-budget combination only on its own support.

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.

To examine applicability beyond corporate fundamentals, we also study cryptocurrency, where spot trading provides a different source of observable state. Prior work documents cross-sectional cryptocurrency momentum (Liu et al., 2022), horizon-dependent links between returns and trading volume (Fousekis & Tzaferi, 2021), and predictive information in order flow (Anastasopoulos et al., 2026). These findings motivate asking when a price–volume formula should be used. Our cryptocurrency programs make the question explicit by pairing each formula with an executable predicate built from price–volume parent conditions and spot-trading state.

Making the support part of the discovered object changes the learning problem. Existing systems mine formulas using quality measured on the available cross-section and often assemble them into collections (Yu et al., 2023; Shi et al., 2025; Zhu & Zhu, 2025; Chen et al., 2026); partitioned regression such as PS-Tree learns local experts inside one aggregate predictor (Zhang et al., 2022). For a conditional factor, the predicate determines which assets are compared, and the comparison set in turn determines which ordering is optimal (Section 2). Evaluating a formula on the wrong set can therefore select the wrong formula. Our contributions are:

- **Formulation.** We define conditional factors as executable (p, s, d, H) programs and show with an expected-rank argument that the predicate's support belongs in the ranking objective.
- **Method.** We construct predicates and scores jointly, reward complete programs by within-support rank quality and support validity, and preserve each predicate–score pair and its source-period direction at execution.
- **Evaluation.** We separate three questions—formula selection, fixed-formula applicability, and complete-program use—with fixed-formula controls in U.S. equities and cryptocurrency, and report periods where the gains reverse together with those where they hold.

2 Problem Setup

2.1 Conditional factors

At decision time t, let $\mathcal{F}_t$ be the available information and $\mathcal{U}_t$ the eligible assets. A conditional factor is

$$f = (p, s, d, H), \qquad \mathcal{A}_t(p) = \{i \in \mathcal{U}_t : p(i,t) = 1\},$$

where p is a Boolean *predicate*, s a scoring formula, $d \in \{-1, +1\}$ a direction fixed during source-period training and selection, and H the forecast horizon. We call $\mathcal{A}_t(p)$ the *support* of the program at time t. Execution follows a WHEN/USE rule:

$$\boxed{\text{WHEN } p(i,t) = 1, \quad \text{USE } \alpha_f(i,t) = d\, s(i,t) \text{ to rank within } \mathcal{A}_t(p).}$$

A program with empty support emits no ranking, and the outputs of active programs are combined by a predetermined rule. Supports of different programs may overlap.

2.2 Why the support belongs in the objective

RankIC is the Spearman correlation between scores and future returns across the compared assets, so its target order depends on which assets are compared. Figure 1 fixes a joint distribution of future returns for three assets. Compared only with j, asset i wins with probability 0.6; once k is added, j has the higher expected rank. Restricting a globally optimal order to a subset therefore need not give the optimal order on that subset.

With future H-period return $Y_{i,t}^{(H)}$, the expected target rank of asset i within the support is

$$m_{i,t}^{p,H} = 1 + \sum_{j \in \mathcal{A}_t(p) \setminus \{i\}} \Pr(Y_{i,t}^{(H)} > Y_{j,t}^{(H)} \mid \mathcal{F}_t),$$

that is, one plus the expected number of supported peers that i beats.

Proposition 1 (Optimal order on a fixed comparison set)**.** *Conditional on $\mathcal{F}_t$ and a fixed finite asset set, if future returns have no ties, sorting by $m_{i,t}^{p,H}$ maximizes expected Spearman RankIC among untied predicted permutations. Dependence across assets is retained in the conditional probabilities.*

Fixed future-return outcomes

Probability	Y_i	Y_j	Y_k
0.3	3	2	1
0.4	1	3	2
0.3	2	1	3

Expected ranks within the set

$\{i, j\}$: $(1.6, 1.4) \Rightarrow i \succ j$

$\{i, j, k\}$: $(1.9, 2.1, 2.0) \Rightarrow j \succ k \succ i$

The ideal order of i, j reverses under the same distribution.

Figure 1: Ranking depends on the comparison set. The distribution of future returns is fixed; a larger rank denotes a higher return.

Appendix C gives the proof. The predicate changes the peers in the sum and can thereby change the target order even when the predictive distribution is unchanged. A program should therefore be scored on its own support, which is what our search feedback does.

3 Method: Joint Discovery of Predicates and Formulas

Figure 2 summarizes the method. A generator samples expression and predicate actions to form a complete program; the predicate determines the support; rank quality on that support, together with support validity, updates the generator. Selected programs are frozen as pairs and executed independently.

3.1 Program construction

For U.S. equities, a symbolic financial state g_θ, a price–volume expression u_ϕ, and a fusion action are built within a prespecified visibility stage c and state template k:

$$p_f = C_c \wedge S_k(g_\theta), \qquad s_f = \Psi(g_\theta, u_\phi).$$

C_c tests how recently a new annual report became visible (the *clock*), S_k tests the high or low side of the state or of its monthly change, and Ψ is a restricted fusion and smoothing operator. Because

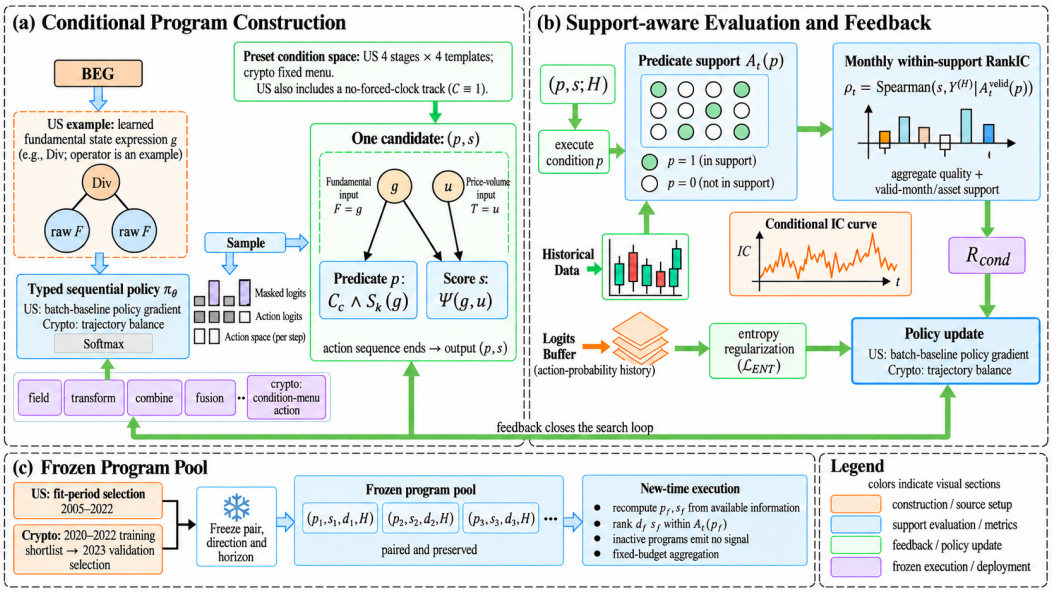


Figure 2: Construction, within-support feedback, and frozen execution of conditional programs. In the U.S. instance, the learned fundamental state $F = g_\theta$ enters both the predicate and the score; the separately generated price–volume expression $T = u_\phi$ enters the score through the sampled fusion $s = \Psi(F, T)$. Cryptocurrency instead uses a price–volume parent and a spot-state menu. Selection preserves each predicate–score pair and its source direction. Market-specific pseudocode and component provenance appear in Appendix A.6.

the same g_θ enters both sides, changing it alters both the supported assets and their scores. The four visibility stages, four state templates, thresholds, and typed grammar are fixed before search; the generator learns expressions and fusion actions within these 16 cells. Cryptocurrency uses the same (p, s, d, H) interface: the generator produces price–volume formulas and chooses among fixed price–volume parent predicates, optionally extended by a spot-trading state such as taker-buy share (Appendix A).

Figure 3 expands one U.S. program, chosen as a lower-median representative by source-period quality before any evaluation summary was read. $F = g_\theta$ is a financial-state rank and $T = u_\phi$ a price–volume rank; F enters both the predicate and the score.

3.2 Within-support feedback

For a U.S. candidate f, each month supplies the Spearman correlation $\rho_{f,t}$ between its scores and matured labels on $\mathcal{A}_t(p_f)$. The reward is

$$R(f) = \kappa_f |\overline{\rho}_f|, \qquad \kappa_f = \sqrt{\frac{M_f}{M_f + 36}} \min\left(1, \sqrt{\frac{\widetilde{n}_f}{100}}\right),$$

where M_f is the number of valid months, $\widetilde{n}_f$ the median valid support size, and $\overline{\rho}_f$ the mean correlation over valid months. The predicate thus affects both the correlations and whether a month can be evaluated; Appendix C shows that $1 - R$ is a shrunken within-support rank loss. The direction d is the sign of the source-period mean conditional correlation. Cryptocurrency rewards directed conditional quality and requires minimum asset and paired-date coverage (Appendix A).

3.3 Generator, selection, and frozen execution

A typed policy samples actions sequentially, and the complete (p, s) receives one within-support feedback, so the reward always refers to the program actually sampled. The U.S. generator uses a batch-centered policy gradient; the cryptocurrency generator maps quality to a terminal reward and uses trajectory-balance updates; both add entropy regularization (Appendix A.5). After the search budget, predetermined source-period rules deduplicate and select programs: within visibility-stage and template cells in U.S. equities, and through a training shortlist followed by validation selection in cryptocurrency. Algorithm 1 states the shared procedure.

At a new decision time, each frozen program recomputes its predicate and score from then-visible information, activates independently, and contributes only on its support. Preserving the pair guarantees that the program used later is the program evaluated during discovery. The direction d is never re-estimated: all fixed-formula controls below use the same d, so they measure the effect of *where* a formula is used rather than a reorientation of its predictive relation.

A discovered U.S. program: a newly visible annual report, earnings changes, and price changes

Visible inputs: changes between reports in EBITDA and pretax income (PI), sales (SALE), closing price, and $W = (\mathrm{HIGH} + \mathrm{LOW} + \mathrm{CLOSE})/3$.

$$\begin{aligned} F_t &= R_t(D(\Delta_{\mathrm{report}}\mathrm{EBITDA}, \mathrm{SALE}) - D(\Delta_{\mathrm{report}}\mathrm{PI}, \mathrm{SALE}))\,, \\ T_t &= R_t(W_t - W_{t-5} + \mathrm{CLOSE}_t - \mathrm{CLOSE}_{t-20})\,, \\ B_t &= T_t\ \mathrm{sgn}(F_t - 0.5), \\ \alpha_t &= R_t(\mathrm{MA}_{10}(B))\,\mathrm{sgn}(F_t - 0.5). \end{aligned}$$

WHEN: a new fiscal-year fourth-quarter report first becomes visible in the current decision interval and $F_t \geq 0.5$.

USE: rank supported assets by α_t; the program is inactive for other assets. F selects the support and also enters B and the final score; the ten-day smoothing precedes filtering.

Figure 3: From then-visible information to within-support ranking. R is the rank over the eligible cross-section, D is protected division, and Δ_{report} compares adjacent visible reports. Appendix A defines the operators and Appendix G lists further representatives, each with its source-period direction.

Input: source-period visible information and matured labels; fixed grammar, predicate menu, search budget, and pool budget.
1. Sample expression and composition actions within the grammar and menu to obtain a complete (p, s).
2. Execute p to obtain the support; evaluate s on it, fix the source direction, and check numerical and support validity.
3. Update the policy with within-support quality and support feedback; repeat until the search budget ends.
4. Deduplicate and select by the predetermined source-period rule, preserving each predicate–score pair.
5. Freeze every (p, s, d, H) and the budget; record invalid, rejected, and unfilled outcomes.
Use: recompute predicates and scores, rank only within each support, and combine by the fixed rule.

Algorithm 1: Discovering factors as conditional programs. Supports may overlap between programs; market-specific templates, source direction, and the selection schedule are fixed in advance.

4 Experimental Setup

Markets. U.S. equities combine CRSP prices and volumes with historical Compustat Snapshot fundamentals; only records visible at the decision time enter a score, and later revisions never backfill history. U.S. programs are trained on 2005–2022; 2023 diagnoses the frozen pool without changing any program; 2024–2025 is the evaluation window. Cryptocurrency uses official Binance daily spot records with a dynamic top-50 universe by trailing quote volume. Programs are trained on June 2020–2022, selected on 2023 by a fixed rule, and evaluated in 2024. We report 5-day and 20-day horizons separately and, for U.S. equities, also an equal 5/20-day mix; every result aggregates three training seeds. Each seed has 64 U.S. or 8 cryptocurrency programs per horizon (Appendix Table 8).

Comparison methods. Search baselines share our market inputs and execution rules. The U.S. AlphaSAGE adapter runs trajectory-balance search on the common typed representation with a comparable sample budget and pool size; GP and PPO build pools in each market's common representation. *State only* retrains our U.S. method without the clock, at equal capacity and budget. In cryptocurrency, AlphaSAGE-8 selects eight formulas under our train/validation rule, and AlphaSAGE-50 keeps its original formulas and weights. PS-Tree is an independent structural comparison that refits one partitioned regression predictor per market and horizon. Appendix D gives inputs, budgets, and objectives.

Fixed-formula controls and metrics. To measure what a predicate contributes, we keep the score, direction, dates, and execution rule fixed and change only the support. The *always-on* control ranks every eligible asset; *same-count random support* samples, at each decision time, as many assets as the predicate selects from a prespecified reference set; in cryptocurrency an *exposure-matched* control also equalizes the capital deployed. For a frozen program, $\Delta = R_{\text{full}} - R_{\text{always-on}}$ is the conditional increment. We report it alongside absolute performance and support coverage, because an increment over a weak always-on reference can be large while absolute performance is poor. Executed accounts use 5 and 10 basis points of one-way cost (Appendix B).

Evaluation history. U.S. pools were selected through 2022; 2023 is diagnostic, 2024 later entered development analyses, and 2025 was not used for fitting or selection. Cryptocurrency pools were trained through 2022 and selected in 2023; no search or selection setting was tuned on 2024. Earlier 12-asset versions had been evaluated on 2024 before we widened the universe and added spot-state features, so 2024 is the first test of these frozen programs but not an unseen period. The 2025 extension was preregistered for the conditional programs and AlphaSAGE-8; GP and PS-Tree were added after it was read. Later controls on opened outcomes are retrospective; source-year curves are in-sample.

5 Results

5.1 Q1: Does conditional evaluation change formula selection?

Yes, but the new choice does not rank better in every year. For each of 192 predicates in a frozen U.S. candidate library, we select one formula either by full-universe quality or by within-support

quality, fixing the source direction in both cases. The selected formula changes for 155 of 192 predicates. We then compare the two selections on the same predicate support. Table 1 shows that the within-support choice ranks better in 2024 but worse in 2023 and 2025, with at most two of three seeds agreeing. Portfolio accounts built from both selections have negative three-seed mean returns in 2024–2025 (Appendix E.2). Support-aware evaluation is therefore a material change to discovery, not a guaranteed improvement.

5.2 Q2: Does a frozen predicate carry applicability information?

Yes in executed returns, including beyond same-count random support in cryptocurrency; the U.S. ranking gain, however, reverses in 2025. Figure 4 holds discovered scores and directions fixed and changes only the support. In U.S. equities, the full clock-plus-state predicate raises cumulative net return of the 5/20-day mix over the same always-on formulas by 1.02 percentage points, with a positive increment for all three seeds (panel a). The clock alone does slightly better (-0.893% versus -1.130%), so the added financial state improves formula-level ranking but not mean return in this setting (Appendix E.3).

For 20-day cryptocurrency spot-state programs (panel b), the full predicate exceeds both the parent-only and always-on versions on average, although seed 1 is slightly below its own always-on counterpart (21.71% versus 21.84%). The full predicate earns 24.37%, compared with 14.98% for same-count random support drawn from the whole universe and 22.19% for random support drawn within the parent predicate. With capital deployment matched, price–volume predicates improve over always-on by 8.60 and 8.71 points at 5 and 20 days, and added spot states improve over their parents by 3.14 and 2.02 points, positive for every seed at both costs. U.S. predicates change the capital deployed, so their return increment also includes an exposure effect.

Panel c follows the same 192 frozen U.S. programs across the full cycle. Their within-support RankIC exceeds that of same-count random support by $+0.01519$ in 2024 (all three seeds positive) but falls short by 0.00223 in 2025 (one seed positive). Roughly half of program-months have no defined within-support RankIC, which we report as missing rather than zero (Appendix Figure 6). Applicability information is therefore present but period-dependent.

5.3 Q3: How do complete programs perform in use?

Absolute outcomes are mixed; predicates help either by amplifying favorable use or by limiting unfavorable use. Table 2 compares complete programs. In U.S. equities, clock-plus-state loses less than every comparison at both horizons, but its return remains negative. In 2024 cryptocurrency, the 20-day spot-state program earns the highest return of all listed methods, whereas both 5-day programs trail PS-Tree and GP. Appendix E.1 reports risk, cost sensitivity, per-seed returns, and return paths.

To see how predicates change returns, we split each program's return difference from its fixed-formula reference by whether the reference earned a positive or negative return on that day (Table 3); this is an ex post decomposition, not a trading rule. The U.S. clock-plus-state predicate and the added cryptocurrency spot state mainly avoid losses: they give up some gains on positive days but avoid more on negative days. Cryptocurrency price–volume predicates instead mainly strengthen gains, with a small cost on negative days. The two effects are distinct and deserve separate measurement (Appendix E.4).

Table 1: Within-support versus full-universe formula selection for 192 U.S. predicates. Paired RankIC difference is the signed within-support Spearman of the within-support choice minus that of the full-universe choice, evaluated on common executable support with directions fixed. Undefined values are not set to zero; all 192 predicates stay in the denominator.

Window	Paired RankIC difference ($\times 10^{-3}$)	Positive seeds	Valid predicates
2023	-6.425	1/3	187/192
2024	$+2.922$	2/3	185/192
2025	-1.986	1/3	182/192
2024–2025	$+1.256$	2/3	185/192

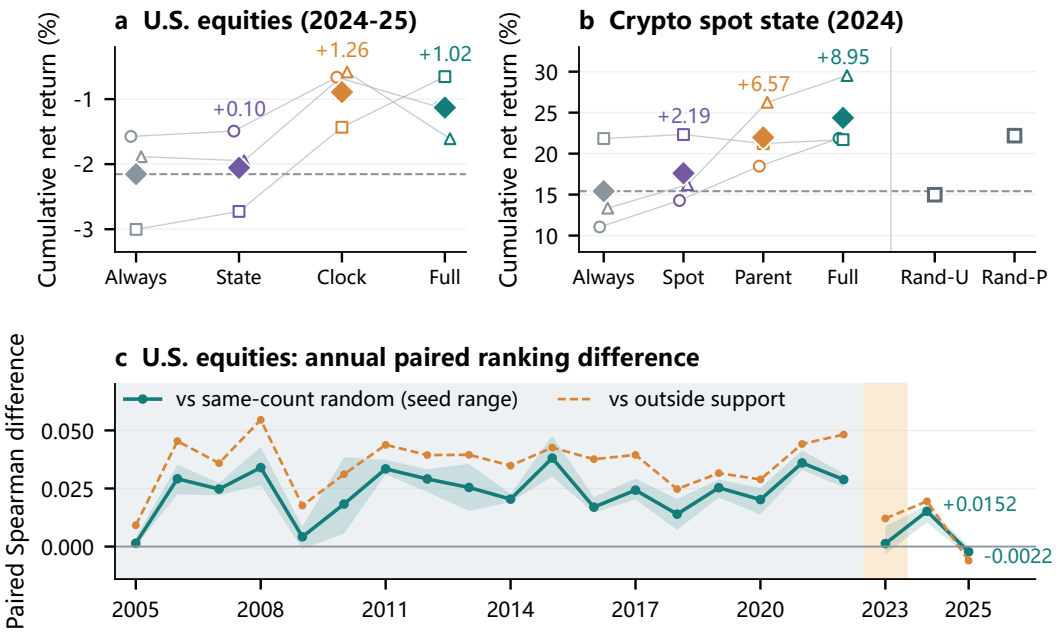


Figure 4: Fixed-formula use at 5-bp cost. Diamonds are three-seed means, small symbols individual seeds, and lines connect the same seed; annotations are increments over always-on. Full is clock plus state (a) or parent plus spot state (b). Rand-U and Rand-P draw the same number of assets at random from the whole universe and from the parent support. Panel c shows annual paired Spearman differences for the 192 U.S. programs against same-count random support (solid, shading: seed range) and against assets outside the support (dashed); gray marks fitted source years.

Table 2: Cumulative net returns: U.S. 2024–2025 and cryptocurrency 2024. Three equal-initial-capital seeds; 5-bp one-way cost and 3% annual borrowing. GP, PPO, and U.S. AlphaSAGE use common-representation search; PS-Tree is an independent predictor. Cryptocurrency AlphaSAGE-8 follows the shared eight-formula selection rule. Risk metrics appear in the appendix.

U.S.			Crypto		
Method	5-day	20-day	Method	5-day	20-day
Clock + state	−0.39%	−2.51%	Price–volume condition	5.24%	21.11%
State only (retrained)	−2.88%	−8.61%	Spot-state condition	3.84%	24.37%
AlphaSAGE (adapter)	−2.69%	−7.42%	AlphaSAGE-8	−1.46%	13.62%
GP (adapter)	−2.63%	−7.83%	GP (adapter)	6.31%	16.93%
PPO (adapter)	−2.67%	−7.08%	PPO (adapter)	4.66%	17.42%
PS-Tree (current inputs)	−3.50%	−7.75%	PS-Tree (current inputs)	13.68%	12.24%

Table 3: Signed use contribution (percentage points, three-seed mean, 5-bp cost). The total difference from the fixed-formula reference is split by days on which the reference return was positive or negative. U.S. clock + state and cryptocurrency price–volume use always-on as reference; the added spot state uses its parent predicate.

Predicate	Positive days	Negative days	Total change
U.S. clock + state	−5.63	+6.66	+1.02
Crypto price–volume	+9.99	−1.28	+8.71
Crypto added spot state (20-day)	−0.18	+2.21	+2.02

Applying the frozen cryptocurrency programs, directions, and universe rule to 2025 changes their absolute standing (Table 4). The 20-day spot-state gain over always-on is positive for every seed and exceeds same-count random support, yet its absolute return is close to zero and below AlphaSAGE-

8, GP, and PS-Tree. The 20-day price–volume predicate loses its 2024 gain, and the 5-day spot state falls below always-on.

Table 4: Preregistered cryptocurrency 2025 extension: three-seed mean net returns at 5 bp. Δ_{AO} and Δ_{rand} are percentage-point differences from the same formulas used always-on and on same-count random support; n.a. marks inapplicable controls for independent methods. GP and PS-Tree rows were added after the preregistered readout.

	5-day			20-day		
Method	Full	Δ_{AO}	Δ_{rand}	Full	Δ_{AO}	Δ_{rand}
Price–volume condition	3.25%	+2.26	+1.93	0.96%	−0.88	−0.88
Spot-state condition	2.24%	−0.29	+0.23	−0.07%	+3.83	+3.67
AlphaSAGE-8	8.18%	n.a.	n.a.	2.00%	n.a.	n.a.
GP (adapter)	−3.67%	n.a.	n.a.	4.58%	n.a.	n.a.
PS-Tree	7.06%	n.a.	n.a.	12.59%	n.a.	n.a.

6 Related Work

Formulaic alpha mining. Formulaic alphas map historically available information to predictions through inspectable inputs and operators (Kakushadze, 2016); searching many candidates raises false-discovery concerns (Harvey et al., 2016). Flexible nonlinear predictors improve return forecasts (Gu et al., 2020), while high-stakes prediction motivates interest in inspectable models (Rudin, 2019). Symbolic search retains explicit expressions: AutoAlpha uses hierarchical evolution (Zhang et al., 2020), deep symbolic regression learns a distribution over expressions (Petersen et al., 2021), AlphaGen rewards contribution to a linear pool (Yu et al., 2023), AlphaForge selects factors using recent performance and dynamically weights them (Shi et al., 2025), AlphaQCM uses distributional reinforcement learning to discover synergistic formulas (Zhu & Zhu, 2025), and AlphaSAGE explores expression graphs with a GFlowNet under quality, structural, and novelty rewards (Chen et al., 2026). These methods evaluate formulas on the available cross-section; we additionally discover the support on which each formula is evaluated and used.

Conditional asset pricing. The five-factor model adds profitability and investment to the market, size, and value factors (Fama & French, 2015); conditional models use lagged information (Ferson & Harvey, 1999), IPCA instruments time-varying exposures with characteristics (Kelly et al., 2019), and AP-Trees build characteristic-conditioned portfolios from interpretable partitions (Bryzgalova et al., 2025). We use characteristics and report visibility to define where a single formula is compared, and let the same state enter the score.

Local relationships and selective prediction. DDGroup finds interpretable subgroups with low-noise linear relationships (Izzo et al., 2023); PS-Tree combines symbolic features, adaptive partitions, and local experts through soft membership in one predictor (Zhang et al., 2022); AUCROSS designs AUC-oriented acceptance under coverage constraints (Pugnana & Ruggieri, 2023). These works motivate reporting quality together with coverage. Our programs instead solve repeated cross-sectional ranking problems, each keeping its own predicate, score, direction, and horizon (Appendix Table 9).

7 Discussion and Limitations

Evidence. Conditional factors can be discovered as explicit, executable predicate–formula pairs. Support-aware feedback changes which formula is chosen, and fixed-formula controls show that frozen predicates select informative supports and raise returns over always-on use in both markets. The U.S. ranking gain reverses in 2025; the retrospective controls and first evaluations of frozen programs have distinct evidential roles.

From applicability to returns. Valid applicability does not by itself yield stable returns: complete U.S. programs still lose money, and the best 2024 cryptocurrency program loses its absolute lead in 2025. How supports, activation, and aggregation should be designed to turn applicability into stable

portfolio returns remains open. Visibility stages and templates are prespecified; a learned-clock variant concentrates 386 of 400 programs on one stage (Appendix F). Seeds share one market path, about half of U.S. program-months have undefined within-support RankIC, and ranking, candidate-level, and return evidence use different statistical units.

A bridge between fields. Conditional factors turn characteristic-conditioned relations from asset pricing into the executable signals used in quantitative finance: financial state decides where a symbolic relation is tested and used. We hope this format encourages closer exchange between the two fields.

AI-assisted research. As language models are increasingly used to propose financial signals, a predicate, formula, information clock, and frozen direction give each proposal an inspectable form, and evaluation on its declared support makes the empirical claim checkable. We have not evaluated LLM-generated programs here.

8 Conclusion

We make an executable applicability predicate part of every discovered formula. Predicates and scores jointly enter search feedback, selection, and later use. An expected-rank analysis explains why the comparison set belongs in the objective, and a frozen library shows that support-aware evaluation changes formula selection. Fixed-formula controls in two markets show gains from use and their reversal across periods, and a signed decomposition separates amplifying favorable use from attenuating unfavorable use. Formula-level applicability is thus a learnable, executable, and measurable component of alpha discovery.

AI Use Statement

Generative AI tools assisted in framing research questions and hypotheses, developing the conditional-program formulation and experimental comparisons, and checking the expected-rank argument. They also assisted with literature discovery, implementation and review of data-processing, evaluation, and verification code, interpretation and organization of empirical evidence, translation and prose editing, and figure preparation, including a generated framework illustration subsequently checked against the implemented method. The reported market observations and numerical results come from the documented data and program outputs.

The authors selected the final design and interpretations, reviewed the mathematical argument, checked cited claims against source publications, inspected and tested AI-assisted code, and reconciled reported values and figures with frozen outputs and verification records. Implementation and accounting checks are distinct from statistical confirmation. We take responsibility for all claims, text, code, figures, and other AI-assisted artifacts in this paper.

Reproducibility Statement

Anonymized code and materials are available at `https://anonymous.4open.science/r/conditional-factors-D37A/` under the MIT License. They include the program generators, the comparison adapters, point-in-time data construction for both markets, the frozen program pools with their source directions, the costed execution engines, and a checker that verifies the aggregate results in Tables 2, 10, and 4 against the manuscript sources. Licensed CRSP and Compustat rows are not distributed; cryptocurrency histories are rebuilt from the public Binance archive, and external baseline runtimes are installed separately at pinned commits. Appendix H describes the verification checks and replication boundaries.

References

Alexia Anastasopoulos, Nikola Gradojevic, Fred Liu, Alex Maynard, and Ilias Tsiakas. Order flow and cryptocurrency returns. *Journal of Financial Markets*, 79:101047, 2026. doi: 10.1016/j.finmar.2026.101047.

Svetlana Bryzgalova, Markus Pelger, and Jason Zhu. Forest through the trees: Building cross-sections of stock returns. *The Journal of Finance*, 80(5):2447–2506, 2025. doi: 10.1111/jofi.13477.

Binqi Chen, Hongjun Ding, Ning Shen, Taian Guo, Jinsheng Huang, Luchen Liu, and Ming Zhang. AlphaSAGE: Structure-aware alpha mining via GFlowNets for robust exploration. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/06115011841a8d500050261c0ac4b1e4-Paper-Conference.pdf.

Eugene F. Fama and Kenneth R. French. A five-factor asset pricing model. *Journal of Financial Economics*, 116(1):1–22, 2015. doi: 10.1016/j.jfineco.2014.10.010.

Wayne E. Ferson and Campbell R. Harvey. Conditioning variables and the cross section of stock returns. *The Journal of Finance*, 54(4):1325–1360, 1999. doi: 10.1111/0022-1082.00148.

Panos Fousekis and Dimitra Tzaferi. Returns and volume: Frequency connectedness in cryptocurrency markets. *Economic Modelling*, 95:13–20, 2021. doi: 10.1016/j.econmod.2020.11.013.

Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273, 2020. doi: 10.1093/rfs/hhaa009. URL https://www.nber.org/papers/w25398.

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

Zachary Izzo, Ruishan Liu, and James Zou. Data-driven subgroup identification for linear regression. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14531–14552, 2023. URL https://proceedings.mlr.press/v202/izzo23a.html.

Zura Kakushadze. 101 formulaic alphas. *Wilmott*, 2016(84):72–81, 2016. doi: 10.1002/wilm.10525. URL https://arxiv.org/abs/1601.00991.

Bryan T. Kelly, Seth Pruitt, and Yinan Su. Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics*, 134(3):501–524, 2019. doi: 10.1016/j.jfineco.2019.05.001.

Yukun Liu, Aleh Tsyvinski, and Xi Wu. Common risk factors in cryptocurrency. *The Journal of Finance*, 77 (2):1133–1177, 2022. doi: 10.1111/jofi.13119.

Robert Novy-Marx. Fundamentally, momentum is fundamental momentum. Working Paper 20984, National Bureau of Economic Research, 2015. URL https://www.nber.org/papers/w20984.

Brenden K. Petersen, Mikel Landajuela, T. Nathan Mundhenk, Cláudio Prata Santiago, Sookyung Kim, and Joanne Taery Kim. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/1912.04871.

Andrea Pugnana and Salvatore Ruggieri. AUC-based selective classification. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 2494–2514, 2023. URL https://proceedings.mlr.press/v206/pugnana23a.html.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 2019. doi: 10.1038/s42256-019-0048-x. URL https://arxiv.org/abs/1811.10154.

Hao Shi, Weili Song, Xinting Zhang, Jiahe Shi, Cuicui Luo, Xiang Ao, Hamid Arian, and Luis Angel Seco. AlphaForge: A framework to mine and dynamically combine formulaic alpha factors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12524–12532, 2025. doi: 10.1609/aaai.v39i12.33365. URL https://ojs.aaai.org/index.php/AAAI/article/view/33365.

Esmeralda S. Whitammer, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in GFlowNets. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-0431. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/27b51baca8377a0cf109f6ecc15a0f70-Abstract.html.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696.

Shuo Yu, Hongyan Xue, Xiang Ao, Feiyang Pan, Jia He, Dandan Tu, and Qing He. Generating synergistic formulaic alpha collections via reinforcement learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5476–5486, 2023. doi: 10.1145/3580305.3599831.

Hengzhe Zhang, Aimin Zhou, Hong Qian, and Hu Zhang. PS-Tree: A piecewise symbolic regression tree. *Swarm and Evolutionary Computation*, 71:101061, 2022. doi: 10.1016/j.swevo.2022.101061.

Tianping Zhang, Yuanqi Li, Yifei Jin, and Jian Li. AutoAlpha: an efficient hierarchical evolutionary algorithm for mining alpha factors in quantitative investment. *arXiv preprint arXiv:2002.08245*, 2020. URL https://arxiv.org/abs/2002.08245.

Zhoufan Zhu and Ke Zhu. AlphaQCM: Alpha discovery in finance with distributional reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 80463–80479, 2025. URL https://proceedings.mlr.press/v267/zhu25ag.html.

Appendix Guide

Appendix A details program construction and learning, B data and execution, C derivations, and D the comparison methods. Appendix E gives additional results in the order of the main text: complete-program performance, then Q1, Q2, and Q3. Appendix F reports exploratory variants, G representative programs, and H verification.

A Program Construction and Learning

A.1 Joint program construction

Let g_θ denote a symbolic financial or price–volume state and u_ϕ a second numerical expression. We write $p = C_c \wedge S_k$, where C_c tests a visibility stage and S_k an asset state; where no discrete reporting event exists, $C_c \equiv 1$. U.S. programs use

$$p(i,t) = C_c(i,t) \wedge S_k(g_\theta; i, t), \qquad s(i,t) = \Psi\big(g_\theta(i,t), u_\phi(i,t)\big).$$

C_c refers to the stage after a new annual report first became visible in the historical snapshot. S_k tests the high or low side of a state expression or of its monthly change. Ψ is a restricted fusion and smoothing operator. Cryptocurrency combines a price–volume parent predicate with an optional spot-trading state:

$$p_{\text{crypto}}(i,t) = S_k(g_\theta; i, t) \wedge B_b(Z_{i,t}).$$

Z includes taker-buy share, mean quote amount per trade, changes in trade count and quote volume, and a five-day taker-buy mean. The price–volume arm sets $B_b \equiv 1$; the spot-state arm extends the fixed menu. Both arms score with price–volume inputs.

Stages and templates are fixed strata outside the search. U.S. search enumerates four visibility stages and four state templates into 16 cells and learns expressions and fusion actions within each cell. The state-only track sets $C_c \equiv 1$ and is searched afresh; report age and changes may still enter numerical inputs. A variant in Appendix F instead learns a clock action. Once selected, $\theta, \phi, c, k, b, \Psi, d, H$ do not change in response to evaluation-period returns; new observations only cause the frozen program to recompute its state and score.

A.2 Visibility clock and numerical conventions

Let J_t indicate that a new fiscal-year fourth-quarter report first becomes visible in the current monthly decision interval. First visibility comes from historical Point Date records; copies and revisions of the same report do not retrigger the event. The four mutually exclusive stages are

$$J_t, \quad \neg J_t \wedge J_{t-1}, \quad \neg J_t \wedge \neg J_{t-1} \wedge J_{t-2}, \quad \neg J_t \wedge \neg J_{t-1} \wedge \neg J_{t-2}.$$

State templates test the high or low side of either the state level or its change since the previous monthly decision. Internal state ranks are computed on the original eligible cross-section before predicate filtering; correlations are reranked on the support. U.S. internal ranks use stable ordinal ties normalized to $[0, 1]$; cryptocurrency uses average tied rank divided by $n + 1$, minus $1/2$. Reported correlations use average ties.

A.3 Fields and operators

g and u are numerical expressions, p is Boolean, and s (or $\alpha = d\,s$) is used for ranking. Internal multiplications and sign switches belong to score computation; only the outer WHEN predicate decides where the program applies.

Table 5: U.S. fields and operators. X^{prev} is the previous visible report value; a_t is calendar age since the current report first became visible.

Type	Definition or candidate set	Execution convention
Base inputs	Fundamentals; prices and volumes O, C, H, L, V, W	Historical Snapshot; $W = (H + L + C)/3$
Scale B	Assets AT, sales SALE, book equity BE, market equity ME	ME uses shares and fiscal-year-end price
Protected division D_{U}	$D_{\mathrm{U}}(x, y) = x/y$ for $\lvert y\rvert > 10^{-6}$	Otherwise undefined
Fundamental views	$D_{\mathrm{U}}(X, B)$; $D_{\mathrm{U}}(X - X^{\mathrm{prev}}, B)$; $a_t[R_t(D_{\mathrm{U}}(X, B)) - .5]$	Level, report difference, age–rank interaction
Binary operators	$x + y,\ x - y,\ xy,\ D_{\mathrm{U}}(x, y)$	Combine state and price–volume subexpressions, then rank
Cross-sectional rank R	Stable ascending zero-based rank divided by $n - 1$	Ties follow fixed asset order; one finite value maps to zero
Time transforms	$R_t(X),\ X_t/X_{t-w} - 1,\ \mathrm{MA}_w(X),\ \mathrm{SD}_w(X),\ X_t - X_{t-w}$	$w \in \{5, 10, 20, 63\}$; protected division for returns
Fusion $\Phi(g, u)$	$g + u,\ u - g,\ gu,\ u - \lvert g - .5\rvert,\ u\,\mathrm{sgn}(g - .5),\ \mathrm{If}_{g \geq .5}(u, -u)$	If selects the first branch when true
Fusion order	For $v = \Phi(g, u)$, use v, $\mathrm{MA}_5(v)$, $\Phi(g, R(\mathrm{MA}_5(v)))$, or $\Phi(g, R(\mathrm{MA}_{10}(v)))$	Part of expression semantics
Output transform	$R(q),\ \lvert q\rvert,\ -q,\ \mathrm{MA}_5(q),\ \mathrm{MA}_{20}(q)$	Multiply by fixed source direction d
Predicate	Four stages; high/low side of g or $R(\Delta_{\mathrm{month}} g)$	Conjoin stage and state

In U.S. examples, $\Delta_w X = X_t - X_{t-w}$ and $\mathrm{Ret}_w(X) = D_{\mathrm{U}}(X_t, X_{t-w}) - 1$. Rolling means and population standard deviations use finite entries after warmup; an all-missing window remains undefined. Book equity follows the original input fallback and adjustments for deferred taxes and preferred stock. In the examples, D denotes D_{U}.

Table 6: Cryptocurrency fields and operators. Time windows use calendar days; Q is USDT quote volume and N trade count.

Type	Definition or candidate set	Execution convention
Price–volume leaves	O, C, H, L, V, W	$W = Q/V$; four leaf positions
Transformed leaves	$\mathrm{MA}_w(X), \mathrm{SD}_w(X), \Delta_w X, \mathrm{Ret}_w(X), \mathrm{Z}_w(X)$	$w \in \{5, 10, 20, 40, 63\}$; 150 choices
Difference/return	$\Delta_w X = X_t - X_{t-w}$; $\mathrm{Ret}_w(X) = X_t/X_{t-w} - 1$	Require $w + 1$ consecutive observations
Standardization	$\mathrm{Z}_w(X) = (X_t - \mathrm{MA}_w(X)_t)/\,\mathrm{SD}_w(X)_t$	Full window, sample SD; zero SD undefined
Cross-sectional rank r	Average tied rank divided by $n + 1$, minus $1/2$	At leaves and binary outputs
Binary operators	$x + y,\ x - y,\ xy,\ D_{\mathrm{C}}(x, y) = x/(\lvert y\rvert + .05)$	Form g, u, then s
Parent predicates	True, $g \geq 0$, $g < 0$, $r(\Delta_{20} g) \geq 0$, $r(\Delta_{20} g) < 0$	Change in rank versus 20 days earlier
Taker-buy share	Taker-buy USDT quote amount$/Q$	Optional high/low side
Trade size	Q/N	Mean quote amount per trade
Trade-count change	$N_t/\,\mathrm{MA}_{20}(N)_{t-1} - 1$	Prior 20-day mean denominator

Type	Definition or candidate set	Execution convention
Quote-volume change	$Q_t / \mathrm{MA}_{20}(Q)_{t-1} - 1$	Prior 20-day mean denominator
Smoothed buy share	MA_5(taker-buy share)	Five-day mean
Extended predicate	$S_k(g) \wedge [r(B) \geq 0]$ or $S_k(g) \wedge [r(B) < 0]$	Five parents, 50 extensions; fixed thresholds

Although O, C, H, L, V have the same meanings across markets, U.S. W is a price proxy whereas cryptocurrency $W = Q/V$. The 10-, 40-, and 63-day windows are feature lookbacks, not additional forecast horizons. Spot-trading fields enter cryptocurrency predicates only; ranking leaves remain price–volume inputs.

A.4 Search budgets and feedback

U.S. candidates require at least 30 finite assets per valid cross-section and at least 12 valid months for reward estimation. Support failures, degenerate scores, and non-executable candidates remain recorded. Each horizon and seed has 16 cells, 167 steps per cell, four candidates per batch, and at most 64 selected programs (four per cell). Horizons are discovered separately; their equal mix is an execution layer, not another trained model.

Cryptocurrency candidates use four transformed price–volume leaves, two subexpressions, and a final binary combination. The direction is the sign of the training full-parent mean correlation, fixed before within-predicate evaluation. Let q_f be mean directed conditional RankIC on valid dates, c_f asset–date coverage, and v_f the fraction of dates on which both the predicate and the parent cross-section are valid. The reward is

$$R_{\mathrm{crypto}}(f) = \begin{cases} q_f, & c_f \geq 0.35,\ v_f \geq 0.60,\ q_f \text{ finite}, \\ -1, & \text{otherwise.} \end{cases}$$

No extra increment reward is used. Search samples 256 batches of 32 candidates (8,192 in all), retains at most 64 distinct qualified training candidates, and then selects up to eight on 2023 validation under the fixed rule. Direction and pool are frozen before the evaluation window.

A.5 Generator updates

The U.S. sequential network samples an action sequence a within cell (c, k) and uses

$$\mathcal{L}_{\mathrm{US}} = -\frac{1}{B} \sum_b (R_b - \bar{R}) \log \pi_\eta(a_b \mid c, k) - 0.0008\, \overline{\mathcal{H}}.$$

$\bar{R}$ is the batch mean and $\mathcal{H}$ the sum of action-distribution entropies. Sampling mixes in 15% uniform actions. Hidden dimension is 96 and the Adam learning rate is 0.002.

The cryptocurrency sequential network uses 128-dimensional state and action embeddings. Fixed action positions give every sequence a unique reverse path. Its trajectory-balance loss is

$$\mathcal{L}_{\mathrm{C}} = \frac{1}{B} \sum_b \left[\log Z + \log \pi_\eta(a_b) - 20\ \mathrm{clip}(R_b, -1, 1.5)\right]^2 - 0.01\, \overline{\mathcal{H}},$$

so the positive terminal reward is $\exp\{20\,\mathrm{clip}(R_b, -1, 1.5)\}$. The Adam learning rate is 10^{-4} and the gradient norm is capped at one. AlphaSAGE's RGCN encoder and its structural-alignment and novelty objectives belong to its external implementation; neither of our generators is retrained at execution time.

A.6 Pseudocode and component provenance

The construction–evaluation–update loop in Figure 2 is inspired by expression search, including AlphaSAGE's tree actions and its feedback from predictive quality, structural alignment, and novelty

(Chen et al., 2026). Our candidate unit extends that loop to a complete conditional program. Below, [O] marks our program design and market-specific implementation, [E] a general learning mechanism, and [S] an original AlphaSAGE component. Batch-baseline policy gradient and trajectory balance are established mechanisms (Williams, 1992; Whitammer et al., 2022). Within-cell retention uses, in sequence, conditional quality, quality relative to the paired price–volume reference, subperiod stability, and coverage-weighted contribution; previously selected programs and their two parents are excluded. Undefined metrics rank last. Zero or undefined source correlation produces zero exposure. Missing labels and zeroed training feedback are recorded separately.

At every new decision time, p_f and s_f are recomputed from then-visible inputs; an active program ranks $d_f s_f$ on its own support and an inactive program emits no signal.

Table 7: Component provenance. [S] is original AlphaSAGE; [O] is our program or market implementation; [E] is a general learning mechanism.

Component	AlphaSAGE [S]	Our U.S. instance	Our crypto instance
Search object	Expression tree and alpha pool	[O] (p, s) with shared g	[O] (p, s) from formula and menu actions
Policy	RGCN tree encoder and GFN	[O] predicate-cell embeddings, sequential heads	[O] fixed-position action embeddings, sequential heads
Feedback	R_{IC}, R_{SA}, R_{NOV}	[O] absolute within-support quality and support	[O] directed within-support quality and coverage thresholds
Update	Trajectory balance and entropy	[E] batch-baseline policy gradient	[E] trajectory balance and entropy
Use	Alpha pool and dynamic linear combination	[O] per-formula predicate, source direction, fixed combination	[O] training shortlist, validation selection, per-formula predicate

Input: visible source panel, matured labels, horizon H, 16 cells, grammar, and a 64-program pool rule.
Output: frozen $\mathcal{F} = \{(p, s, d, H)\}$ and the full candidate ledger.
1. [O] Initialize a conditioned sequential policy in each prespecified cell (c, k).
2. **for** 167 batches per cell: [E] mix 15% uniform legal actions; sample four sequences.
3. [O] Decode g, u and fusion actions; construct $p = C_c \wedge S_k(g)$ and $s = \Psi(g, u)$ with the same g.
4. [O] Execute p by month; evaluate s only on assets with valid membership, label, and score. Record signed conditional Spearman, valid months, and support size.
5. [O] Compute $R = \kappa|\bar{\rho}|$; undefined quality gives zero training feedback but is recorded. [E] Update by $R_b - \bar{R}$ and $\mathcal{L}_{\mathrm{US}}$.
6. [O] Save complete actions, pairs, source quality, and rejection reasons. Deduplicate and select at most four per cell under fixed source rules.
7. [O] Set d from each program's conditional source correlation; freeze (p, s, d, H), recording unfilled positions.

Algorithm 2: U.S. discovery within fixed visibility-stage and state-template cells.

Input: visible training/validation panels, matured labels, H, grammar, parent-predicate and spot-state menus.
Output: at most eight frozen programs and candidate ledgers.
1. [O] Initialize the sequential policy; [E] initialize a learnable $\log Z$.
2. **for** 256 batches: [E] sample 32 complete action sequences with log probabilities and entropy.
3. [O] Decode score and parent predicate; for the spot-state arm, add a spot-state action to form p. Fix the source direction d on training parent support, then evaluate $d\,s$ on p.
4. [O] Compute q, coverage, and paired-valid-date fraction. Set $R = q$ when $c \geq 0.35$, $v \geq 0.60$, and q is finite; otherwise $R = -1$.
5. [E] Set $\log \mathcal{R} = 20\,\mathrm{clip}(R, -1, 1.5)$; update the policy and $\log Z$ by $\mathcal{L}_{\mathrm{C}}$ with entropy and gradient clipping.
6. [O] Deduplicate, preserve pairs and directions, and retain at most 64 qualified training candidates. Evaluate them on validation and select at most eight. Freeze and return, recording unfilled positions.

Algorithm 3: Cryptocurrency complete-program sampling and trajectory-balance update.

B Data and Execution Protocol

B.1 Information time, universe, and events

Fundamentals enter through historical Snapshot Point Date versions at first visibility; later revisions never backfill past information. The annual clock detects a new fourth-quarter report period, not repeated snapshots. State changes require the necessary then-known history. Cryptocurrency selects the top 50 symbols each month by trailing 30-day mean USDT quote volume, subject to at least 28 valid trading days and 90 days of history. After first admission, an asset's identity is retained; 26 events and 14 exchange notices handle exits and migrations. STRAX's same-name replacement starts a new numerical history. Notice coverage during training and validation is limited to what was acquired for the study.

B.2 Direction, pool budget, and seeds

The two U.S. tracks and AlphaSAGE each have three seeds, two horizons, and 64 programs per pool: 18 pools and 1,152 programs, of which the two tracks contribute 768. Seed accounts run separately; the 5/20-day mix combines targets within each seed before costs and then aggregates the three seed accounts. All original directions, unfilled positions, and anomalies remain recorded. Cryptocurrency retains eight programs per seed and horizon; each seed compounds its own account, and the three are combined with equal initial capital. External methods keep their own grammar, pool, and search budget under the shared data and execution protocol.

B.3 Signal formation, budget, and costs

On each active $\mathcal{A}_t(p)$, the top and bottom 20% of α_f receive equal-sized long and short budgets, producing weights $w_{f,t}$. For pool capacity K_H,

$$z_t^{(H)} = \frac{1}{K_H} \sum_{f:H_f=H} w_{f,t}, \qquad z_t^{(5,20)} = \tfrac{1}{2} z_t^{(5)} + \tfrac{1}{2} z_t^{(20)}.$$

The budget of an inactive or invalid program stays in cash; other programs do not inherit it. Opposing positions net before trading costs. U.S. signals form at month end and rebalance at the next close. Cryptocurrency uses daily signals, enters at the $t+2$ open, and holds H staggered accounts. Trading costs are 5 or 10 basis points of traded notional; short positions also incur a 3% annualized borrowing cost. These are executed-account rules, distinct from formula-level RankIC diagnostics.

B.4 Missing prices and valuation

Unknown held returns remain missing. During a suspension or absent quote, the last available value is used for valuation only and cannot be traded. Explicit cumulative-return adjustments are applied once when pricing resumes, while terminal returns and cash distributions follow recorded event

Table 8: Data, splits, and program pools. Each seed discovers its own pool.

Item	U.S. equities	Cryptocurrency spot
Inputs	Historical fundamental snapshots and CRSP prices/volumes	Spot prices/volumes and trading structure
Time predicate	Four stages defined by first visibility of a new annual report	$C \equiv 1$; no disclosure gate
Asset state	Financial expression and monthly change	Price–volume expression and spot-trading state
Training	2005–2022 (2004 warmup)	June 2020–2022
Validation	2023, frozen-pool diagnosis	2023, fixed-rule program selection
Evaluation	2024–2025	2024; preregistered 2025 extension
Pool size	64 programs per seed and horizon	8 programs per seed and horizon
Aggregation	Three seeds equally weighted; equal 5/20-day mix	Three seeds equally weighted; horizons reported separately

dates. Unverifiable reorganization proceeds remain outstanding valuation balances rather than received cash. The reproducibility materials retain event identities, terminal-price proxies, unresolved items, and daily exposures. Removing 15 endpoint price proxies gives full-cycle returns of 15.47%, 4.55%, and 0.02% for clock-plus-state, state-only, and AlphaSAGE, preserving their order. The main reset 2024–2025 account applies only the three valuation adjustments fixed before evaluation.

C Mathematical Derivations

C.1 Optimal ordering on a fixed set

Let $n = |\mathcal{A}_t|$, r_i be the predicted rank, Q_i the future-return rank, and $m_i = \mathbb{E}[Q_i \mid \mathcal{F}_t]$. With no ties, both rank lists have mean $(n+1)/2$ and population variance $(n^2-1)/12$. Therefore

$$\mathbb{E}[\rho(r, Q) \mid \mathcal{F}_t] = \frac{12}{n(n^2-1)} \sum_i \left(r_i - \frac{n+1}{2}\right)\left(m_i - \frac{n+1}{2}\right).$$

If $m_i > m_j$ but $r_i < r_j$, swapping their predicted positions raises the unnormalized inner product by $(r_j - r_i)(m_i - m_j) > 0$. Eliminating inversions proves Proposition 1. The three-asset construction in Figure 1 changes only the comparison set, keeping the joint future-return distribution fixed.

C.2 Within-support ranking loss

On one support, standardize score ranks and outcome ranks to zero mean and unit variance, U_i, V_i. The directed squared rank loss then satisfies

$$\ell_t(f, d) = \frac{1}{2n_t} \sum_{i \in \mathcal{A}_t} (dU_i - V_i)^2 = 1 - d\rho_t(f), \qquad n_t = |\mathcal{A}_t|.$$

Both coordinate systems depend on the complete support: removing another asset can change the loss coordinates of retained assets. For the U.S. reward in Section 3.2,

$$\boxed{1 - R(f) = (1 - \kappa_f) + \kappa_f \min_{d \in \{-1,+1\}} \overline{\ell}(f, d).}$$

Small support shrinks the objective toward the uninformative loss of one; adequate support gives observed rank quality more weight. Since $\min_d \bar{\ell} = 1 - |\bar{\rho}|$, the expression follows directly. With $\rho_t^+ = \max(\rho_t, 0)$ and $\rho_t^- = \max(-\rho_t, 0)$,

$$|\bar{\rho}| = \overline{|\rho|} - 2\min(\overline{\rho^+}, \overline{\rho^-}),$$

so the objective penalizes sign cancellation across months without a separate variance or downside-risk term. Direction is fixed once in the source period and never flipped month by month. Policy-gradient rewards are not differentiated through asset ranks; a negative advantage simply means a candidate reward below the batch mean.

C.3 Two sources of conditional ranking difference

Let U, V be standardized score and outcome ranks on the full reference set and U_A, V_A the ranks recomputed within support A. Then

$$\rho_A - \rho_{\mathcal{U}} = \underbrace{\mathbb{E}[UV \mid A] - \mathbb{E}[UV]}_{\text{selection under the original reference}} + \underbrace{\mathbb{E}[U_A V_A \mid A] - \mathbb{E}[UV \mid A]}_{\text{reranking, recentering, and rescaling}}.$$

For scores $(1, 2, 3, 4, 5, 6, 7, 8)$ and outcomes $(5, 6, 7, 8, 1, 2, 3, 4)$, full-set RankIC is $-11/21$. Restricting to the first four assets leaves the mean original-rank product at $-11/21$ but yields a within-set RankIC of 1. Ranking differences, asset contributions, and portfolio returns therefore need separate definitions.

C.4 Same-count random reference

For a same-stage comparable set C and support $S \subset C$, define the pairwise concordance of the directed score $f = ds$ by

$$J(S) = \binom{|S|}{2}^{-1} \sum_{i<j\in S} \operatorname{sign}(f_i - f_j)\operatorname{sign}(y_i - y_j),$$

with $|S| \geq 2$, finite scores and labels, and tied pairs contributing zero. A uniformly sampled subset B of C of the same size contains each pair with equal probability, hence

$$\mathbb{E}_B[J(B)] = J(C), \qquad M_{\text{random}} = J(S) - J(C).$$

This is the exact same-count random reference for *pairwise concordance*. Empirical Spearman and executed-return controls instead evaluate their own fixed random-support samples; low- and high-score tails give further prespecified references. Choosing, with the month's observed labels, whichever reference is most favorable would form a retrospective stress test, not a prospective objective.

D Comparison Methods

D.1 Search baselines

The U.S. AlphaSAGE adapter is a trajectory-balance search on the common typed representation and samples 10,752 candidates per seed and horizon; each of our U.S. tracks samples 10,688. Both keep 64-program pools selected by the shared source rule. GP and PPO use 64-formula U.S. pools and eight-formula cryptocurrency pools under the same market inputs and execution. In cryptocurrency, AlphaSAGE uses its official graph policy: AlphaSAGE-8 selects eight formulas under our train/validation rule with an equal budget, and AlphaSAGE-50 keeps its original 50 expressions and weights. The two identities are never interchanged.

Dynamic missing values for AlphaSAGE-50. Within the valid range of each expression, scores are normalized across then-eligible assets; an undefined expression contributes a neutral zero, while non-members and assets with no valid expression are excluded. The dynamic membership mask is retained. This adapter was checked on pre-2024 data before use in 2024.

D.2 PS-Tree

PS-Tree (Zhang et al., 2022) refits a single predictor per market, horizon, and seed on the current inputs, with feature standardization and output inverse transformation restored: 202 tabular U.S. features and 70 cryptocurrency features. Training uses only the market's source period, at most 400 assets with matured labels per decision date, 400 individuals, 40 generations, at most four leaves, expression height at most six, and minimum leaf fraction 0.1. Its target is the cross-sectional future-return rank, and the source-period mean daily Spearman fixes its direction. One whole predictor forms one account under our execution and cost rules; it is never split into a factor pool. On the U.S. 5/20-day mix at 5 bp, PS-Tree returns -5.10% and GP -4.99%. Table 9 contrasts the two predictive objects.

Table 9: How PS-Tree and conditional factors organize prediction and execution.

	PS-Tree	Conditional factors (ours)
Predictive object	Adaptive partitions and local symbolic experts form one predictor	Each (p_f, s_f, d_f, H_f) executes independently within a factor pool
Learning feedback	Local regression and partition updates use predictive performance	Complete programs receive within-support ranking quality and support feedback
Execution	Soft memberships combine local experts	Predicate evaluated per formula; supports may overlap; inactive formulas emit no signal

E Additional Results

E.1 Complete-program performance

Table 10 adds risk and cost sensitivity to the returns in Table 2. At 10 bp, the 20-day cryptocurrency ordering is unchanged, whereas 5-day returns fall sharply for every method, and the 5-day spot-state return falls to about zero. AlphaSAGE-50, which keeps its original pool, loses money at both horizons. Table 11 shows per-seed returns: the 20-day cryptocurrency programs are positive and above AlphaSAGE-8 for every seed.

Table 10: Risk and cost sensitivity of complete programs (three equal-initial-capital seed accounts; 3% annualized borrowing cost). Maximum drawdown (MDD) is a magnitude. Panel A: U.S. equities, 2024–2025, 5-bp one-way cost; Sharpe is the mean of seed-level zero-benchmark daily Sharpe ratios annualized with 252 trading days. Panel B: cryptocurrency spot, 2024; Sharpe is computed on the three-seed mean daily account with a zero benchmark and 365 calendar days, while IC and RankIC use final target positions. GP and PPO use the same market-specific accounting and metric functions as the corresponding conditional programs. AlphaSAGE-50 retains its original pool and weights (Appendix D).

A. U.S. equities, 5 bp

	5-day			20-day		
Method	Return	Sharpe	MDD	Return	Sharpe	MDD
Clock + state	−0.39%	−0.12	2.01%	−2.51%	−0.59	5.82%
State only (retrained)	−2.88%	−0.65	4.74%	−8.61%	−0.60	16.44%
AlphaSAGE (adapter)	−2.69%	−0.48	5.16%	−7.42%	−0.64	13.38%
GP (adapter)	−2.63%	−0.44	6.03%	−7.83%	−0.64	13.67%
PPO (adapter)	−2.67%	−0.47	5.49%	−7.08%	−0.58	13.51%

B. Cryptocurrency spot

			5 bp			10 bp		
Method	IC	RankIC	Return	Sharpe	MDD	Return	Sharpe	MDD
5-day								
Price–volume condition	0.019	0.063	5.24%	0.47	8.86%	1.67%	0.19	9.08%
Spot-state condition	0.017	0.061	3.84%	0.38	8.19%	−0.04%	0.05	8.88%
AlphaSAGE-8	0.015	0.061	−1.46%	−0.16	8.28%	−3.39%	−0.42	8.93%
AlphaSAGE-50	−0.000	−0.002	−9.17%	−1.33	9.78%	−12.18%	−1.80	12.64%
GP (adapter)	—	—	6.31%	0.53	7.44%	2.84%	0.28	8.31%
PPO (adapter)	—	—	4.66%	0.41	6.81%	1.45%	0.17	7.77%
20-day								
Price–volume condition	0.037	0.087	21.11%	1.68	5.03%	20.10%	1.61	5.15%
Spot-state condition	0.043	0.094	24.37%	2.06	4.09%	23.32%	1.98	4.10%
AlphaSAGE-8	0.041	0.092	13.62%	1.55	4.30%	13.01%	1.49	4.37%
AlphaSAGE-50	0.005	0.005	−6.42%	−0.84	10.57%	−7.55%	−1.00	11.45%
GP (adapter)	—	—	16.93%	1.26	6.63%	15.79%	1.18	6.86%
PPO (adapter)	—	—	17.42%	1.27	5.94%	16.23%	1.19	5.96%

Table 11: Cumulative cryptocurrency net return by training seed, 2024, 5-bp cost.

Horizon	Method	seed 0	seed 1	seed 2
5-day	Ours: Price–volume condition	11.31%	4.30%	0.10%
5-day	Ours: Spot-state condition	2.08%	0.68%	8.76%
5-day	AlphaSAGE-8	0.02%	−0.71%	−3.70%
5-day	AlphaSAGE-50	−22.02%	−13.99%	8.48%
20-day	Ours: Price–volume condition	22.65%	24.90%	15.77%
20-day	Ours: Spot-state condition	21.86%	21.71%	29.53%
20-day	AlphaSAGE-8	13.26%	16.42%	11.20%
20-day	AlphaSAGE-50	−13.39%	−22.41%	16.53%

Figure 5 shows the corresponding return paths. In the separately reset U.S. account, the 5/20-day mix ends at -1.13%, -5.43%, and -4.79% for clock-plus-state, state-only, and AlphaSAGE at 5 bp, with three-seed ranges $[-1.61, -0.65]\%$, $[-5.83, -5.05]\%$, and $[-4.96, -4.68]\%$; at 10 bp the returns are -1.57%, -5.91%, and -5.13%. Slicing the continuous history (panel c) instead gives -1.03%, -5.29%, and -4.61% in 2024–2025 at 5 bp, because the continuous account carries earlier positions and valuation balances.

E.2 Q1: formula selection

The selection study in Table 1 uses a frozen library of 192 U.S. predicates. For each predicate, the full-universe and within-support rules each pick one formula under the same source-period direction rule, and both choices are compared on the common executable support. All 24 portfolio accounts built from the selection arms complete in 2024–2025, and their three-seed mean returns are negative. The corresponding 24 accounts in 2023 are blocked by held-price observations outside the valuation rule fixed in advance, so neither partial returns nor zero-filled returns are reported.

E.3 Q2: fixed-formula applicability

Predicate components. Table 12 switches individual predicate components on the same formulas. U.S. gains come mainly from the visibility stage: clock-only returns -0.893% on the 5/20-day mix, compared with -1.130% for the full predicate. Adding the financial state raises paired formula-level RankIC over clock-only by 0.00931 and 0.00750 at 5 and 20 days (positive for all three seeds) but does not raise return. Filtering the price–volume parent formula by fundamentals raises the mixed return by 0.521 points (all three seeds positive); the corresponding fixed-formula increment of the state-only pool is 0.086 points. In cryptocurrency, the parent and spot-state components each help, and together give the highest 20-day return.

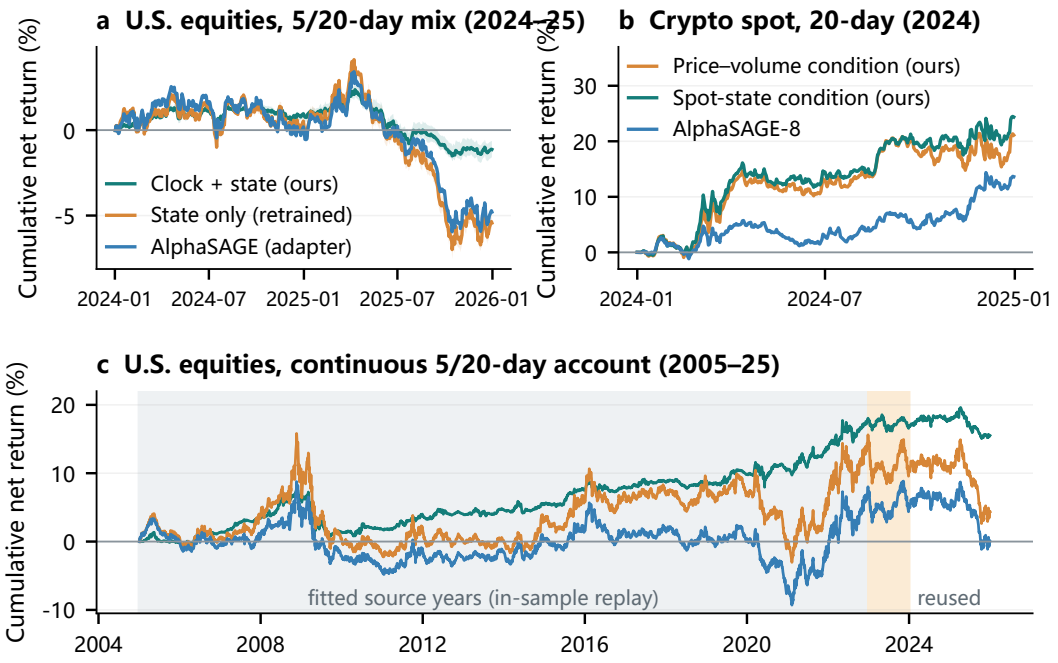


Figure 5: Cumulative net returns at 5-bp cost; each line combines three equal-initial-capital seed accounts. (a) U.S. 5/20-day mix in the reset 2024–2025 account, with shading for the seed range. (b) Cryptocurrency 20-day programs in 2024. (c) Continuous U.S. replay from 2005 with the colors of panel a; gray marks the fitted source years, pale orange the 2023 frozen-pool diagnosis, and the unshaded segment the 2024–2025 evaluation. Missing quotes and terminal valuation follow Appendix B.

Table 12: Conditional ablations of the same frozen formulas. Score, source direction, and budget are fixed within each market. U.S. arms use the clock-trained pool; cryptocurrency arms use the spot-state pool. Actual use size is retained; further size controls are separate.

Market / horizon	Predicate	Cumulative net return
U.S. equal-horizon	Always-on	−2.154%
U.S. equal-horizon	Financial state only	−2.056%
U.S. equal-horizon	Clock only	−0.893%
U.S. equal-horizon	Clock + financial state	−1.130%
Crypto 20-day	Always-on	15.414%
Crypto 20-day	Parent only	21.981%
Crypto 20-day	Spot state only	17.604%
Crypto 20-day	Parent + spot state	24.365%

Random support and long-only selection. Table 13 replaces each cryptocurrency support by same-count random assets. The full predicate exceeds every random repetition at 20 days (21.11% versus a 5.70% random mean for price–volume, 24.37% versus 14.98% for spot state) and also at 5 days. Table 14 keeps only the long side of each full-predicate account and compares it with an equal-weight universe under the same budget; all differences are positive for every seed.

Table 13: Same-count random asset support. Each formula and day retain the actual accepted count; three fixed random repetitions each aggregate the same three training seeds. Ranges describe support-selection variation on the observed market path.

Horizon	Formula pool	Full	Random mean	Random range
5-day	Price–volume condition	5.24%	−5.00%	−5.79% / −4.51%
5-day	Spot-state condition	3.84%	−4.47%	−6.32% / −2.78%
20-day	Price–volume condition	21.11%	5.70%	5.60% / 5.84%
20-day	Spot-state condition	24.37%	14.98%	14.44% / 15.29%

Table 14: Long-only selection check. Retain positive target positions of the original full-condition account; the reference equally weights the same budget over that day's eligible cryptocurrency universe. Both use the original execution schedule and costs.

Horizon	Pool	Long side	Equal-weight universe	Difference	Positive seeds
5-day	Price–volume condition	25.47%	20.67%	4.79%	3/3
20-day	Price–volume condition	37.70%	23.36%	14.34%	3/3
5-day	Spot-state condition	29.20%	20.97%	8.23%	3/3
20-day	Spot-state condition	36.61%	22.40%	14.21%	3/3

Uncertainty. Cryptocurrency fixed-formula increments use 4,000 common-date circular-block bootstrap draws with block lengths of 20, 40, and 60 days. With 40-day blocks, the 20-day added spot-state increment of 2.023 points has a descriptive interval of $[0.236, 4.130]$ points; price–volume intervals at 5 and 20 days are $[-1.416, 19.305]$ and $[-0.019, 21.779]$, and the 5-day added-state interval also crosses zero. Seeds share the market period and no multiple-comparison correction is applied.

Formula-level quality and coverage. Figure 6 compares within-support RankIC and executable coverage across methods. In the U.S. 2024–2025 evaluation, 188 of 192 full-predicate programs have at least one defined RankIC, and 2,340 of 4,608 program-months are valid. Median RankIC is $+0.0087$ and median executable coverage 3.72%. The AlphaSAGE, GP, and PPO adapters have median RankIC $+0.0093$, $+0.0093$, and $+0.0134$, each with 4,608 valid program-months and about 90% median coverage. For the same cryptocurrency formulas on the same dates, price–volume predicates raise RankIC from $0.0360/0.0468$ to $0.0632/0.0809$ at 5/20 days, and the added spot state contributes another $0.0085/0.0038$. U.S. full predicates also give positive paired ranking increments over always-on for every seed at both horizons. About 32 of the 64 U.S. clock-plus-state programs

per seed are active at a time; a cryptocurrency eight-program pool always has at least four assets available.

Full-cycle levels and coverage. Figure 7 follows the 192 frozen U.S. 20-day clock-plus-state programs from 2005, complementing the paired differences in Figure 4c. The paired increment over same-count random support is $+0.00127$ in 2023 (one seed positive), $+0.01519$ in 2024, and -0.00223 in 2025. The share of program-months with a defined paired RankIC declines from 55.1% in 2023 to 49.8% in 2025. The selected share of about 12.2% in panel b is measured against each score's own base support, whereas the 3.72% coverage above is measured against all eligible point-in-time members.

Ranking decomposition. Under the decomposition in Appendix C, the 20-day cryptocurrency price–volume ranking increment of 0.0341 comprises 0.0287 from selecting assets with stronger

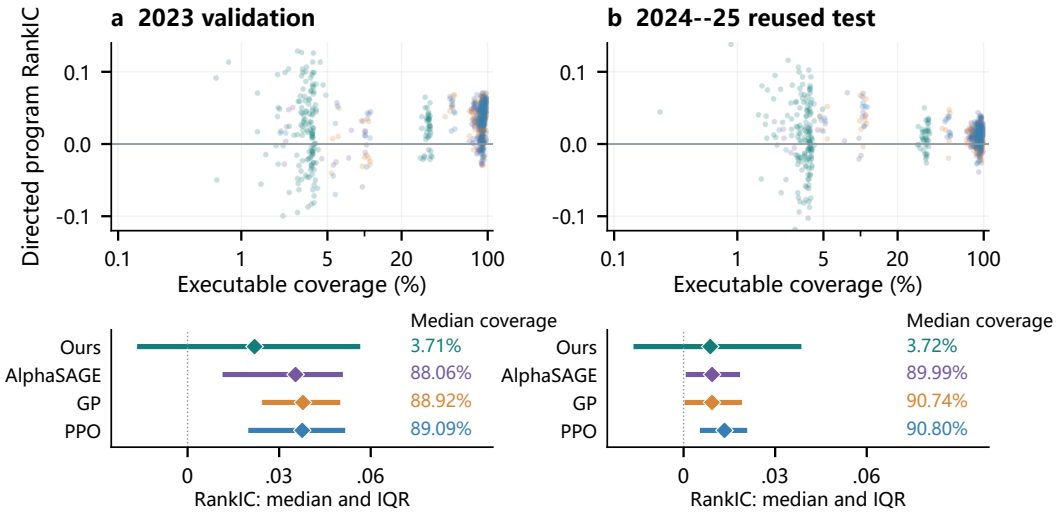


Figure 6: Formula-level directed RankIC and executable coverage in 2023 and the U.S. 2024–2025 evaluation. Every method keeps all 192 programs in the denominator; undefined RankIC is not replaced by zero. The lower row shows medians and interquartile ranges of defined values. Coverage is the share of eligible asset–decision opportunities on which a formula can execute, not portfolio exposure.

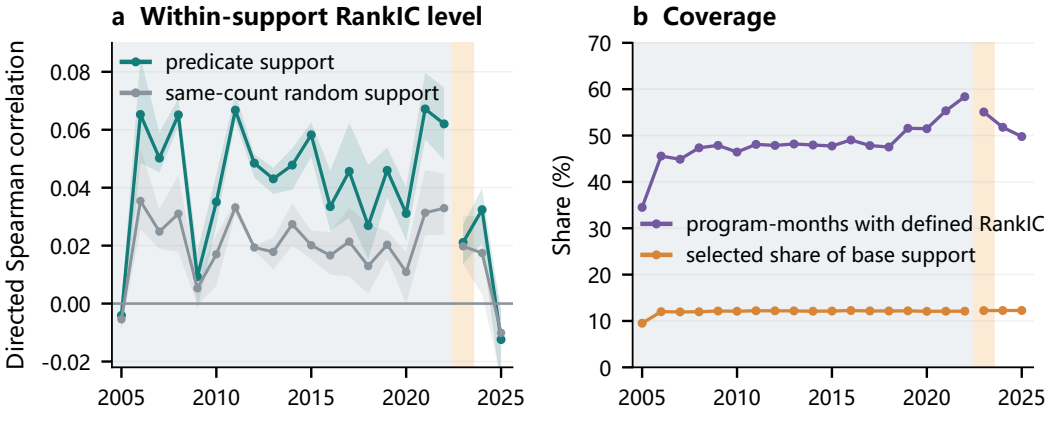


Figure 7: The same 192 U.S. programs over the full cycle. (a) Directed within-support Spearman correlation on the predicate support and on same-count random support. (b) Program-months with defined RankIC as a share of all program-months, and supported assets as a share of each score's base support. Annual means are formed within program, then equally across programs and seeds; shading spans seed means. Gray marks fitted source years and pale orange the 2023 diagnosis.

contributions under the original reference and 0.0054 from within-set reranking; the spot-state increment of 0.0244 over always-on splits into 0.0208 and 0.0036. Representative formulas, including negative examples, appear in Appendix G.

E.4 Q3: use decomposition and the 2025 extension

Table 3 groups each seed's daily conditional-minus-reference contributions by whether the reference return was positive, negative, or zero, then averages seeds equally; the groups sum to the cumulative difference before rounding. The U.S. full predicate and cryptocurrency price–volume predicates use their own always-on formulas as reference; the added spot state uses its parent predicate at matched capital deployment. Table 15 gives the asset-level view: within the always-on reference set, cryptocurrency predicates retain 47–53% of the positive contributions and exclude 43–47% of the negative contributions while keeping 48–52% of the weight.

Table 15: Positive and negative contributions in the always-on reference asset set. A fixed asset weight times its horizon return defines contribution. Positive retention and negative exclusion use only fully labeled reference sets, computed per seed before averaging.

Horizon	Pool	Positive retained	Negative excluded	Weight retained	Complete labels
5-day	Price–volume condition	51.00%	43.04%	50.41%	82.39%
20-day	Price–volume condition	47.13%	47.34%	47.72%	59.61%
5-day	Spot-state condition	51.79%	44.26%	50.89%	82.74%
20-day	Spot-state condition	52.78%	46.82%	52.01%	57.71%

In the 2025 extension (Table 4), the always-on returns of the same formulas are 0.99% and 1.84% for price–volume and 2.52% and -3.90% for spot state at 5 and 20 days; the same-count random references are 1.32%, 1.84%, 2.01%, and -3.74%. At 10 bp, AlphaSAGE-8 returns $+6.16\%$ and $+1.53\%$ at 5 and 20 days, still above the corresponding conditional programs. A 50-formula AlphaSAGE diagnostic on the finite overlap of its expressions follows a different missing-value rule and is not substituted for the common eight-formula comparison.

F Exploratory Variants and Structural Controls

F.1 Learned clock actions

The main method stratifies search by visibility stage and state template. A joint variant instead builds an interaction state from fundamentals and price–volume expressions, uses that state in both predicate and score, and learns a clock action among the four stages and no clock. It assigns 50 programs to each of four state templates, samples 10,688 complete candidates per seed, selects 200 programs by source-period conditional quality, and freezes individual directions. This implementation is included with the accompanying code.

In the 2024–2025 evaluation window at 5 bp, the variant returns -0.545%, above the main 20-day result of -2.513% (or -1.589% on the same two seeds), and its increment over its own always-on version is $+2.126$ points. Yet 386 of 400 selected formulas (96.5%) collapse to one stage: no new annual report in the current or previous decision interval, but one two intervals earlier. Six choose the current and eight the previous interval. Across 24 evaluation months, seeds 0 and 1 have four and two months with no active formulas. We keep the stage-stratified method as the main report and show both the variant's higher return and its clock concentration. Capacity (200 versus 64 programs), restart structure, and selection rule differ between the two systems.

F.2 Objective and structure comparison

On the same variant family, a four-cell study crosses two-stage versus joint discovery with the original conditional-quality objective versus a fixed same-clock random-reference increment. The learnable clock head is disabled in all four cells. Table 16 reports two-seed means in the 2024–2025 evaluation window at 5 bp. The two-stage original-quality increment is positive for both seeds; the

Table 16: Structure and objective controls at full budget. Full return is cumulative net return; the conditional increment is relative to each arm's own always-on pool.

Discovery	Objective	Full return	Conditional increment
Two-stage	Original conditional quality	$+0.170\%$	$+3.217$pp
Joint	Original conditional quality	-0.536%	$+1.225$pp
Two-stage	Random-reference increment	-2.287%	-2.298pp
Joint	Random-reference increment	-4.239%	-3.375pp

joint cell has opposite seed signs. Both random-reference cells have negative mean increments, and the two-stage original-quality return turns negative at 10 bp. An objective that improves a relative increment can still have weaker absolute quality and executed return, which is why we reward absolute within-support quality and report increments alongside absolute performance (Section 4).

F.3 Other objectives and predicate extensions

Figure 8 shows that coverage and pairwise utility are positively associated across development candidates. A tail-increment objective produces conditional increments of -1.05 and -1.20 points on the two seeds, and coverage-floor reselection returns -7.48%. A control that constructs predicate and score from separate states yields $+1.19$ points versus $+1.225$ for shared construction, without a clear advantage from sharing. A separate two-stage exploration with a different parent-expression count and menu-evaluation budget returns -1.08% versus -7.27% always-on ($+6.19$ points), below the -0.545% of the learned-clock variant in absolute return. Each result follows its own frozen protocol.

In the main cryptocurrency spot-state experiment, 13 selected 5-day and 10 selected 20-day programs use an added trading-state predicate; the others keep only their price–volume parent. The exposure-matched split measures the added-state effect and is positive in the combined return at both horizons.

G Representative Conditional Formulas

U.S. representatives are the lower medians of source-period training quality within seed and horizon for the clock track, chosen before evaluation summaries were read. The other complete programs, seed overlap, and distributions are available in the reproducibility materials. F denotes the financial-state rank, T the price–volume-parent rank, and A information age; B, Q abbreviate intermediate computations. Cryptocurrency examples keep one representative per seed, horizon, and predicate family. Each WHEN/USE description preserves the original predicate and direction.

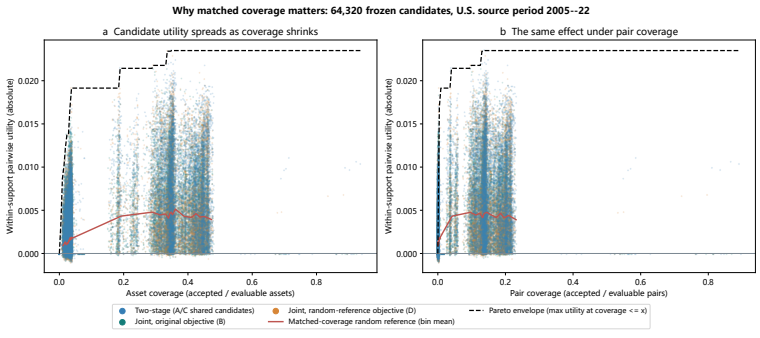


Figure 8: Coverage and pairwise utility for 64,320 frozen development candidates; their Spearman association is $+0.328$. Expressions, templates, and supports vary together, so the plot describes a joint distribution rather than a causal coverage effect.

G.1 U.S. clock + state: five-day forecast

Seed 0; program 10 **WHEN:** a new fiscal-year fourth-quarter report first becomes visible in the current decision interval and $F_t \geq 0.5$.
USE: Rank supported assets in descending order of $\alpha_t = Q_t$.

$$\begin{aligned}
F_t &= R((D(\Delta_{\text{report}}\text{EBITDA}, \text{SALE}) - D(\Delta_{\text{report}}\text{PI}, \text{SALE}))) , \\
T_t &= R((\text{VWAP} - \text{VWAP}_{t-5} + \text{CLOSE} - \text{CLOSE}_{t-20})) , \\
B_t &= T_t \ \text{sgn}(F_t - 0.5), \\
Q_t &= R(\text{MA}_{10}\,(B))_t \ \text{sgn}(F_t - 0.5).
\end{aligned}$$

Fields: CLOSE: close; EBITDA: earnings before interest, taxes, depreciation and amortization; PI: pretax income; VWAP: $(\text{HIGH} + \text{LOW} + \text{CLOSE})/3$ price proxy.

Seed 1; program 147 **WHEN:** the new report first became visible in the previous, but not current, interval and $R_t(F_t - F_{t-1\text{ month}}) \geq 0.5$.
USE: Rank supported assets in descending order of $\alpha_t = Q_t$.

$$\begin{aligned}
F_t &= R((A_t\,(R(D(\text{AT}, \text{AT})) - 0.5) \cdot A_t\,(R(D(\text{CSHO}, \text{AT})) - 0.5))) , \\
T_t &= R((R(\text{OPEN}) \cdot \text{MA}_{10}\,(\text{VOLUME}))) , \\
B_t &= T_t \ \text{sgn}(F_t - 0.5), \\
Q_t &= \text{MA}_5\,(B)_t .
\end{aligned}$$

Fields: AT: total assets; CSHO: shares outstanding; OPEN: opening price; VOLUME: trading volume.
Interpretation note: This formula contains an age term multiplied by a constant self-ratio. Stable rank ties can carry asset-order information, so we do not assign that term a financial high/low interpretation; it remains in the complete representative set.

Seed 2; program 295 **WHEN:** the new report first became visible two intervals ago, but not in the current or previous interval, and $R_t(F_t - F_{t-1\text{ month}}) < 0.5$.
USE: Rank supported assets in descending order of $\alpha_t = |Q_t|$.

$$\begin{aligned}
F_t &= R((D(\text{RECT}, \text{BE}) \cdot A_t\,(R(D(\text{PRCC}_\text{F}, \text{AT})) - 0.5))) , \\
T_t &= R((\text{LOW} - \text{LOW}_{t-20} + R(\text{VWAP}))) , \\
B_t &= F_t \cdot T_t, \\
Q_t &= \text{MA}_5\,(B)_t .
\end{aligned}$$

Fields: LOW: daily low; PRCC_F: historical fiscal-year-end price proxy; RECT: receivables; VWAP: price proxy.

G.2 U.S. clock + state: twenty-day forecast

Seed 0; program 385 **WHEN:** a new fiscal-year fourth-quarter report first becomes visible in the current interval and $R_t(F_t - F_{t-1\text{ month}}) \geq 0.5$.
USE: Rank supported assets in descending order of $\alpha_t = -Q_t$.

$$\begin{aligned}
F_t &= R((D(\Delta_{\text{report}}\text{TXDITC}, \text{SALE}) \cdot A_t\,(R(D(\text{OANCF}, \text{ME})) - 0.5))) , \\
T_t &= R(D(D(\text{LOW}, \text{LOW}_{t-63}) - 1, D(\text{HIGH}, \text{HIGH}_{t-10}) - 1)) , \\
B_t &= F_t \cdot T_t, \\
Q_t &= F_t \cdot R(\text{MA}_5\,(B))_t .
\end{aligned}$$

Fields: HIGH/LOW: daily high/low; OANCF: operating cash flow; TXDITC: deferred taxes and investment tax credit.

Seed 1; program 525 **WHEN:** a new fiscal-year fourth-quarter report first becomes visible in the current interval and $F_t < 0.5$.
USE: Rank supported assets in descending order of $\alpha_t = \mathrm{MA}_5(Q)_t$.

$$\begin{aligned} F_t &= R((A_t\,(R(D(\mathrm{DLC},\mathrm{ME})) - 0.5) + D(\mathrm{CHE},\mathrm{AT}))\,), \\ T_t &= R((D(\mathrm{VOLUME},\mathrm{VOLUME}_{t-63}) - 1 + \mathrm{SD}_{63}(\mathrm{HIGH}))\,), \\ B_t &= F_t \cdot T_t, \\ Q_t &= F_t \cdot R(\mathrm{MA}_{10}\,(B))_t\,. \end{aligned}$$

Fields: CHE: cash and short-term investments; DLC: debt in current liabilities; HIGH: daily high; VOLUME: trading volume.

Seed 2; program 681 **WHEN:** the new report first became visible two intervals ago, but not in the current or previous interval, and $F_t \geq 0.5$.
USE: Rank supported assets in descending order of $\alpha_t = \mathrm{MA}_5(Q)_t$.

$$\begin{aligned} F_t &= R(D(A_t\,(R(D(\mathrm{OANCF},\mathrm{SALE})) - 0.5)\,, D(\mathrm{PRCC_F},\mathrm{BE}))\,), \\ T_t &= R((\mathrm{SD}_{10}(\mathrm{CLOSE}) - \mathrm{MA}_{10}\,(\mathrm{VOLUME}))\,), \\ B_t &= \mathrm{If}_{F_t \geq 0.5}(T_t, -T_t), \\ Q_t &= B_t. \end{aligned}$$

Fields: CLOSE: close; OANCF: operating cash flow; PRCC_F: fiscal-year-end price proxy; VOLUME: trading volume.

G.3 Cryptocurrency price–volume condition: five-day forecast

Seed 0

$$\begin{aligned} g_t &= r_t(r_t(\mathrm{SD}_{10}(H)) + r_t(\mathrm{Z}_{63}(H))), \\ u_t &= r_t(r_t(\mathrm{Ret}_{10}(H)) + r_t(\mathrm{SD}_5(V))). \end{aligned}$$

WHEN: $g_t \geq 0$.
USE: Rank supported assets in descending order of $\alpha_t = -r_t(g_t + u_t)$.
Meaning: Shared price–volume state lies on the high side of the current cross-section.

Seed 1

$$\begin{aligned} g_t &= r_t(r_t(\mathrm{Ret}_5(W)) + r_t(\mathrm{MA}_{40}(C))), \\ u_t &= r_t(r_t(\mathrm{MA}_{40}(W)) - r_t(\mathrm{MA}_{20}(V))). \end{aligned}$$

WHEN: $g_t \geq 0$.
USE: Rank supported assets in descending order of $\alpha_t = -r_t(g_t - u_t)$.
Meaning: Shared price–volume state lies on the high side of the current cross-section.

Seed 2

$$\begin{aligned} g_t &= r_t(r_t(\mathrm{SD}_{63}(C)) - r_t(\mathrm{Z}_{63}(W))), \\ u_t &= r_t(r_t(\mathrm{SD}_{10}(H)) + r_t(\mathrm{Ret}_{20}(W))). \end{aligned}$$

WHEN: $g_t < 0$.
USE: Rank supported assets in descending order of $\alpha_t = r_t(g_t u_t)$.
Meaning: Shared price–volume state lies on the low side of the current cross-section.

G.4 Cryptocurrency price–volume condition: twenty-day forecast

Seed 0

$$\begin{aligned} g_t &= r_t(r_t(\mathrm{Z}_{63}(W)) + r_t(\mathrm{Z}_{40}(V))), \\ u_t &= r_t(r_t(\mathrm{MA}_{10}(W)) + r_t(\mathrm{SD}_{40}(V))). \end{aligned}$$

WHEN: $g_t \geq 0$.
USE: Rank supported assets in descending order of $\alpha_t = -r_t(g_t + u_t)$.
Meaning: Shared price–volume state lies on the high side of the current cross-section.

Seed 1

$$g_t = r_t(r_t(\mathrm{Ret}_{40}(H)) - r_t(\mathrm{SD}_5(C))),$$
$$u_t = r_t(r_t(\mathrm{SD}_{10}(O)) + r_t(\mathrm{Ret}_{40}(W))).$$

WHEN: $g_t \geq 0$.
USE: Rank supported assets in descending order of $\alpha_t = -r_t(g_t u_t)$.
Meaning: Shared price–volume state lies on the high side of the current cross-section.

Seed 2

$$g_t = r_t(r_t(\mathrm{SD}_{40}(L)) + r_t(\mathrm{SD}_{63}(V))),$$
$$u_t = r_t(r_t(\mathrm{Z}_{10}(C)) + r_t(\mathrm{SD}_{63}(V))).$$

WHEN: $r_t(\Delta_{20} g) \geq 0$.
USE: Rank supported assets in descending order of $\alpha_t = -r_t(g_t + u_t)$.
Meaning: The 20-day change in shared state lies on the high side of the current cross-section.

G.5 Cryptocurrency spot-state condition: five-day forecast

Seed 0

$$g_t = r_t(r_t(\mathrm{Ret}_{20}(H)) - r_t(\mathrm{MA}_5(C))),$$
$$u_t = r_t(r_t(\mathrm{Ret}_{10}(O)) + r_t(\mathrm{SD}_{40}(H))).$$

WHEN: $r_t(B^{\mathrm{count}}) \geq 0$.
USE: Rank supported assets in descending order of $\alpha_t = -r_t(g_t + u_t)$.
Meaning: Trade-count change versus the preceding 20-day mean lies on the high side, together with the recorded parent requirement.

Seed 1

$$g_t = r_t(r_t(\mathrm{MA}_{20}(L)) - r_t(\Delta_5 C)),$$
$$u_t = r_t D_{\mathrm{C}}(r_t(\mathrm{Ret}_{63}(W)), r_t(\mathrm{SD}_{63}(L))).$$

WHEN: $g_t < 0$.
USE: Rank supported assets in descending order of $\alpha_t = r_t(g_t u_t)$.
Meaning: Shared price–volume state lies on the low side of the current cross-section.

Seed 2

$$g_t = r_t(r_t(\mathrm{Ret}_{40}(C)) - r_t(\mathrm{MA}_{63}(L))),$$
$$u_t = r_t(r_t(\mathrm{SD}_{10}(O)) + r_t(\mathrm{SD}_{10}(O))).$$

WHEN: $g_t \geq 0$.
USE: Rank supported assets in descending order of $\alpha_t = -r_t(g_t + u_t)$.
Meaning: Shared price–volume state lies on the high side of the current cross-section.

G.6 Cryptocurrency spot-state condition: twenty-day forecast

Seed 0

$$g_t = r_t(r_t(\mathrm{MA}_5(W)) + r_t(\mathrm{Ret}_{63}(C))),$$
$$u_t = r_t(r_t(\mathrm{MA}_{20}(V)) + r_t(\mathrm{MA}_{63}(W))).$$

WHEN: $r_t(B^{\mathrm{count}}) < 0$.
USE: Rank supported assets in descending order of $\alpha_t = -r_t(g_t + u_t)$.
Meaning: Trade-count change versus the preceding 20-day mean lies on the low side, together with the recorded parent requirement.

Seed 1

$$g_t = r_t(r_t(\mathrm{MA}_{20}(V)) + r_t(\mathrm{SD}_{63}(H))),$$
$$u_t = r_t(r_t(\mathrm{SD}_{20}(V)) - r_t(\mathrm{SD}_{40}(O))).$$

WHEN: $r_t(B^{\mathrm{quote}}) < 0$.
USE: Rank supported assets in descending order of $\alpha_t = -r_t D_{\mathrm{C}}(g_t, u_t)$.
Meaning: Quote-volume change versus the preceding 20-day mean lies on the low side, together with the recorded parent requirement.

Seed 2

$$g_t = r_t(r_t(\mathrm{Ret}_{40}(C)) - r_t(\mathrm{MA}_{63}(L))),$$
$$u_t = r_t(r_t(\mathrm{SD}_{10}(O)) + r_t(\mathrm{SD}_{10}(O))).$$

WHEN: $g_t \geq 0$.
USE: Rank supported assets in descending order of $\alpha_t = -r_t(g_t + u_t)$.
Meaning: Shared price–volume state lies on the high side of the current cross-section.

H Verification and Reproducibility

H.1 Formula and account checks

For the 18 U.S. pools, source selection, directions, and 1,152 formula prefixes were checked against frozen records. Evaluation accounts were independently sampled through a scalar accounting path; the 5/20-day mix combines targets before costs. The price–volume-parent study reconstructed training metrics for 768 formulas; 254 parent directions differ from mixed-formula directions, and all are training-derived. A further 108 component accounts and 324 Spearman samples were checked. Internal tie-order sensitivity and account agreement are recorded separately.

For cryptocurrency, all 96 conditional programs and 300 AlphaSAGE expressions keep their seeds and directions. Checks cover 228 account paths, 6,588 paired-exposure formations, and 50,616 rank-decomposition entries. The representative WHEN/USE examples were matched on predicate sets, order, and terminal positions over 366 dates. These checks come from the same research workflow; implementation agreement is distinct from independent statistical confirmation.

H.2 Materials and replication boundaries

Reproduction materials separate data acquisition, historical universe membership, input fields, frozen programs, executed evaluation, and figure construction. Every reported number specifies its program population, direction, labels, dates, costs, and account aggregation. Licensed financial rows are kept separate from public code. Cryptocurrency materials include daily support, four same-formula arms, same-count random support, bidirectional exposure checks, within-predicate ranking decomposition, signed use, and equal-budget long-only comparisons. U.S. materials include Snapshot construction, both complete tracks, four-arm and state-source splits, parent-formula analyses, full-history paths, and event valuation. Full expressions and displayed abbreviations are checked for identity. The materials are in the anonymous repository given in the Reproducibility Statement.