FEEDBACK LOOPS INVALIDATE SIGNIFICANCE TESTS IN LLM ALPHA MINING

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM alpha mining, like other automated discovery systems, proposes candidate formulas and scores them on historical data. The scores are fed back to guide the next proposals. The system then presents its highest scoring formula as a discovery. It backs that claim with a multiple testing correction, which promises at most a 5% chance that the discovery is false. That promise assumes the candidates were chosen without looking at the data. A feedback loop breaks this assumption, because every new proposal depends on how earlier ones scored. As a result, the more these systems search, the less their discoveries can be trusted. Prior work established this risk in theory, but its remedies change how the search runs. We measure the problem in real LLM discovery loops and fix it without changing them. First, we prove that no correction computed from the chosen candidates can keep its promise for every feedback loop. For a natural class of loops, the chance of a false discovery tends to 100% as the search grows. Second, we measure the effect on stock market data shuffled so that no formula can predict returns. There, the strongest standard test for data snooping declares an LLM alpha miner's best formula significant 15% of the time. The Benjamini-Hochberg procedure does so 29% of the time. A search that never sees its scores, and the same loop with its scores shuffled, both stay near 5%. The feedback is therefore the cause. Four times the budget lifts the two rates to 28% and 70%, while both controls stay flat. The gap widens as the search grows. The same failure appears in symbolic regression, outside finance. Third, we give a fix. Counting candidates more carefully does not help. Instead, we rerun the unchanged search many times on shuffled data and set the significance threshold from those runs. This keeps the 5% promise for any searcher. On later market data the search never saw, 92% of the formulas it accepts remain significant. Its main cost is rerunning the search, and a simple statistical shortcut cuts the reruns needed from about 100 to about 20. Practitioners can therefore attach a trustworthy significance claim to any discovery loop, at modest cost and without redesigning it.

1 INTRODUCTION

LLM alpha mining searches for formulas that predict returns. A proposer emits candidate formulas, each is scored on a fixed panel of market data, the scores go back to the proposer, and the loop repeats. Program search for mathematical constructions (Romera-Paredes et al., 2024), equation discovery (Shojaee et al., 2025), automated research pipelines (Lu et al., 2024) and a growing line of alpha miners (Yu et al., 2023; Tang et al., 2025; Shi et al., 2025) all share this structure. The winner is then reported with a correction for the number of candidates tried (Bonferroni or Holm, a false-discovery-rate procedure (Benjamini & Hochberg, 1995; Benjamini & Yekutieli, 2001), or one of the bootstrap tests built for data snooping over many correlated strategies (White, 2000; Hansen, 2005)), and that correction carries a promise: at most a 5% chance that the reported discovery is spurious.

Every one of those corrections keeps that promise under one assumption: that the candidates were chosen without looking at the data. A feedback loop violates it, because what the proposer tries next is a function of scores already computed on the evaluation panel. The number of hypotheses tested is then not merely unknown, the concern of the factor-zoo literature (Harvey et al., 2016; Hou et al., 2020), but undefined, and the corrections fail in the direction that manufactures discoveries.

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.

That adaptive reuse voids such guarantees is established in theory (Dwork et al., 2015a;b; Russo & Zou, 2016), with remedies that restrict what the analyst may see. Those are not what discovery systems deploy, and whether the corrections they do deploy survive a real adaptive search has not been measured. We measure it, and give a remedy that leaves the loop untouched.

Contributions.

1. **A real discovery loop breaks the strongest existing test, and a causal control isolates the cause** (Section 5.2). On exact null markets built from CSI 300 data, an LLM alpha-mining loop rejects under Hansen's SPA at 15.2% and under Benjamini–Hochberg at 28.5% against a nominal 5% (26% and 36% on a parametric null), while the same loop with its feedback re-paired at random sits at nominal, as does a non-adaptive search of matched budget. The failure grows with search: at four times the budget SPA size doubles to 28% and Benjamini–Hochberg reaches 70% while both causal controls stay flat, and the practitioner's own acceptance bar never fires on a null at any budget.
2. **No fixed-design correction is valid for all adaptive searchers** (Proposition 1), and the failure reproduces with no market at all (Section 5.3): varying only the proposer's use of feedback moves SPA from 6% to 43% and Benjamini–Hochberg to 86%, with a control at nominal.
3. **Neither natural repair works, and they fail differently** (Section 5.4). Counting the trials better does not help: an effective trial count saturates at $3.6\times$ the control while every procedure's size keeps rising (Remark 3). Restricting what the searcher may see (the route taken by the reusable holdout (Dwork et al., 2015b) and the ladder (Blum & Hardt, 2015), which we run as arms) does restore nominal size, but the ladder halves what the search finds and the reusable holdout spends its whole query budget before the search ends.
4. **Calibrating the null on the entire procedure is exact** (Proposition 2), at the cost of R re-runs of the search and nothing else. Its power is in Bonferroni's class, and on a held-out period evaluated once, 92% of what it accepts keeps its significance. An extreme-value fit to the null maxima cuts the runs needed from about 100 to about 20 (Section 4).

Together these place three options on a common axis. A fixed-design correction is free and leaves the search untouched, and is invalid under adaptivity. The mechanisms built for adaptive queries are free and valid, and change what the search can find. Null calibration is valid and leaves the search untouched, and is paid for in compute. Alpha mining is the primary setting, because its evaluation data are the most obviously reused and its correction practice the most elaborate; a symbolic-regression domain and a US equity panel test whether the result depends on either.

2 Related work

Multiple testing over a fixed set of strategies. The data-snooping literature in finance corrects the best of M strategies for the fact that M were tried: the Reality Check (White, 2000; Sullivan et al., 1999), Hansen's studentised refinement (Hansen, 2005), stepwise extensions (Romano & Wolf, 2005; Hsu et al., 2010), the Benjamini–Yekutieli hurdles Harvey et al. (2016) propose for the factor zoo and the double-bootstrap hurdles of Harvey & Liu (2020), and the deflated Sharpe ratio and probability of backtest overfitting (Bailey & López de Prado, 2014; Bailey et al., 2017). Behind all of them is the resampling-based max-t adjustment of Westfall & Young (1993), the ancestor of our remedy; the difference is what sits inside the resampling loop. All take the candidate set as given. We show that no correction of that form survives a candidate set which is itself a function of the evaluation data (Proposition 1), and that its size tends to one as the search lengthens (Corollary 1).

Adaptive data analysis. An analyst choosing queries adaptively can overfit a holdout with polynomially few queries, and validity is restored by answering through a differentially private mechanism (Dwork et al., 2015a;b); preventing this is computationally hard in general (Hardt & Ullman, 2014); the bias is bounded by the mutual information between the data and the analyst's choices (Russo & Zou, 2016). The leaderboard is the closest setting to ours (Blum & Hardt, 2015; Roelofs et al., 2019), and related bias appears under adaptive collection (Nie et al., 2018), in selective inference (Fithian et al., 2014), and in online false-discovery control (Ramdas et al., 2017). That literature establishes

the risk and offers remedies that restrict what the analyst may see. We differ twice over: we measure whether the corrections practitioners *do* apply survive a real adaptive search, and we run the two closest remedies as arms rather than citing them (Section 5.4), finding that one halves what the search discovers and the other exhausts its query budget before the search ends. Our remedy leaves the loop untouched and pays in compute instead.

LLM-driven discovery. FunSearch (Romera-Paredes et al., 2024) and AlphaEvolve (Novikov et al., 2025), LLM-SR (Shojaee et al., 2025), the AI Scientist (Lu et al., 2024), and a fast-growing line of alpha miners (Yu et al., 2023; Wang et al., 2023; Tang et al., 2025; Shi et al., 2025) all evaluate proposals on a fixed dataset and condition the next proposal on the result. The loop is their contribution, and none audits the statistics of its own selection. The search is also growing more sophisticated, and Shi et al. (2025) drive it with Monte Carlo tree search over backtest feedback, which by Corollary 1 widens the gap between reported and real significance rather than closing it. Evaluation practice has consolidated around the same omission: Luo et al. (2026) benchmark language models on this task and score improvement in information coefficient *across search iterations*, which is the adaptive channel of Definition 1 measured and rewarded, with no null calibration beside it. The remedy of Section 4 is designed to drop into exactly such a protocol: it changes nothing about the search being scored.

3 WHY FIXED-DESIGN CORRECTIONS FAIL

An evaluation set $D = \{z_1, \dots, z_T\}$ is fixed. A *candidate* is a function f from a class $\mathcal{F}$ with score $S_D(f)$, in our experiments a t-statistic on the mean of a per-observation contribution. A *searcher* is a sequence of proposal kernels

$$f_i \sim \pi_i\big(\,\cdot\,\big|\,\mathcal{H}_{i-1}\big), \qquad \mathcal{H}_{i-1} = \big\{(f_j,\, S_D(f_j))\big\}_{j<i}, \qquad i = 1, \dots, M, \tag{1}$$

with *discovery statistic* $\max_{i \le M} S_D(f_i)$, or the set of f_i exceeding a threshold.

Definition 1 (Adaptivity)**.** *The searcher is* non-adaptive *if the realised candidate set $\{f_1, \dots, f_M\}$ is independent of D: each π_i is measurable with respect to the candidates $\{f_j\}_{j<i}$ alone and does not depend on the scores; it is* adaptive *otherwise.*

Non-adaptivity is the condition under which the fixed-design corrections below are valid, and the `random` arm of Section 4.4 is that control. The experiments add a *causal* control the definition does not cover: a kernel fed the scores with their pairing to candidates destroyed (`llm_shuffled`) still sees the score multiset, a function of D, but cannot learn which f scored high. Its validity on the exact null is verified rather than assumed (Section 5.2).

Let H_0 denote the global null: no $f \in \mathcal{F}$ has any true effect, so that under H_0 the scores $\{S_D(f)\}_{f \in \mathcal{F}}$ form a mean-zero random field indexed by $\mathcal{F}$ with covariance Σ induced by D.

Assumption 1 (Fixed design)**.** *The candidate set $\{f_1, \dots, f_M\}$ is chosen independently of D; equivalently, the joint law of $(S_D(f_1), \dots, S_D(f_M))$ under H_0 is the law of a fixed M-dimensional marginal of the field.*

Under Assumption 1 the null distribution of $\max_i S_D(f_i)$ is a functional of $(M, \Sigma_{f_1..f_M})$ alone, and every procedure considered here calibrates against that functional, differing only in how much of Σ it uses: the worst case for Bonferroni and Holm, positive regression dependence for Benjamini–Hochberg, an arbitrary-dependence inflation for Benjamini–Yekutieli, and a data estimate for the bootstrap tests and the deflated Sharpe ratio (Appendix A). The bootstrap procedures are the strongest, because they estimate Σ rather than bound it. They resample the *rows* of D while holding the *columns* $f_1, \dots, f_M$ fixed (Figure 2b), which is Assumption 1 made operational.

3.1 IMPOSSIBILITY

Proposition 1 (No fixed-design correction is valid for all adaptive searchers)**.** *Let a procedure's critical value be any functional of $(M, \hat{\Sigma})$ computed on the realised candidates, where $\hat{\Sigma}$ is the covariance of their scores estimated from D. For every nominal level α and every $\epsilon > 0$ there exists an adaptive searcher for which the procedure's size under H_0 exceeds $1 - \epsilon$. Equivalently, under*

H_0 the distribution of $\max_{i \le M} S_D(f_i)$ depends on the kernels $\{\pi_i\}$ and not only on M and the covariance of any fixed M-subset of the field.

Simulating the construction directly, a bootstrap max-t test rejects 54% of the time at nominal 5% where the same test against a fixed anchor rejects 2.7%, every column reported in both cases (Appendix H.1).

Proposition 1 says such a searcher exists. The next statement says the damage is not a fixed tax on adaptivity but one that grows without bound with the length of the search, which is what the budget sweep of Section 5.3 measures.

Corollary 1 (The failure grows with the budget)**.** *In the construction of Proposition 1 with $N = K$ variants and $M = 2K$, let $c(\rho) = \rho + \sqrt{1-\rho^2}$. Then $c(\rho) > 1$ for every $\rho \in (0,1)$, the search's statistic exceeds the critical value of any $(M, \hat{\Sigma})$-measurable correction by $(c(\rho)-1)\sqrt{2\log M}\,(1 + o(1))$, and the size of every such correction tends to 1 as $M \to \infty$. The worst case is $\rho = 1/\sqrt{2}$, where $c = \sqrt{2}$.*

The mechanism is visible in one line. The search anchors its correlated block on $\max_{k \le K} S_D(g_k) \approx \sqrt{2\log K}$; the correction, resampling rows with the columns held fixed, anchors the same block on an unconditioned draw of size $O(1)$. Both inherit $\sqrt{1-\rho^2}\sqrt{2\log N}$ from the variants, so the difference is $\rho\sqrt{2\log K}$ against a null whose fluctuations are $O(1/\sqrt{2\log K})$. Simulation of the construction gives sizes of 0.16, 0.58, 0.98 and 1.00 at $K = 10$, 10^2, 10^3 and 10^4 for $\rho = 1/\sqrt{2}$ (Appendix H.1; `tools/verify_budget_corollary.py`). The practical reading is uncomfortable: for a searcher of this shape more compute enlarges the error rather than averaging it out, and no correction computed on the output can see the difference.

Remark 1 (Multiplicity is not validity). Replacing M with an effective trial count, as the Harvey–Liu–Zhu and deflated-Sharpe families implicitly do, does not rescue the fixed design: the searcher in the proof reports every column, so its multiplicity is fully visible to the bootstrap, which computes the right null for those columns anchored on the wrong one. Empirically N_{eff} saturates while size keeps rising (Section 5.4).

3.2 Validity of full-procedure null calibration

Let $\mathcal{A}$ denote the entire procedure (kernels, evaluation, selection rule) as a measurable map from a dataset to a discovery statistic $A(D)$. Let σ be a transformation of D under which H_0 is invariant; in our setting, an independent permutation of the cross-section within each date, which destroys any relation between features and forward returns while preserving both marginals and all within-date dependence. Under H_0, $A(D) \overset{d}{=} A(\sigma D)$, and for $\sigma_1, \ldots, \sigma_R$ drawn uniformly,

$$\hat{p} = \frac{1 + \#\{r : A(\sigma_r D) \ge A(D)\}}{R+1} \tag{2}$$

where $\#$ counts the replicates satisfying the condition, satisfies $\Pr_{H_0}(\hat{p} \le \alpha) \le \alpha$ for every α, by exchangeability of $A(D), A(\sigma_1 D), \ldots, A(\sigma_R D)$ (Lehmann & Romano, 2005, Thm. 15.2.1). The argument is indifferent to the internals of $\mathcal{A}$: it is the max-t resampling adjustment of Westfall & Young (1993) with the search itself, rather than a fixed set of statistics, placed inside the resampling loop.

Proposition 2. *The permutation p-value* (2) *is exact for any searcher, adaptive or not, provided $\mathcal{A}$ is re-run in full on each $\sigma_r D$.*

Every standard correction is a functional of the *output* of $\mathcal{A}$ on D; null calibration is a functional of $\mathcal{A}$ itself. By Proposition 1 that is the only place the information resides. The remedy costs R re-runs of the search and nothing else (Appendix H.1).

4 Null calibration for adaptive search

4.1 Null constructions

Three transformations of the evaluation data under which H_0 holds by construction are used, so that results can be checked against what each preserves (Appendix E, Table 4). *Within-date permutation*

(perm) independently permutes the cross-section of forward returns on every date, preserving every feature's time series, the cross-sectional return distribution, and all dependence among candidates that runs through the features; it is exact for rank-correlation statistics by exchangeability. *Forward time shift* (shift) re-aligns returns against features by one forward offset of at least 231 trading days, the longest grammar window plus the horizon, with no wrap, preserving the serial dependence that perm destroys; its replicates are dependent, no exactness is claimed, and it serves the per-candidate procedures only. *Parametric simulation* (sim) draws returns from a factor model fitted to the real panel with the feature–return link removed. Block resampling is excluded rather than reported: a block whose offset falls inside a feature's window supplies the feature its own target, and the control then rejects on every replicate.

4.2 Calibrated hurdle and p-value

For a procedure $\mathcal{A}$ and R null replicates, the calibrated hurdle at level α is the $(1-\alpha)$ quantile of $\{A(\sigma_r D)\}_{r\le R}$, and the p-value of the real run is (2). A candidate is *null-calibrated accepted* if its $|t|$ exceeds the hurdle.

The size of this rule cannot be read off the same replicates that set the hurdle, so it is measured out of sample: for each replicate the hurdle is recalibrated on the other $R-1$ and the held-out replicate judged against it. At $R=100$ that leave-one-out size is 6.0% against a nominal 5%, identically for every arm and every null construction. The identity is the point: the rule uses only the exchangeability of the null draws, never the number of candidates or how they were chosen, so its size cannot depend on the searcher, and the one point of excess is the finite-R bias of an interpolated quantile. Every competing correction's size instead moves with the arm and the budget, reaching 70% on the feedback arm.

That the empirical quantile needs more null markets than a practitioner would guess is a fact about the quantile, not about the information in R draws: each replicate contributes a maximum over many dependent scores, which is asymptotically extreme-value, so a two-parameter fit uses the whole sample's shape rather than its top order statistic. Substituting a moment-matched Gumbel fit is closer to nominal than the empirical quantile at every budget below 50, on every arm and both markets we tested: on CSI 300 at $R=20$ it gives held-out sizes of 4.5–7.5% against the quantile's 8.9–9.4%, so twenty runs buy roughly what a hundred otherwise would. Its absolute calibration depends on how well each arm's tail matches the family, and the mis-specification is deliberate: the null maxima have a bounded upper tail, so forcing an unbounded one errs conservatively. Fitting the correct three-parameter family is worse at these sample sizes, because the extra parameter costs more in variance than it removes in bias (Appendix H.9).

4.3 Effective trial count

An effective trial count N_{eff} is reported for each arm by inverting the null maximum: if the largest $|t|$ over M candidates were the maximum of N independent draws from a single-trial reference F, then $\Pr(\max \le x) = F(x)^N$. The reference is the non-adaptive random arm's score distribution on the same nulls, *shared* across arms, because normalising each arm against its own pool would give a low N_{eff} to any search that concentrates its candidates, selection bias or not. For a non-adaptive arm N_{eff}/M is below one by construction, since candidates from one grammar are dependent. The estimate converges slowly in R, so N_{eff} is read as an ordering of arms rather than as a count (Appendix H.2).

4.4 Arms

All arms share one grammar, one evaluator, a budget of $M=500$ evaluations in batches of 20, and one scoring statistic; they differ only in the proposal kernel of (1). random draws uniformly from the grammar prior (the non-adaptive control); gp is tournament-selection genetic programming with fitness $|S_D|$ (adaptive); llm shows a language model, each round, the top-k scored formulas and a random sample of the rest and asks for new ones (adaptive); llm_shuffled runs the identical loop with the scores permuted across formulas before display, the causal control of Section 3; llm_zeroshot is the same model with no feedback table. The contrast that carries the paper is llm against llm_shuffled: nothing differs but the pairing. Every prompt template and grammar

arm	adaptive	R	$\lvert t\rvert \geq 2$	practitioner	Bonf.	BH	BY	RC	SPA	DSR	median p_{SPA}	KS p vs $U[0,1]$
`random`	no	400	99.0*	0.0	4.0	5.8	1.5	6.2	6.8	1.5	0.476	0.296
`gp`	yes	400	98.0*	0.0	6.8	12.8*	4.0	8.8*	9.0*	1.8	0.409	**5.4e-04**
`llm`	yes	400	100.0*	0.0	12.5*	28.5*	8.5*	15.0*	15.2*	0.8	0.288	**1.7e-20**
`llm_shuffled`	no (unpaired)	400	99.2*	0.0	2.0	2.8	0.5	4.8	4.8	1.8	0.530	0.290
`llm_zeroshot`	no	400	99.0*	0.0	1.2	4.2	0.8	3.5	3.8	1.5	0.467	0.439

Table 1: Empirical size on within-date-permutation null markets, $R = 400$, nominal 5%: the fraction of replicates on which the procedure made at least one discovery. * marks a Wilson 95% interval excluding 5%. `random` and `llm_zeroshot` are non-adaptive; `llm_shuffled` receives every score with the pairing destroyed, and is non-adaptive under Definition 1. Arm contrasts use the exact paired test (Section 5.3).

is hashed into every trajectory, and a null replicate whose hash differs from the real run's is rejected; across the 2,000 replicates behind Table 1 there were none.

4.5 Protocol

The data are split by date into a search period A, a validation period B, and a holdout period C. Every search, null replicate and statistic in Section 5 uses period A only. Period C was evaluated once, after every acceptance rule was fixed, by a script that refuses to run twice (Section 5.5).

5 Experiments

5.1 Setup

The primary domain is alpha mining on the CSI 300 constituent set with point-in-time membership and daily OHLCV from a public bundle ending in September 2020; period A is 2010–2016, B is 2017–2018, C is 2019 onward. Candidates are expressions in a formulaic-alpha grammar over price and volume fields. The score is the mean daily rank information coefficient against next-day returns and its t-statistic; the return series supplied to SPA, the Reality Check and CSCV is a long–short decile portfolio entered for both f and $-f$, so the direction is chosen by the test. The language model is `gpt-oss-20b` at low reasoning effort. Null calibration uses $R = 400$ replicates per arm on the within-date and covariance-preserving nulls and $R = 100$ elsewhere, with every arm run on the *same* panels so that arm contrasts are paired. Three further settings guard against explanations specific to any one of them: a simulation with a dial controlling how much the proposer conditions on scores, a symbolic-regression domain, and a US equity panel from CRSP. Every configuration is in Appendices B, C and G.

5.2 A real discovery loop on real markets

Table 1 gives the fraction of 400 permutation-null replicates on which each procedure made at least one discovery. The non-adaptive control is at nominal on every row (SPA 6.8%, Wilson $[4.7, 9.6]$; Bonferroni 4.0%; Benjamini–Hochberg 5.8%), and so are the two arms that run the identical loop without paired feedback (`llm_shuffled` 4.8%, `llm_zeroshot` 3.8% on SPA). `llm`, whose proposer reads its own backtest results, reaches 15.2% on SPA ($[12.1, 19.1]$), 12.5% on Bonferroni and 28.5% on Benjamini–Hochberg ($[24.3, 33.1]$), with every interval excluding 5%; `gp` sits just above nominal at 9.0% on SPA and 12.8% on Benjamini–Hochberg. Because every arm runs on the same 400 panels the contrast is paired, and the exact test separates `llm` from the control and from both causal controls at $p \leq 10^{-4}$ on every procedure. Figure 6 shows why the size understates it: the SPA p-values are uniform for every arm without paired feedback (Kolmogorov–Smirnov $p = 0.30$, 0.29, 0.44) and are not for `llm` (1.7×10^{-20}) or `gp` (5.4×10^{-4}). Hansen's bootstrap resamples the evaluation rows while holding the realised candidates fixed, so it is correct for a searcher that did not choose those candidates using those rows and incorrect for one that did, in proportion to how much it did.

The effect is not an artefact of the permutation. On a covariance-preserving simulated null `llm` reaches SPA 26% and Benjamini–Hochberg 36% against a control at 8% and 6%, and the paired head-to-head is again decisive against both `random` and `llm_shuffled` ($p \leq 10^{-4}$). The control

is nominal on Benjamini–Hochberg but mildly above it on the bootstrap tests here, so on this null it is the arm contrast rather than the absolute level that carries the result. From the practitioner's side, $|t| \geq 2$ accepts 408 of `llm`'s 484 unique candidates where its calibrated hurdle accepts 352, and the effective trial count per candidate is 0.32 for `random` against 1.08 for `llm` and 0.19 for both unpaired arms. Appendix H.7 gives every table and uniformity test.

Against one draw, one null and one market, four checks reproduce the ordering: five seeds per arm, the forward shift, the parametric null, and the whole pipeline rerun on CSI 500 (Appendix H.7). Two settings test whether it depends on finance at all. On symbolic regression, where the null permutes the target across samples and nothing of the market remains, `llm` is again the only arm SPA calls out (12% against 2% for `random`). On a CRSP panel of the largest 300 US names over the same period boundaries, the non-adaptive control holds nominal on every procedure (SPA 4%, Benjamini–Hochberg 5%, p-values uniform at KS $p = 0.64$), so that panel's dependence is handled by the corrections and any adaptive excess on it is attributable to adaptivity (Appendix H.6).

5.3 Which adaptive searches break the tests

First, without a market. Figure 3 realises Proposition 1 with candidate count, dependence structure and evaluation-set size held fixed, varying only how much the proposer conditions on scores it has already seen. At the control every procedure is at nominal (SPA 6.0%, Benjamini–Hochberg 3.3%); as coupling rises to 0.95 SPA reaches 42.7%, Benjamini–Hochberg 86.0%, and Benjamini–Yekutieli, the hurdle Harvey et al. (2016) recommend for this problem, 42.7%. One observation there is available from no real market: as coupling rises, N_{eff} plateaus at 3.6× the control while the participation ratio of the realised score matrix falls from 6.1 to 1.5, so to the bootstrap the adaptive search resembles *fewer* trials exactly as it becomes more dangerous (Appendix H.3).

Then, on real markets. Genetic programming is adaptive under Definition 1, its fitness being S_D and correctly paired, yet it departs from nominal only mildly where the language model departs strongly: SPA size 9.0% against 15.2% on the within-date null, and 14% against 26% on the covariance-preserving null (Section 5.2). Proposition 1 is an existence result and does not predict which adaptive searcher fails by how much. We measured four candidate quantities, each stated as a prediction first; two are excluded and two hold only as orderings of the arms (Table 8, Appendix H.4). What both adaptive arms share is the mechanism of Proposition 1's construction: a language model cannot score anything on D, but it can propose variants of the formulas that scored well, and genetic mutation does the same with a different proposal distribution.

Because every arm sees the same 400 panels, the arm contrast in Table 1 is paired, and the exact McNemar test on the per-panel rejection indicators separates `llm` from the non-adaptive control at $p \leq 10^{-4}$ on Benjamini–Hochberg, the Reality Check and SPA alike, and from both causal controls at $p < 10^{-4}$ on all three. The same test on the per-panel SPA p-values (a Wilcoxon signed-rank, which uses how close each procedure came to rejecting rather than only whether it did) agrees and is sharper still: `llm` against `random` at $p = 1.4 \times 10^{-10}$ (median SPA p of 0.29 against 0.48) and against `llm_shuffled` at $p = 5.7 \times 10^{-16}$, with neither non-adaptive arm departing from the control ($p = 0.087$ and $p = 0.47$). Both are reported because the coarser one is the more conservative and the conclusion does not depend on choosing the sharper.

`gp` is more informative than a pass or a fail, because *which* procedures it breaks is the point. The exact paired test separates it from the control on Benjamini–Hochberg ($p = 6 \times 10^{-5}$) and Benjamini–Yekutieli ($p = 0.02$) but not on the Reality Check ($p = 0.20$) or SPA ($p = 0.26$). The false-discovery-rate procedures assume a dependence structure; the bootstrap tests estimate it. So the evolutionary arm's excess is concentration that a dependence-aware test absorbs, while `llm`'s survives the bootstrap tests too. That is the line Proposition 1 draws: the bootstrap is right for a fixed candidate set and wrong for one chosen by looking at the rows it resamples.

The failure grows with search. Raising the budget from 500 to 10,000 answers the question a practitioner asks first. Does the bar they already use protect them? It never fires at all, at any budget, so it cannot fail. What fails is the correction, and Figure 1 is the paper's claim in one view: the feedback arm reaches 28% and 70% by a budget of 2,000 alone, the evolutionary arm climbs more slowly to 22% and 64% by 10,000, and no control moves at any budget. Bonferroni and Benjamini–Yekutieli, which assume nothing about dependence, hold to 2,000 and then break outright,

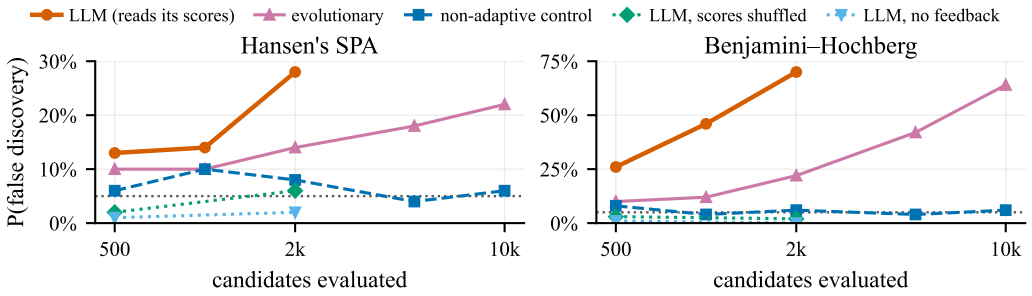


Figure 1: Chance of at least one discovery where none exists, against candidates evaluated; nominal dotted. Only the arms that condition on scores climb. $R = 100$ at budget 500 and $R = 50$ above; the causal controls ran at 500 and 2,000 (Table 7).

reaching 26% and 18% on `gp` and 34% and 38% on `llm`. The paired test separates `llm` from both causal controls at 2,000 ($p = 0.007$, $p = 0.0002$). So what grows is not the multiplicity but the adaptivity acting on it. The effective trial count says why: against the control's own null singles at each budget, N_{eff} per unique candidate rises from 1.1 to 4.4 for the feedback arm between 500 and 2,000 and from 0.46 to 10.1 for the evolutionary arm across the range, while the control's stays near one throughout. An adaptive searcher given more evaluations does not converge on the null, it learns it better.

Corollary 1 makes this specific enough to be wrong: the probit of empirical size should be linear in $\sqrt{\log M}$, with a positive slope for an adaptive arm and none for a non-adaptive one. Fitted to the sweep it holds in all nine arm–procedure cells, in the predicted order `llm` $>$ `gp` $>$ `random` ≈ 0 on every procedure. The sweep spans a factor of twenty and the corollary is asymptotic, so what is tested is the sign and the ordering rather than any slope's magnitude (Appendix H.1).

5.4 Two repairs, and what each one costs

There are two natural ways to try to rescue a fixed-design correction: count the trials better, or stop the searcher from seeing the scores. We measure both.

Counting trials better does not work at all. Replacing M with an effective trial count, as the Harvey–Liu–Zhu and deflated-Sharpe families implicitly do, fails in both directions at once. In the simulation N_{eff} reaches $3.4\times$ the control at the first increment of adaptivity and never exceeds $3.6\times$, while SPA's size travels from 16% to 43%: the quantity offered as the repair is flat exactly where the damage accumulates. Told N_{eff} in place of M, Bonferroni rejects 8.7% where nothing is adaptive. The failure is structural, not a matter of estimator quality, because the searcher of Proposition 1 reports every column it evaluated, so its multiplicity is fully visible to the bootstrap.

Restricting what the searcher sees works, and is not free. The adaptive-data-analysis literature takes the other route. The reusable holdout answers each query from a training half while the halves agree, and from a noised holdout otherwise, spending a fixed query budget (Dwork et al., 2015b); the ladder reveals a score only when it beats the incumbent by more than η (Blum & Hardt, 2015). We run both as arms on the US panel (same searcher, grammar, budget, evaluator and seed, with only the searcher's view of the scores changed) against $R = 100$ shared permutation nulls.

On the procedure where the unprotected adaptive arm is plainly broken, both work. Benjamini–Hochberg rejects on 14% of null markets for `gp`, a Wilson interval of $[8.5, 22.1]$ that excludes 5%; under the reusable holdout that falls to 6% and under the ladder to 4%, against a non-adaptive control at 5%. All arms see the same 100 panels, and the exact paired test separates `gp` from both mechanisms ($p = 0.039$ and $p = 0.002$) and from the control ($p = 0.035$), as the literature predicts.

What it costs is where they differ. The ladder pays in yield: it reveals 0.38% of queries, and on the real market the search ends with 65 candidates at $|t| \geq 2$ against the unprotected arm's 129, half the discoveries for the same 500 evaluations. The reusable holdout appears to pay nothing (126

candidates against 129) because it *runs out*. Averaged over the 100 null runs it spends 98.4 of its 100 holdout queries and refuses 11.8 more; on the real market the budget is exhausted at the 288th of 457 evaluated candidates, so the final 37% of the search is answered from the training half alone, which is exactly the conditioning the mechanism exists to prevent. Its size came out nominal here regardless, but the guarantee it advertises does not cover the run that was made, and nothing observable from inside the run distinguishes the two cases. Enlarging the query budget is not an escape: each query's accuracy degrades as the budget grows.

Two limits belong with these numbers. On Hansen's SPA the comparison cannot resolve anything: the unprotected arm is only mildly non-nominal here (8%, $[4.1, 15.0]$) and no mechanism is distinguishable from it under the paired test. And these are genetic-programming arms. We did not run the mechanisms on the language-model arm, where the unprotected inflation is largest, so this section establishes the shape of the trade-off rather than its size on the arm that breaks the corrections hardest. The direction is not in doubt, since both act on the searcher's view of its scores, but the magnitudes here are the evolutionary arm's.

The two repairs therefore fail against the same obstacle from opposite sides. What makes an adaptive search dangerous is the information flowing from the evaluation data into the next proposal. Counting trials does not measure that flow; throttling it far enough to be safe throttles the search, and throttling it cheaply means running out of throttle.

5.5 Out of sample, once

Everything above is on null markets, where validity is checkable because no effect exists. Whether what the rule accepts on the *real* market survives is answerable only once: period C was evaluated a single time, after every acceptance rule was frozen. Of the candidates clearing each arm's own hurdle on period A, the fraction still significant at $|t| \geq 2$ on period C is 0.92 for the feedback arm (325 of 352), and 0.68, 0.82, 0.69 and 0.87 for `random`, `gp`, `llm_shuffled` and `llm_zeroshot`; on CSI 500, a universe no search ever saw, 88–99% hold up on every arm. This is the payoff of the guarantee rather than a claim about the searchers: a rule valid under adaptivity should produce candidates that keep their significance out of sample, and it does on every arm.

It says nothing about novelty. Many survivors are volume and price-level expressions, and a sort on dollar volume earns the size and liquidity premium in any period. Neutralising each survivor's daily IC against dollar volume, reversal, momentum and realised volatility leaves about half of it, measured on period B: spending a held-out set twice, because the first answer raised a second question, is the failure this paper is about (Appendix H.7).

6 Discussion

A discovery loop that reads its own scores breaks every correction we applied, including the bootstrap tests built for this problem, and the failure grows with the budget. The same loop with its feedback destroyed stays at nominal, so the pairing is the cause, and counting trials better does not repair it.

What repairs it is calibrating on the procedure rather than on its output. Freeze the procedure, build R datasets on which the null holds by transformation, run it unchanged on each, and take the $(1 - \alpha)$ quantile of the discovery statistics as the hurdle. Report a non-adaptive proposer of matched budget, and for a language model the shuffled-feedback arm. No cheaper proxy will do.

Proposition 1 is an existence result. The corrections carry no guarantee under adaptivity, which is not the claim that every adaptive searcher breaks them. The evidence rests on one grammar and one model family, and CSI 500 replicates the marginal result but not the causal contrast (Appendix H.8).

These corrections assume a candidate set chosen without looking at the data. That described formulas written by hand. It does not describe a loop that reads its own scores. Nothing about the loop need change, only the null it is measured against.

Reproducibility statement. The code, prompts and result files behind every number in this paper will be released publicly on acceptance. Grammars and prompts with their hashes are in Appendices B–C, the configuration in Appendix G, the nulls and their checks in Section 4.1 and Appendix E, and the holdout protocol in Section 4.5.

REFERENCES

David H. Bailey and Marcos López de Prado. The deflated Sharpe ratio: Correcting for selection bias, backtest overfitting, and non-normality. *Journal of Portfolio Management*, 40(5):94–107, 2014.

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

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57(1):289–300, 1995.

Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, 29(4):1165–1188, 2001.

Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *PMLR*, pp. 1006–1014, 2015.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 117–126, 2015a.

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

William Fithian, Dennis Sun, and Jonathan Taylor. Optimal inference after model selection. *arXiv preprint arXiv:1410.2597*, 2014.

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

Moritz Hardt and Jonathan Ullman. Preventing false discovery in interactive data analysis is hard. In *55th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 454–463, 2014.

Campbell R. Harvey and Yan Liu. False (and missed) discoveries in financial economics. *Journal of Finance*, 75(5):2503–2553, 2020.

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

Kewei Hou, Chen Xue, and Lu Zhang. Replicating anomalies. *Review of Financial Studies*, 33(5): 2019–2133, 2020.

Po-Hsuan Hsu, Yu-Chin Hsu, and Chung-Ming Kuan. Testing the predictive ability of technical analysis using a new stepwise test without data snooping bias. *Journal of Empirical Finance*, 17 (3):471–484, 2010.

Erich L. Lehmann and Joseph P. Romano. *Testing Statistical Hypotheses*. Springer, 3 edition, 2005.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

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

Xinkun Nie, Xiaoying Tian, Jonathan Taylor, and James Zou. Why adaptively collected data have negative bias and how to correct for it. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, volume 84 of *PMLR*, 2018.

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

Dimitris N. Politis and Joseph P. Romano. The stationary bootstrap. *Journal of the American Statistical Association*, 89(428):1303–1313, 1994.

Dimitris N. Politis and Halbert White. Automatic block-length selection for the dependent bootstrap. *Econometric Reviews*, 23(1):53–70, 2004.

Aaditya Ramdas, Fanny Yang, Martin J. Wainwright, and Michael I. Jordan. Online control of the false discovery rate with decaying memory. In *Advances in Neural Information Processing Systems 30*, 2017.

Rebecca Roelofs, Vaishaal Shankar, Benjamin Recht, Sara Fridovich-Keil, Moritz Hardt, John Miller, and Ludwig Schmidt. A meta-analysis of overfitting in machine learning. In *Advances in Neural Information Processing Systems 32*, pp. 9175–9185, 2019.

Joseph P. Romano and Michael Wolf. Stepwise multiple testing as formalized data snooping. *Econometrica*, 73(4):1237–1282, 2005.

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

Daniel Russo and James Zou. Controlling bias in adaptive data analysis using information theory. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 51 of *Proceedings of Machine Learning Research*, pp. 1232–1240, 2016.

Yu Shi, Yitong Duan, and Jian Li. Navigating the alpha jungle: An LLM-powered MCTS framework for formulaic factor mining. *arXiv preprint arXiv:2505.11122*, 2025.

Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K. Reddy. LLM-SR: Scientific equation discovery via programming with large language models. In *International Conference on Learning Representations (ICLR)*, 2025. Oral.

Ryan Sullivan, Allan Timmermann, and Halbert White. Data-snooping, technical trading rule performance, and the bootstrap. *Journal of Finance*, 54(5):1647–1691, 1999.

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.

Saizhuo Wang, Hang Yuan, Leon Zhou, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. Alpha-GPT: Human-AI interactive alpha mining for quantitative investment. *arXiv preprint arXiv:2308.00016*, 2023.

Peter H. Westfall and S. Stanley Young. *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*. Wiley, New York, 1993.

Halbert White. A reality check for data snooping. *Econometrica*, 68(5):1097–1126, 2000.

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*, 2023. AlphaGen.

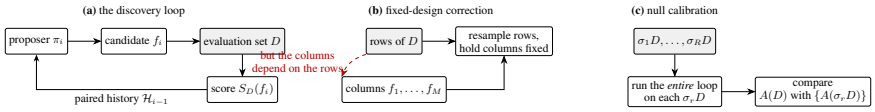


Figure 2: **(a)** The loop: each proposal is conditioned on earlier scores on the same evaluation set. **(b)** Standard corrections resample the rows of D with the candidate columns held fixed; under adaptivity those columns are a function of the rows. **(c)** Null calibration re-runs the procedure on data where the null holds by construction, exact for any searcher.

procedure	uses M	uses Σ	how
Bonferroni, Holm	yes	no (worst case)	union bound
Benjamini–Hochberg	yes	no (PRDS)	step-up on ordered p
Benjamini–Yekutieli / HLZ	yes	no (arbitrary)	BH with $\sum_k 1/k$ inflation
White's Reality Check	yes	estimated	bootstrap the M-vector
Hansen's SPA	yes	estimated	studentised bootstrap, recentred
Deflated Sharpe	yes	via Var of trial Sharpes	extreme-value $\mathbb{E}[\max]$

Table 2: Every procedure calibrates against a functional of (M, Σ) computed on a fixed candidate set.

A Procedures and implementation notes

Settings that determine size. SPA and the Reality Check use the stationary bootstrap (Politis & Romano, 1994) with the block length chosen per series by the Politis–White rule (Politis & White, 2004). Every candidate enters SPA, the Reality Check and CSCV as both f and $-f$, so that the trading direction is chosen by the test rather than from the data. The deflated Sharpe ratio is told $2M$ trials for the same reason. The N_{eff} estimator of Section 4.3 uses the non-adaptive arm's individual scores as a reference shared across arms. Every null panel is probed before use with random lagged-return features at short and long lags (Appendix E); panels that fail the probe are excluded from every arm's calibration and rebuilt, and the number of replicates reported is the number that passed.

Three errors in earlier versions, and how they were detected. Each of the following inflated the apparent effect of adaptivity while it stood, and each was detected by the non-adaptive control failing to sit at nominal. (i) N_{eff} normalised against each arm's own candidate pool assigns a low count to any search that concentrates its candidates, whether or not it carries selection bias; two arms with the same null maximum read 18 against 222. (ii) A fixed bootstrap block length of ten rejected at 18% on i.i.d. noise with no selection. (iii) Orienting each candidate to its profitable direction before a one-sided test raised the control's SPA size to 17%, and before CSCV it let the sign read every held-out block, moving the probability of backtest overfitting on pure noise from 0.50 to 0.70. Every reported number post-dates the correction of all three.

B Grammars

Alpha grammar. Expressions are trees over a dates×stocks matrix. Unary operators: `abs`, `log` (NaN for non-positive input), `sign`, and three cross-sectional operators applied on each date—`rank` (percentile in $[0, 1]$), `zscore`, `scale` (rescale so $\sum |x| = 1$). Binary: `add`, `sub`, `mul`, `div` (protected; NaN when the denominator is tiny), `min`, `max`. Time-series operators with an integer window d: `delay`, `delta`, `ts_mean`, `ts_std`, `ts_sum`, `ts_max`, `ts_min`, `ts_argmax`, `ts_argmin` (days since the window extremum, divided by d), `ts_rank` (percentile of today within the trailing window), `decay_linear`, `ts_corr`, `ts_cov`. Fields: `open`, `high`, `low`, `close`, `volume`, `vwap`. Windows $d \in \{1, 2, 3, 5, 10, 20, 30, 60, 120, 230\}$. Constants $\{-2, -1, -0.5, 0.5, 1, 2, 5, 10\}$. Maximum depth 6, maximum 40 nodes; anything outside the grammar is rejected before evaluation and counted as a parse failure. This is the validity limit. Proposals are separately capped at depth 4 (Table 5), so the grammar admits expressions no arm proposes.

Regression grammar. Unary: `neg`, `abs`, `sqrt` (of $|x|$), `square` (clipped), `log` (of $|x| + 10^{-8}$), `exp` (argument clipped to $[-20, 20]$), `sin`, `cos`, `inv` (protected). Binary: `add`, `sub`, `mul`, `div`

(protected). Variables $x_0, \ldots, x_{p-1}$; constants $\{-2, -1, -0.5, 0.5, 1, 2, 3\}$. Maximum depth 6, maximum 40 nodes.

Canonicalisation. In both grammars the arguments of commutative operators (`add`, `mul`, `min`, `max`) are sorted by their string form before the expression is keyed, so that `add(x,y)` and `add(y,x)` are one candidate. M, the participation ratio, N_{eff}/M and every duplicate statistic count canonical forms. Non-commutative operators are not collapsed: `sub(x,y)` and `sub(y,x)` are distinct trials whose returns are negatives of each other, and both enter the tests as such.

C PROMPTS

Every template below, together with the grammar specification it embeds, is hashed (SHA-1) and the hash is written into each trajectory's configuration. A null replicate whose hash differs from the real run's does not calibrate it. Alpha domain: `539b7bc3e0c1...`; regression domain ($p = 6$, $n = 800$): `8485b2c8f0bf...`. The feedback template shows the top $k = 15$ candidates by $|\text{score}|$ and a random sample of 10 of the rest; the truncation rule is part of the procedure being calibrated and is identical in real and null runs.

Alpha domain, system message.

```
You are a quantitative researcher searching for cross-sectional equity alpha
factors. You propose formulaic alphas in a restricted expression language.

You will be given the language specification and, in later rounds, a table of
formulas that have already been evaluated together with their in-sample scores.
Your job is to propose NEW formulas that you expect to score well.

Rules you must follow exactly:
- Use ONLY the operators, data fields, windows and constants listed. Anything
  else is rejected without being evaluated.
- Every formula must be a single expression on one line, with no assignment,
  no comments, and no explanation.
- Do not repeat a formula that already appears in the table.
- Output ONLY a JSON array of strings. No prose before or after it.

Example of a valid response:
["rank(mul(-1.0, delta(log(close), 5)))", "ts_corr(rank(close), rank(volume), 20)"]
```

Alpha domain, user message (feedback rounds). The grammar specification of Appendix B is prepended verbatim, then:

```
HOW YOUR FORMULAS ARE SCORED
Each formula produces a score for every stock on every day. On each day we
compute the Spearman rank correlation between those scores and the NEXT day's
return (the "daily IC"). The reported number is the mean daily IC over the
sample and its t-statistic. Higher absolute IC is better. A mean IC of 0.02 is
weak but real; 0.05 is strong; anything above 0.15 almost certainly means the
formula is exploiting an artefact rather than a real effect.

ROUND {r}. {n} formulas evaluated so far.

BEST FORMULAS SO FAR (mean IC, t-statistic, formula):
  {top-k table}

A RANDOM SAMPLE OF OTHER FORMULAS TRIED (so you can see what did NOT work):
  {sample table}

Propose 20 NEW formulas. Build on what worked, but do not simply permute the
window lengths of the best formula -- explore genuinely different structures too.

Output ONLY the JSON array.
```

The first-round message replaces the two tables and the final instruction with "Propose 20 diverse candidate alpha formulas. Aim for variety: include momentum, reversal, volatility, volume and price-volume interaction ideas, at a range of time horizons." The `llm_zeroshot` arm receives the first-round message every round. The `llm_shuffled` arm receives the feedback message with the (IC, t) pairs permuted across formulas before the tables are rendered.

Regression domain. The system message is the alpha message with "quantitative researcher searching for cross-sectional equity alpha factors" replaced by "scientist searching for a closed-form expression that predicts a target variable y from features x0, x1, . . . ", "formulas" by "expressions", and the example by `["mul(x0, x1)", "add(sin(x2), mul(-0.5, square(x0)))"]`. The scoring block is:

```
HOW YOUR EXPRESSIONS ARE SCORED
Each expression is evaluated on all 800 samples to produce a vector of
predictions. The score is the Pearson correlation between those predictions and
the target y, together with its t-statistic. Higher absolute correlation is
better. Expressions that are undefined or constant on too many samples are
rejected.
```

It stops there. The alpha block's calibration sentence—"0.02 is weak but real; 0.05 is strong"—is domain knowledge with no counterpart in symbolic regression, and an invented equivalent would give this arm a hint the alpha arm never had. The first-round variety instruction reads "include products and ratios of features, nonlinear transforms of single features, and additive combinations"; the feedback instruction warns against permuting "the constants of the best expression" rather than its windows.

D REASONING MODELS AS PROPOSERS

The proposer is `gpt-oss-20b`, a reasoning model: its chat template renders a reasoning-effort setting into the system message, and at generation time the model emits an *analysis* channel before the *final* channel that carries the answer. At the template's default effort with a 1,024-token completion budget, the model exhausts the budget in the analysis channel and never reaches the final channel. No JSON array is ever produced. The extractor, which is deliberately permissive, then falls through to a line-by-line scan and recovers every parenthesised string from the model's scratchpad. On the first probe run this yielded 287 "proposals" from sixteen completions, of which 246 were rejected by the grammar: a parse-failure rate of 0.857. Had it gone unnoticed, the `llm` arm would have been searching the model's chain-of-thought rather than its proposals, and would not have been the object the paper describes.

Table 3 shows the sweep that fixed it, run on the identical logged prompt. The effect is not the token budget. Raising the budget to 4,096 lets every completion finish but *worsens* parsing, from 0.461 to 0.681: a longer scratchpad gives the line scanner more to pick up, and a bracketed list inside the reasoning can be matched before the answer is. Reasoning effort is the variable that matters. At low effort every completion finishes inside 2,048 tokens, the parse-failure rate falls to 0.063, and the same four completions finish roughly eight times faster, because the model stops reasoning at length about a task that does not need it. The gain is in wall time rather than decoding speed: tokens per second not quite doubles, and the model simply emits far fewer tokens. The experiments use low effort with a 2,048-token budget; both settings are recorded in every run's configuration fingerprint, so a null replicate that differed from the real run in either would be rejected by the configuration check.

budget (tokens)	reasoning effort	wall (s)	tok/s	completions finished	parse-failure rate
1,024	default	134.1	29.5	2 / 4	0.461
4,096	default	177.2	41.6	4 / 4	0.681
4,096	low	18.8	52.8	4 / 4	0.295
2,048	low	17.5	54.7	4 / 4	0.063

Table 3: Decoding sweep for the reasoning-model proposer, same prompt throughout, four completions per setting on one GCD. "Finished" counts completions that ended before the budget. Raising the budget alone makes parsing worse; lowering reasoning effort fixes it and is eight times faster.

Two consequences generalise beyond this model. First, any discovery loop that wraps a reasoning model must verify that its parser is reading the answer channel and not the scratchpad; a permissive extractor will silently do the latter and report a plausible failure rate rather than an obviously broken one. Second, reasoning effort is part of what the proposer *is*. The setting is rendered into the prompt, so it belongs in the method section and in the run fingerprint, not in an implementation note; the arm reported here is `gpt-oss-20b` at low effort, and a different effort is a different proposer.

E NULL CONSTRUCTIONS: CHECKS AND THE EXCLUDED CONSTRUCTION

	`perm`	`shift`	`sim`
feature time series	preserved	preserved	fitted model
cross-sectional return distribution, per date	preserved	preserved	fitted
within-date dependence among candidates	preserved	preserved	fitted factors
return autocorrelation	destroyed	preserved	fitted
stock identity (unconditional stock means)	destroyed	kept	absent
valid for portfolio-return statistics (SPA, RC, CSCV)	yes	no (fixed basket)	fitted
feature–return relation	destroyed	destroyed	absent
exact for rank-correlation statistics	yes	no (dependent offsets)	no

Table 4: What each null construction preserves and destroys.

Sanity probe. Before a null is used for calibration it is probed with 200 expressions drawn from the grammar, in two arms: lagged returns at lags one to five, and lagged returns at lags drawn up to the longest grammar window. On the within-date permutation of CSI 300 the short-lag probe's mean IC is 7.6×10^{-5} with standard deviation 1.0×10^{-3} against an i.i.d. reference of 7.5×10^{-4}, and its largest |IC| is 2.0×10^{-3}; the excess spread over the i.i.d. reference is the within-date dependence the permutation preserves. Storey's $\hat{\pi}_0(\lambda = 0.5)$ on the `random` arm's unique candidates, per permutation replicate ($R = 100$): median 0.94, mean 0.97, interquartile range $[0.57, 1.38]$, full range $[0.36, 1.69]$. The median sits at one, as a valid null requires; the spread is wide because roughly 290 candidates drawn from about six effective clusters give $\hat{\pi}_0$ the sampling variance of a handful of p-values rather than of 290, so a single cluster landing near the threshold moves the whole estimate. The heaviest replicates sit below one (replicate 86, with 166 candidates past $|t| = 2$, at 0.45; replicate 40, with 146, at 0.37; replicate 61, with 96, at 0.44) but not uniquely so (replicate 72, with four, at 0.40); the rank correlation between $\hat{\pi}_0$ and the count past the threshold is only -0.26. Nothing here is a panel failing the null; it is the dependence of Section 5.3 seen from a different angle.

Why the shift is forward-only. A circular shift—one that takes the last o dates' returns from the start of the sample—is not a null for this grammar. On CSI 300 at $R = 100$ the non-adaptive control had a median 37% of its candidates past $|t| \geq 2$, against a median of 5 candidates on the within-date permutation, and the excess is driven by the mean IC rather than its standard error, is unrelated to lookback-window length (Spearman -0.01 between a candidate's compounded window and its $|t|$), and is absent for pure volume-level features ($|t| < 1$). What predicts it is the offset: the count past $|t| = 2$ has Spearman $+0.91$ with o across the 100 replicates, replicates with $o < 500$ are indistinguishable from the within-date null, and those with $o > 1100$ have about 130 of 290 candidates over the line. The channel is the wrap. A price level is the cumulative return since listing, so a return from the start of the sample lies inside every price-based feature evaluated late in the sample whatever the feature's window; `close` itself has window zero, which is also why the block bootstrap below inflates its shortest windows most. The forward shift has no wrap: every return the null pairs with a feature lies strictly after every observation the feature is built from, and the last o dates are dropped.

Stock identity. A forward shift keeps each return with its own stock, and whether that matters depends on the statistic. The rank IC ranks within a date and is blind to a persistent characteristic: pure level features of price and volume have $|t| < 1.5$ on clean forward offsets, the control's count past $|t| = 2$ is perm-like, and Bonferroni, Benjamini–Hochberg and Benjamini–Yekutieli sit at nominal on it (6%, 6%, 0%). The long-short decile return that SPA and the Reality Check consume is not blind to it. A sort on a persistent characteristic earns that characteristic's premium in every era, so on a panel with identity kept the extreme-decile portfolio of a price-level or dollar-volume proxy is profitable while its all-stock rank correlation is not: on one replicate the largest portfolio $|t|$ is 6.2 against 3.3 on the within-date null while the largest IC $|t|$ is 2.6 against 2.9, the correlation between a candidate's portfolio t and its IC t falls from 0.70 to 0.03, and the non-adaptive control rejects under SPA on 96% of replicates. Two repairs were tried. Demeaning each series fails: only the segment after o is paired, so a series' full-sample mean is not the mean of the returns actually used, and the difference is an era-driven per-stock offset that one offset applied to every date turns

into a level every slow feature picks up (142 of 285 control candidates past $|t| = 2$ with demeaning against 17 without on one seed; demeaning a stock's own series is worse still, because its mean also contains every trailing-return feature's window). Permuting the stock labels by a fixed derangement destroys identity, leaves the IC statistics unchanged (10 candidates past the line against 17 bare on the same seed; level probe $|t| \leq 0.8$), and removes the dollar-volume proxies from the top of the portfolio-t list, but does not bring the control's SPA to nominal ($p = 0.002$, 0.04 and 0.06 on three seeds against 0.22 on the permutation panel; on one of them 127 of 283 candidates have a portfolio $|t| \geq 2$). The channel is more general than identity. Any feature with a persistent cross-sectional ranking defines a nearly fixed basket of stocks; pairing that basket with any fixed assignment of another era's returns gives the basket that era's idiosyncratic premium, and a fixed basket's mean return over 1,370 days is not zero. The within-date permutation redraws the basket every day, which is why it is exact for portfolio statistics, and a shift with any fixed pairing cannot be. The shift null is therefore used for the per-candidate procedures, for which it is valid (control at nominal: Bonferroni 6%, Benjamini–Hochberg 6%, Benjamini–Yekutieli 0%), and the bootstrap tests are not applicable under it (Section 4.1). The probe carries a third arm—the $|t|$ of the raw price level and of constant-per-stock characteristics—which flags the circular construction, and the portfolio t of the same features is the check on the basket channel for any construction that keeps a fixed pairing.

Two properties of the shift null. Its replicates are dependent: offsets a few days apart pair the features with return windows that overlap almost completely, so R shift replicates are far fewer effective draws than R within-date permutations, and the offsets are drawn from a one-year range to reduce the overlap at the cost of dates. And because the forward truncation drops a different tail of the sample for every offset, the offsets do not form a group and no exactness is claimed; the shift null is the robustness check on serial dependence in Section 5.2, not the primary null.

The excluded construction. A stationary-bootstrap resampling of whole date rows of the return panel is excluded. Within a block every date carries one offset o between the row's true date and its position; whenever $o \in [-230, -1]$ the "forward" return assigned to date t is a past return already contained in any feature with a lookback window longer than $|o|$—`delta(close,5)` at t contains $\texttt{ret}_{t-3}$ by construction. On CSI 300 at $R = 100$ the non-adaptive control rejected under SPA on every replicate and under Bonferroni on 54%, uniformly across arms.

A Newey–West standard error with 20 lags takes the control's fraction of candidates past $|t| = 2$ on this construction from 27% to 4%, which would suggest the panel is a valid null tested with the wrong statistic—the correction leaves the circular shift's inflation untouched, so it is not a blunt instrument. It is not the right reading here, and the sign of the information coefficient is what distinguishes them. A standard-error correction can only remove inflation that lives in the variance, which leaves the sign of each replicate's IC a coin flip; a genuine feature–target relation fixes the sign. Across 16 block-permuted panels the median feature keeps the same IC sign on 94% of replicates with a mean IC 1.6–2.7 standard deviations from zero, against 56% and 0.1 under the within-date permutation. The location has moved, and the HAC standard error is absorbing a real effect rather than correcting a spurious one.

Two channels produce it, and a feature escapes only if it is exposed to neither. A long trailing window crosses block seams, as above. Separately, block permutation moves whole date rows, so each day's cross-section is preserved and any persistent characteristic holds a nearly fixed long–short basket that earns the sample's premium whatever order the days arrive in—the same channel that makes the portfolio tests unreadable under the forward shift (Section 4.1). Grouping features by both properties, the fraction of replicates on which the IC keeps its sign is 94% for long-window transient features, 94–100% for persistent ones whether their window is long or short, and 72% only for features that are both short-window and transient. Group sizes here are small (one to four features each), so this is a diagnosis of the construction rather than a measurement of either channel's size. The forward shift of Section 4.1 has a single offset of at least 231 days for the whole panel and no wrap, so no return it pairs with a feature can lie inside that feature's window or its level.

Why `blockperm` is excluded and `sim` is not. The simulated null's control is also above nominal (Table 10), so the criterion cannot be whether the control sits at 5%. It is whether the construction leaves a recoverable contrast between arms.

null	R	control SPA size	KS p	what is interpretable
`perm`	400	6.8% [4.7, 9.6]	0.30	absolute size and contrasts
`sim`	400	7.8% [5.5, 10.8]	2.3×10^{-4}	contrasts only
`blockperm`	248	100.0% [98.5, 100.0]	$< 10^{-300}$	neither

Under `sim` the inflation is common to all five arms (`llm_shuffled` 7.5%, `llm_zeroshot` 7.8%) and every arm runs on the same panels, so it cancels in the discordant counts of the paired test: `llm` at 26.2% against a 7.8% control is a 3.4× ratio at $p \leq 10^{-4}$ paired. Under `blockperm` the statistic is saturated. Two arms sit at 100%, nothing distinguishes them, and the arm ordering inverts—`llm` rejects on 89.1% of replicates, *below* the non-adaptive control's 100%—because a test that always rejects cannot rank searchers. `sim` is reported with its absolute levels qualified; `blockperm` is reported nowhere.

F Simulation details

Field. Shared noise $E \in \mathbb{R}^{T \times d}$, $T = 500$, $d = 20$, i.i.d. standard normal, drawn once per replicate and held fixed as the evaluation set. A fixed random map $W \in \mathbb{R}^{k \times d}$, $k = 8$. A candidate is $z \in \mathbb{R}^k$; its loading is $\ell(z) = zW/\|zW\|$ and its score series is $E\,\ell(z)^\top \in \mathbb{R}^T$, so every candidate has mean zero and unit variance in expectation and any two candidates are correlated through the shared factors. The discovery statistic of a candidate is the one-sided t-statistic of its series mean; a procedure discovers if it rejects for at least one candidate.

Kernels. Rounds $r = 1, \ldots, 25$, batch $b = 20$, $M = 500$. Let $z_r^\star$ be the candidate with the largest t among everything evaluated before round r. Under `perturb`, each proposal in round $r \geq 2$ is, independently with probability a, $z_r^\star + 0.35\,\xi$ with $\xi \sim N(0, I_k)$, and otherwise a fresh $N(0, I_k)$ draw. Under `screen`, each proposal is, with probability a, the $\arg\max$ of the t-statistic over $K = 20$ fresh $N(0, I_k)$ draws scored on E—the $K - 1$ losers are discarded and never enter the realised matrix—and otherwise a fresh draw. $a = 0$ is the shared non-adaptive control. Selection is one-sided throughout.

Statistics recorded per replicate. *Coupling:* for every proposal in rounds $r \geq 2$, the Pearson correlation of its series with the series of $z_r^\star$, averaged. *Participation ratio:* $(\sum_i \lambda_i)^2 / \sum_i \lambda_i^2$ over the eigenvalues of the $M \times M$ correlation matrix of the realised series. *Maximum* $|t|$ and every candidate's $|t|$.

N_{eff}. Per adaptivity level, the median over replicates of the maximum $|t|$, x_{med}, is inverted against the shared reference F—the empirical distribution of the individual $|t|$ values pooled over all replicates of the $a = 0$ control—as $N_{\text{eff}} = \log(1/2) / \log F(x_{\text{med}})$.

Procedures. Bonferroni, Holm, BH and BY at $\alpha = 0.05$ on one-sided p-values. Reality Check and SPA with 400 stationary-bootstrap resamples, block length by the Politis–White rule per replicate, studentised as in Hansen (2005). Empirical size is the fraction of 300 replicates with at least one discovery; intervals are Wilson 95%.

G Hyperparameters and seeds

All settings live in one configuration file whose hash is recorded in every result file; a null replicate whose hash differs from the real run's is rejected by the calibration script.

Seeds: real runs use seed 0 (LLM arms additionally seeds 1–4 for the seed-spread analysis); null replicate r uses seed $1000 + r$, so replicate sets are identical across arms and null types. A run stopped by a walltime limit is resumed from its last checkpoint with the full previous history replayed to the proposer, so it is the same procedure on the same data; the proposer's random stream after the resume point differs from an uninterrupted run's. No run behind any table in this paper was resumed. A run in which the proposer returns no new candidate in a round stops; the rule is part of the procedure and fires on real and null runs alike, so such runs are analysed as they ended and never replaced by a fresh draw, which would condition the null set on reaching the budget. The unique-candidate count each arm

Protocol	
period A (search) / B (validation) / C (holdout)	2010–2016 / 2017–2018 / 2019–2020-09-25
primary / secondary market	CSI 300 / CSI 500, point-in-time membership
Search, all arms	
budget (candidates evaluated) / batch	500 / 20
maximum expression depth (proposal)	4
Genetic programming	
population / tournament / elitism	100 / 5 / 5
$p_{\text{crossover}}$ / p_{mutate} / parsimony per node	0.6 / 0.35 / 5×10^{-4}
Language model	
model / temperature	`gpt-oss-20b` / 1.0
reasoning effort / completion budget	low / 2,048 tokens
feedback table: top-k / random sample	15 / 10
Evaluation	
long–short book	top and bottom decile, equal weight
minimum coverage	30% of stock-days
practitioner rule	IC ≥ 0.02, $\lvert t\rvert \geq 2$, annualised Sharpe ≥ 1
Null calibration	
constructions / replicates R / α	perm, shift, sim / 400 (perm, sim), 100 (shift) / 0.05
shift: minimum offset	231 dates (longest window 230 + horizon 1)
SPA, RC: bootstrap resamples	500 markets, 400 simulation, 300 symbolic regression
SPA, RC: block length	Politis–White, per series
CSCV: blocks / combinations	16 / 2,000 sampled
Simulation	
T / d / k / rounds $\times$ batch	500 / 20 / 8 / 25×20
perturb step / screen K	0.35 / 20
replicates / SPA resamples	300 / 400
Symbolic regression	
n / features / signal-to-noise	800 / 6 / 1.0
targets	product, ratio, harmonic, additive

Table 5: Settings behind every reported number.

actually reached is in Table 11. A run that ends in a proposer error is not a draw of the procedure and is rerun. Every result file records the git commit, Python and library versions, and the job identifier of the run. Scheduler job identifiers on the compute facility (named in the camera-ready; withheld here because the allocation identifies the authors): Table 1 and Figure 6, 5427577 (size calibration, permutation null) with the SPA p-value CDFs from 5425682; Table 10, 5427579; Table 11, 5427551; Table 13, 5428822 and 5428823; Table 12, 5430518–5430523. Null-market search trajectories (random, genetic, language-model arms): permutation 5425512, 5425513, 5423631 (CSI 300) and 5425549, 5425550, 5427553 (CSI 500); simulation 5425541, 5425542, 5425538; forward shift 5430519, 5430520, 5430521; the excluded block bootstrap 5425539, 5425540, 5425537.

G.1 Settings displaced from Section 5.1

Alpha mining, in full. The grammar covers price and volume fields with time-series and cross-sectional operators (Appendix B). The language model is `gpt-oss-20b`, served with vLLM on data-centre GPUs at low reasoning effort with a $2{,}048$-token budget; the effort setting is part of the proposer and is recorded in every run's fingerprint (Appendix D).

Simulation. A fixed evaluation set of $T = 500$ observations on $d = 20$ shared noise factors. A candidate is a point in a $k = 8$-dimensional latent space mapped by a fixed random matrix to unit-norm factor loadings, so its score series is a linear combination of the shared factors: every candidate is null, and candidates are dependent through the factors. A search runs 25 rounds of 20 proposals. Under the `perturb` kernel a proposal is, with probability a, the incumbent best plus Gaussian noise of scale 0.35, and otherwise a fresh draw; under the `screen` kernel it is, with probability a, the best of $K = 20$ fresh draws scored on the evaluation set, of which only the winner is emitted. Each setting uses 300 replicates, with 400 bootstrap resamples for SPA and the Reality Check. On the same runs

we record each proposal's coupling to the incumbent, the participation ratio PR of the realised score matrix—the inverse sum of squared normalised eigenvalues of its correlation matrix, an effective number of independent columns—and N_{eff} against the $a = 0$ column (Appendix F).

Symbolic regression. Targets are Feynman-style closed forms of two variables among $p = 6$ features drawn uniformly on $[-2, 2]^p$, with $n = 800$ samples and Gaussian noise at a set signal-to-noise ratio; the remaining features are distractors. The grammar is arithmetic with unary transforms and the budget is $M = 400$. The score is the Pearson correlation between $f(X)$ and y, and the per-sample series whose mean is that correlation is what SPA and CSCV consume, so every statistic transfers unchanged. The null permutes y across samples. The five arms are the same five; the LLM prompt omits calibration advice, since the alpha prompt's guidance on which IC values are weak or strong has no counterpart here (Appendix C).

US equities. The panel is built from the CRSP daily stock file via WRDS: ordinary common stock on NYSE, AMEX and Nasdaq, 2009–2021, from which the largest 300 names by market capitalisation are carried at any time. Membership is ranked on the last trading day of the *preceding* six-month period and then held fixed, with a $5 price screen at rebalance. Ranking within a period and applying the result to that same period is look-ahead worth about five percentage points a year on US large caps, and it passes every structural check. Prices are $|\texttt{prc}|$, since CRSP stores the price negative when the day's quote is a bid–ask midpoint rather than a trade; returns are CRSP's holding-period return, already adjusted for splits and dividends, so no adjustment factor is applied and split dates carry no artificial jump. CRSP daily has no VWAP, so that field is synthetic, $(\text{high} + \text{low} + \text{close})/3$, and is recorded as such in the panel metadata. The resulting panel spans 2,702 dates and 566 distinct permnos with 299 names per day, and its equal-weighted portfolio compounds at 13.1% a year against the S&P 500's $\approx 13.5\%$ over the same window. CRSP is licensed data and is not redistributable; the extraction script is included so that a subscriber can rebuild the panel exactly.

H Extended results and derivations

Material that supports the main text without being needed to follow it: the extended discussion of the theory (H.1), of the effective trial count (H.2), of the simulation (H.3), of the mechanisms that separate the adaptive arms (H.4), of power (H.5) and of the second domain (H.6).

H.1 Theory, extended

Contributions 2–4, as stated before condensing (moved verbatim).

2. No fixed-design correction is valid for all adaptive searchers (Proposition 1), and the failure is reproduced with no market (Section 5.3): holding candidate count, dependence and evaluation-set size fixed and varying only the proposer's use of feedback moves SPA from 6% to 43% and Benjamini–Hochberg to 86%, with a control at nominal.
3. Counting trials better cannot repair it (Remark 3, Table 6). An effective trial count recovered from the null maximum saturates at $3.6\times$ the control while the size of every bootstrap and false-discovery-rate procedure continues to rise; the quantity that tracks the failure is the gap between the multiplicity the search ran and the multiplicity its realised candidates exhibit, which no correction computed on the output can observe.
4. Calibrating the null on the entire procedure is exact and affordable (Proposition 2, Section 5.2). It costs R re-runs of the search and nothing else. Against ground truth on panels with planted signal its false discovery rate is zero where every fixed-design correction is also valid, at a power in Bonferroni's class; the guarantee it adds is the one the corrections lose under adaptivity, and its price is stated rather than hidden.

Proof sketch of Proposition 1 (moved verbatim).

Proof sketch. The searcher is of the form (1)—it conditions on the paired history and on nothing else—and every candidate it evaluates is reported. Let $\mathcal{F}_0$ contain K *representatives* $g_1, \ldots, g_K$ whose null scores are independent standard normals, and for each g_k a family of N *variants* $h_{k,1}, \ldots, h_{k,N}$

with $S_D(h_{k,n}) = \rho\, S_D(g_k) + \sqrt{1-\rho^2}\,\varepsilon_{k,n}$, $\varepsilon_{k,n}$ independent standard normals and $0 < \rho < 1$ fixed—a candidate class with cluster structure, as any grammar has. For $i \le K$ let π_i propose g_i (non-adaptive). For $i > K$ let π_i propose the variants of $g_{k^\star}$, $k^\star = \arg\max_{k \le K} S_D(g_k)$, which is measurable with respect to $\mathcal{H}_K$ and requires the pairing: with scores shuffled, $k^\star$ is a uniform draw. Take $M = K + N$. The realised set contains every column, so $\hat{\Sigma}$ is the true covariance of the $K + N$ columns: K independent columns and one block of N correlated with $g_{k^\star}$. A procedure calibrated on $(M, \hat{\Sigma})$ takes as its null the law of $\max$ over these columns with the block *anchored on an unconditioned representative*, $\rho Z + \sqrt{1-\rho^2}\max_n \varepsilon_n$ with $Z \sim N(0,1)$, whereas the search anchored it on the maximum of K: $\max_{i \le M} S_D(f_i) \ge \rho \max_{k \le K} S_D(g_k) + \sqrt{1-\rho^2}\max_n \varepsilon_{k^\star,n}$. The two differ by $\rho\,(\max_{k \le K} Z_k - Z)$, which grows without bound in K while every other term is common to both, so for $N \ge K$ the size exceeds $1 - \epsilon$ for K large enough. Nothing in $(M, \hat{\Sigma})$ records which column the block was anchored on; that information lives in the kernels. □

Recommended practice, as stated before condensing (moved verbatim).

Recommended practice. For a discovery loop that scores candidates on a fixed evaluation set and conditions later proposals on the result, a multiple-testing correction computed on the evaluated candidates communicates a guarantee that does not exist (Proposition 1). The alternative is to freeze the procedure—proposer, prompt, budget, evaluator, selection rule—by hash, construct R datasets on which the null holds by transformation (for a cross-sectional predictor, permute the target within each date), run the procedure unchanged on each, and take the $(1-\alpha)$ quantile of the R discovery statistics as the hurdle and (2) as the p-value. This is exact for any procedure (Proposition 2). Two controls make the result interpretable: a non-adaptive proposer of matched budget, which must sit at nominal under every procedure reported and otherwise indicates an implementation error, and, for a language-model proposer, the shuffled-feedback arm, which isolates the information channel at the cost of one line of code. The remedy's cost is R searches; at $R = 100$ the smallest attainable p-value is $1/101$, and the null runs pack onto CPU nodes since only the proposer needs an accelerator. A cheaper proxy proposer must not be substituted for the null runs: the guarantee comes from re-running the procedure itself.

What the remedy costs. the only place the information resides. The remedy costs R re-runs of the search and nothing else—no change to the proposer, prompt, grammar or selection rule, and no restriction on the analyst's access to the data as in Dwork et al. (2015b). $R = 100$ gives a minimum attainable $\hat{p}$ of $1/101$, sufficient at $\alpha = 0.05$. The transformation σ determines what the null preserves; Section 4.1 uses three, including one that preserves the serial dependence of returns and features.

Definition 1: the two controls. Non-adaptivity is exactly the condition under which the fixed-design corrections of the next subsection are valid, and it is the theoretical control: the `random` arm of Section 4.4. The experiments also use a second, *causal* control that the definition does not cover. A kernel fed the scores with the pairing destroyed—$\{(f_j,\, S_D(f_{\sigma(j)}))\}_{j<i}$ for a uniformly random permutation σ—still conditions on the multiset of scores, a function of D, and is adaptive by the letter of the definition; what it cannot do is learn which f scored high, which is the channel through which every mechanism in Section 3 operates. This is the `llm_shuffled` arm. It differs from the adaptive arm in the pairing and in nothing else, so the contrast between them isolates the pairing as the cause; that it is also valid on the exact null—at nominal, with uniform p-values—is verified in Section 5.2 rather than assumed, and Section 5.3 shows that destroying the pairing returns the kernel's measured coupling to the evaluation data to the level of uniform draws.

Proposition 1: numerical check and the off-book route. The construction is the one in the proof of Proposition 1, and the mechanism is visible in a bootstrap max-t test at nominal 5%: with $K = 50$ representatives, $N = 200$ variants and $\rho = 0.9$ its size is 54% when the block is anchored on the selected representative and 2.7% when it is anchored on a fixed one, every column reported in both cases. A second route to the same conclusion is a kernel that evaluates candidates it does not report—the off-book screening of Section 5.3—for which $\hat{\Sigma}$ sees one trial where the search ran many; it lies outside (1) unless the hidden evaluations are added to the history, and we do not need it.

Remark: what the bootstrap resamples.
Remark 2 (What the bootstrap resamples). RC and SPA are exactly valid for a fixed candidate set under weak dependence, and our non-adaptive control confirms it (Section 5.2). Under adaptivity the bootstrap resamples rows of D but conditions on columns that are functions of those rows; its null is the law of the statistic *given the search's output*, not the law of the search.

Remark: multiplicity is not validity, extended. Empirically (Section 5.3) N_{eff} reaches $3.4\times$ the control at the first increment of adaptivity and never exceeds $3.6\times$, while the size of every bootstrap and false-discovery-rate procedure continues to rise; Bonferroni and Holm, which bound rather than estimate, go from 2% on the non-adaptive control to 7–8% at the first increment and sit there, insensitive to the dial thereafter. Replacing M by N_{eff} in Bonferroni would loosen it—N_{eff} is below M for every arm—and could not track a size that moves while N_{eff} does not; measured, it rejects 8.7% on the non-adaptive control and 8–13% at every setting of either dial (Section 5.3). What tracks the size is the ratio of N_{eff} to the effective number of distinct columns the bootstrap sees—the multiplicity the search ran over the multiplicity it shows—which is invisible to any functional of the realised candidates.

H.2 The effective trial count, extended

The arms, as stated before condensing (moved verbatim). All arms share one grammar, one evaluator, a budget of $M = 500$ evaluations in batches of 20, and one scoring statistic; they differ only in the proposal kernel of (1). `random` draws uniformly from the grammar prior and is the non-adaptive control, for which every correction in Appendix A is provably valid. `gp` is tournament-selection genetic programming with a parsimony penalty and fitness $|S_D|$; adaptive. `llm` shows a language model, each round, the top-k scored formulas and a random sample of the rest and asks for new ones; adaptive. `llm_shuffled` runs the identical loop, prompt, table shape and score distribution with the scores permuted across formulas before display: the pairing is destroyed and nothing else. It is the causal control of Section 3: adaptive by the letter of Definition 1, since it sees the score multiset, but unable to learn which formula scored high, and verified at nominal with uniform p-values on the exact null (Section 5.2). `llm_zeroshot` is the same model with no feedback table—the pre-trained prior alone; non-adaptive.

The contrast that carries the paper is `llm` against `llm_shuffled`: anything the former does that the latter does not is attributable to information flowing from the evaluation data into the proposer, because nothing else differs. `llm_zeroshot` separates the prior from the loop; `gp` separates adaptivity from language models. Every prompt template and the grammar specification are hashed and the hash stored in every trajectory; a null replicate whose hash differs from the real run's is rejected. Across the 2,000 replicates behind Table 1 there were no mismatches.

How the estimate depends on R. The estimate depends on the tail of the reference distribution, which is heavy—the candidates within a replicate are dependent—and is therefore estimated slowly in R: the same estimator on the same arms gives an `llm`/`random` ratio of 2.1 at $R = 30$ and 3.4 at $R = 100$, the control's own N_{eff}/M moving from 1.07 to 0.32 as the tail fills in. We report $R = 100$ values and read N_{eff} as an ordering of arms rather than a count.

Semantic duplicates. Expressions that are distinct as text but produce the same signal after ranking (monotone re-expressions such as `rank(f)` or `scale(f)`) are 1–21% of M across arms and replicates, and the feedback arm is not the worst: its largest share is 12% against `llm_shuffled`'s 21%. So M is not inflated by them in a way that favours the adaptive arm.

The shift null, as stated in the main text before condensing. *Forward time shift* (`shift`) re-aligns the return panel against the feature panel by a single forward offset of at least 231 trading days—the longest grammar window plus the forecast horizon—without wrapping, so that every return a candidate is scored against lies strictly after every observation the candidate is built from; the last o dates are dropped. It preserves the full serial dependence of both returns and features, which `perm` destroys; a daily information-coefficient (IC) series on this null has the autocorrelation of a real one. Its replicates are dependent—nearby offsets pair the same returns—and no exactness is claimed for it; it is the robustness check of Section 5.2, not the primary null. The construction is non-circular for a measured reason: a wrapped offset pairs late features with returns from the start of

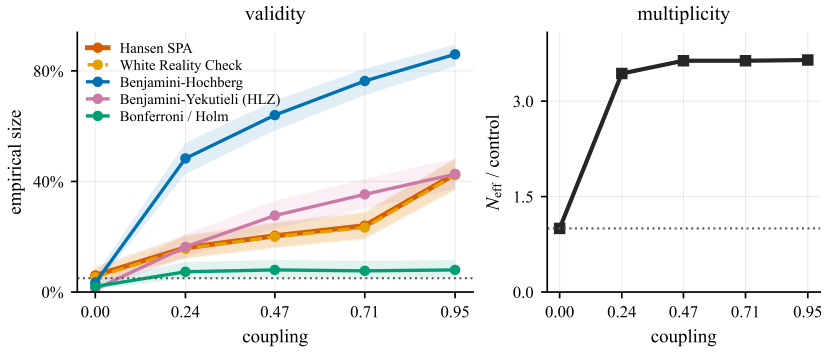


Figure 3: Simulation: every hypothesis is null, 500 candidates per run, 300 replicates, an evaluation set of 500 observations, and only the proposer's use of feedback varies. The x-axis is the measured coupling of each proposal to the incumbent best; the left-most point is the non-adaptive control, for which every procedure here is provably valid. **Left:** empirical size of each procedure, Wilson 95% bands, with the dotted line at the nominal 5%. **Right:** N_{eff} on the same runs relative to that control. Multiplicity saturates while validity continues to degrade.

kernel	dial	coupling	PR	$N_{\text{eff}}/N_{\text{eff}}^{(0)}$	N_{eff}/PR	SPA	RC	BH	BY	Bonf.	Bonf. (N_{eff})
control	0	0.00	6.1	1.00	4.8	6.0	5.3	3.3	1.0	2.0	8.7*
`perturb`	0.25	0.24	5.4	3.43	18.7	16.0*	15.7*	48.3*	16.3*	7.3	12.0*
`perturb`	0.5	0.47	3.7	3.64	29.1	20.3*	20.0*	64.0*	27.7*	8.0*	12.0*
`perturb`	0.75	0.71	2.3	3.64	46.8	24.0*	23.3*	76.3*	35.3*	7.7*	12.3*
`perturb`	1	0.95	1.5	3.65	72.6	42.7*	42.3*	86.0*	42.7*	8.0*	12.7*
`screen`	0.25	0.14	6.0	1.64	8.1	9.3*	9.0*	8.3*	1.7	2.3	8.0*
`screen`	0.5	0.29	5.4	2.11	11.5	12.0*	11.7*	16.3*	2.0	2.7	9.3*
`screen`	0.75	0.43	4.6	2.16	13.9	13.7*	13.3*	21.7*	2.3	3.0	9.3*
`screen`	1	0.58	3.8	2.08	16.2	18.3*	18.3*	27.0*	3.0	3.0	10.0*

Table 6: Simulation, both kernel families; the control row is shared. Coupling is the measured correlation of each proposal with the incumbent; PR the participation ratio of the realised score matrix; $N_{\text{eff}}/N_{\text{eff}}^{(0)}$ the effective trial count relative to the control; sizes are the fraction of 300 replicates with at least one discovery at nominal 5%.

the sample, and a price level is the cumulative return since listing, so those returns are inside every price-based feature whatever its window. Unlike `perm` it keeps each return with its own stock. For the rank IC that is immaterial on our market, but the long-short portfolio return that the bootstrap tests consume is not: any feature with a persistent cross-sectional ranking holds a nearly fixed basket, and under any fixed pairing with another era's returns that basket earns the era's premium. The shift null is therefore used for the per-candidate procedures only, and the bootstrap tests are not applicable under it (Appendix E).

H.3 The simulation, extended

Section 5.3, the right panel and the second kernel family (moved verbatim). The right panel realises Remark 3. N_{eff}, recovered from the null maximum against the control's single-trial distribution, rises to $3.43\times$ the control at the first increment and then plateaus at 3.64–3.65. Multiplicity saturates because a search can only be so redundant; validity continues to degrade because adaptivity continues to corrupt the null after it has stopped adding effective trials. The participation ratio of the realised score matrix—the effective number of distinct columns the bootstrap sees—falls from 6.1 to 1.5 over the same range: to the bootstrap, the adaptive search resembles *fewer* trials as it becomes more dangerous.

The second kernel family (Figure 4) selects the best of $K = 20$ candidates scored on the evaluation data and emits only the winner, so the realised set stays as distinct as random draws. It breaks SPA as well—18.3% at $a = 1$, with the Reality Check at 18.3% and BH at 27.0%—at a participation ratio

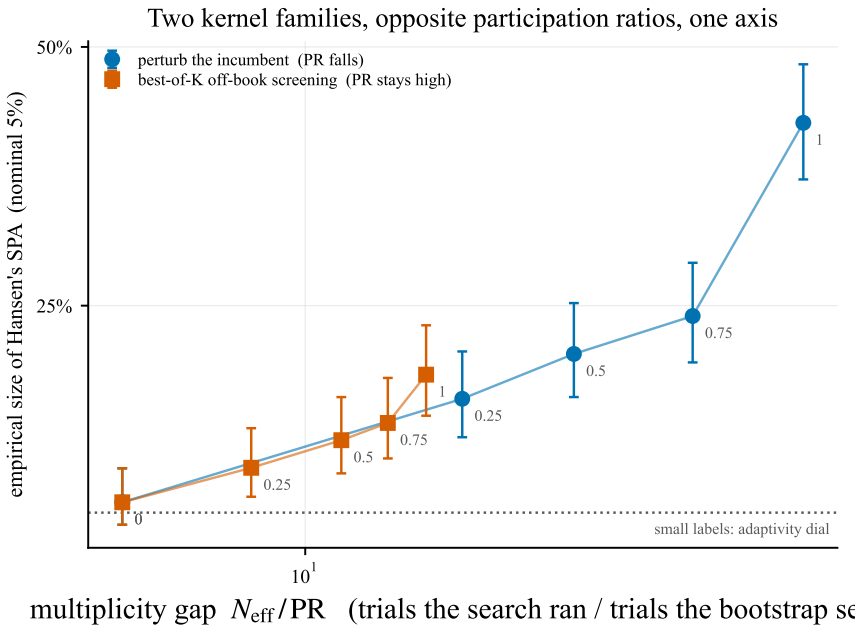


Figure 4: Simulation. Two kernel families at opposite ends of the participation-ratio axis both break SPA; N_{eff}/PR—trials the search implicitly ran over trials the bootstrap sees—orders every point of both up to sampling error (300 replicates, Wilson 95% bars; labels give the adaptivity dial). Whether the same ratio orders the real arms is examined in Section 5.3.

of 3.8 against the control's 6.1. The failure therefore occurs at both ends of the participation-ratio axis, and what orders every point of both families is the ratio N_{eff}/PR, from 4.8 at the control to 16.2 under `screen` and 72.6 under `perturb`. That gap is invisible to any correction computed on the realised candidates, which is the content of the proof of Proposition 1 stated as a number.

H.4 Mechanisms, extended

The mechanisms table (moved from Section 5.3).

The dependence of a candidate set, visible in the null (moved verbatim from Section 5.3). Before turning to what separates the adaptive arms, the null distribution of the *non-adaptive* arm shows what the corrections are up against. Across the 100 permutation replicates the number of the control's ∼290 candidates with $|t| \geq 2$ has median 5 and mean 11.8—4.1% of candidates, the tail a standard normal gives—but on four replicates it is 95, 96, 146 and 166, while the maximum $|t|$ on those replicates is only 2.9–3.6 and SPA does not reject. Every daily IC on a permuted market is exactly mean-zero for every candidate by exchangeability, so no replicate carries a signal; what varies is that the random arm's expressions collapse onto a few near-identical clusters (long-lag level features of price and volume; participation ratio about six), and a permutation that is loud for one cluster carries its members over the line together. The same dependence is why the control's N_{eff}/M is 0.32—a few clusters are worth a few dozen independent trials—why Benjamini–Hochberg sits above 5% on a valid null in this $R = 100$ set (8%, though at the $R = 400$ of Table 1 it settles to 5.8% and the excess is within sampling error), and why the bootstrap tests, which estimate $\hat{\Sigma}$, are the right comparison class. Dropping such replicates as "contaminated" would select the null on the outcome; they are the null.

The reading has a direct test, which it passes. Re-evaluating each replicate's candidate set on the other's panel, the counts follow the panel and not the set: the 285 expressions found on permutation 1000 place 31 candidates over the line on their own panel and 163 on permutation 1086, while the 305 expressions found on 1086 place 30 and 166. One permutation puts half of any random candidate set over $|t| = 2$, the other a tenth. On both runs 39% of the variance of the candidate

arm	budget	R	rule	Bonf.	BH	BY	SPA	N_{eff}/M
`random`	500	100	0.0	6.0	8.0	2.0	6.0	0.32
`random`	1000	50	0.0	2.0	4.0	0.0	10.0	0.72
`random`	2000	50	0.0	4.0	6.0	0.0	8.0	0.61
`random`	5000	50	0.0	2.0	4.0	0.0	4.0	0.89
`random`	10000	50	0.0	4.0	6.0	0.0	6.0	1.02
`gp`	500	100	0.0	5.0	10.0*	3.0	10.0*	0.46
`gp`	1000	50	0.0	2.0	12.0*	0.0	10.0	2.33
`gp`	2000	50	0.0	6.0	22.0*	4.0	14.0*	2.99
`gp`	5000	50	0.0	20.0*	42.0*	14.0*	18.0*	7.94
`gp`	10000	50	0.0	26.0*	64.0*	18.0*	22.0*	10.10
`llm`	500	100	0.0	12.0*	26.0*	10.0*	13.0*	1.08
`llm`	1000	50	0.0	14.0*	46.0*	20.0*	14.0*	3.16
`llm`	2000	50	0.0	34.0*	70.0*	38.0*	28.0*	4.35
`llm_shuffled`	500	100	0.0	2.0	3.0	1.0	2.0	0.19
`llm_shuffled`	2000	50	0.0	2.0	2.0	0.0	6.0	–
`llm_zeroshot`	500	100	0.0	0.0	1.0	1.0	1.0	0.19
`llm_zeroshot`	2000	50	0.0	0.0	0.0	0.0	2.0	–

Table 7: Empirical size against search budget on the within-date null (CSI 300, $R = 50$ above budget 500). The practitioner's rule—IC ≥ 0.02, $|t| \geq 2$, Sharpe ≥ 1—is the row a practitioner actually applies; N_{eff}/M is the effective trial count per unique candidate against the control's null singles at the same budget. Nominal 5%; layout as Table 1.

candidate quantity	prediction	`gp`	`llm`	verdict
coupling to the incumbent (corr. of each proposal with the current best, null markets, mean over replicates)	higher for the arm that breaks	0.092	0.077	does not separate; monotone with size only within one simulated kernel family
participation ratio of the realised set (null markets, median over replicates)	higher PR breaks the bootstrap	9.0	19.1	separates these two arms, but the simulator breaks SPA hardest at $\text{PR} = 1.5$ (Table 6); not the axis
N_{eff}/PR, implicit over visible multiplicity (real market, both-signs matrix)	monotone with size	12.1	102.8	orders both simulated families (Figure 4); on the real arms it separates `llm` from every other arm by 4.6–13.8× but does not order the low end (`random` 22.4, `llm_zeroshot` 15.0)
off-book screening: N_{eff}/M against each arm's own round-0 reference	≈1 for `gp`; ≫ 1 for `llm`	0.48	0.80	ordering holds (control 0.27); levels do not—both below one for the reason in Section 4.3

Table 8: Candidate explanations for why the two adaptive arms differ. Each row's prediction was fixed before its measurement. Participation ratios are of the both-signs daily-return matrix that SPA consumes; N_{eff} in row 3 uses the shared `random` reference of Section 4.3, in row 4 each arm's own round-0 candidates.

return matrix lies on its first principal component (participation ratio 6.2 and 6.4); on panel 1086 the candidates over the line load on that component (mean absolute loading 0.073) and the rest essentially do not (0.008), whereas on panel 1000 the split is faint (0.061 against 0.044). The factor need not

itself clear the line: the $|t|$ of its return series is 1.83 on panel 1086 against 1.12 on panel 1000. It carries every candidate that loads on it most of the way, and their idiosyncratic parts decide, one by one, which of the cluster's members straddle 2—so a draw that is loud for the factor is loud for half the candidate set at once. Because every arm is evaluated on the same 400 permutations, the same few draws inflate every arm's Bonferroni and Benjamini–Hochberg counts together, and the arm contrast in Table 1 is paired by panel. For SPA, which sees $\hat{\Sigma}$, the overlap is small—`llm` and `random` both reject on only seven of the 400 panels—and the exact paired test returns the verdicts the intervals suggest: `llm` against `llm_shuffled` differs on both nulls ($p = 1.7 \times 10^{-7}$ within-date, $p = 3.3 \times 10^{-14}$ covariance-preserving), and `llm` against `random` on both as well ($p = 9.6 \times 10^{-5}$ and $p = 3.8 \times 10^{-13}$). At the $R = 100$ of an earlier run the within-date contrast was not significant ($p = 0.14$); quadrupling the replicates resolved it in the direction the intervals already indicated, which is the resolution $R = 400$ was bought for.

Two of the four candidates are excluded. Coupling to the incumbent is the same for both arms and is therefore not what separates them, although within a single simulated kernel family size is monotone in it (Figure 3). The same measurement confirms Definition 1 on real data: on null markets the coupling of `llm_shuffled` and `llm_zeroshot` to the incumbent is 0.030 and 0.031, against 0.028 for uniform draws and 0.077–0.092 for the two arms that receive paired scores. Destroying the pairing returns the kernel to the non-adaptive floor. The participation ratio of the realised set does separate the two arms on null markets, and within the `llm` arm a replicate's participation ratio predicts its own SPA rejection after conditioning on how many of its candidates cleared $|t| \geq 2$ (partial Spearman of the participation ratio with the SPA p-value given the count, -0.22, $p = 0.028$), whereas for the other arms only the count predicts; but the simulation breaks SPA hardest at a participation ratio of 1.5, so a low participation ratio does not protect a search and a high one is not required for failure. What separates searches in the simulation is N_{eff}/PR, the multiplicity the search ran over the multiplicity the bootstrap sees. The remaining candidate is *off-book screening*—the number of candidates a kernel considers for each it emits. A language model cannot score anything on D; what it can do is reason over the table it is shown, and propose variants of the formulas that scored well. That is the mechanism of Proposition 1's construction, exploration anchored on the selected best, and it does not need hidden evaluations: the variants are reported, and the bootstrap sees them, anchored on the wrong column. Genetic mutation does the same thing with a different proposal distribution, which is why the proposition does not predict which of the two arms breaks harder. What would separate them is how many variants each emits per anchor and how tightly they track it. The test in the table is indirect: N_{eff} for each arm computed against that arm's own pre-feedback candidates as the reference, which should sit near M for a kernel whose variants add nothing and well above it for one whose variants carry the anchor's selection. The test returns the ordering and not the levels. Against its own round-0 reference `gp` has $N_{\text{eff}}/M = 0.48$ and `llm` 0.80, with the control at 0.27 under the same construction; both sit below one because the candidates of any one grammar are dependent (Section 4.3), so the absolute prediction was wrong in the way Section 4.3 explains and the relative one holds. Off-book screening therefore survives as the ordering of the two arms, not as a count of hidden evaluations. The same ordering appears in N_{eff}/PR on the real market, which separates `llm` from every other arm while leaving the low end unresolved (Table 8).

H.5 Power, extended

Section 5.2 before condensing (moved verbatim). A valid test that accepts nothing is not a remedy, so we measure what null calibration costs in power. On synthetic panels with planted alphas of known strength we run the same searches, apply every acceptance rule, and score each accepted candidate against an oracle: an independent panel from the same generator with fresh noise and ten times the length, so that any candidate carrying a material share of a planted signal is detected on it and the verdict approximates the generator's own truth. A candidate is a true discovery if its IC on the oracle panel is significant with the same sign. Requiring the accepted expression to match a planted one would be too strict, since searchers legitimately rediscover a signal in another form; a correlation threshold would relocate the arbitrariness into the threshold. Two things about what is measured. The false discovery rate reported here is a measured rate, not a controlled level: the calibrated hurdle of Section 4 controls the probability that any null candidate clears it, a family-wise guarantee, and says nothing per candidate. And the arm is the non-adaptive one, on which Benjamini–Hochberg is valid, so what the comparison measures is each rule's behaviour when signals are weak and dense rather

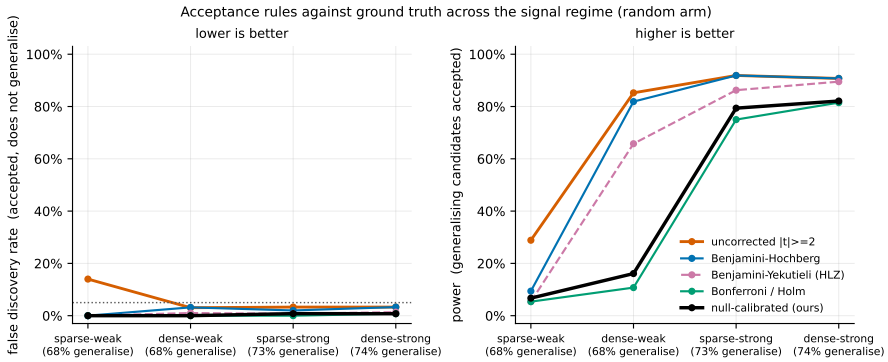


Figure 5: False discovery rate and power of each acceptance rule against ground truth on a ten-times-longer oracle panel, across four signal regimes ordered by planted signal (non-adaptive arm). Every corrected rule's FDR is at or near zero on this arm, where all of them are valid; the rules differ in power, with null calibration in Bonferroni's class.

than a repair of a broken procedure. The same comparison on the adaptive arm, which is where the repair would be visible, is not reported.

No single amount of planted signal is representative: at five planted alphas up to IC 0.08 most candidates genuinely generalise and every rule performs well; at two alphas at IC 0.01–0.02, the corrected rules accept 8–14 candidates and reject almost everything else. Figure 5 therefore reports FDR and power as curves across four regimes ordered by planted signal. Truth is $|t| \geq 2$ on the oracle's 15,000 days, an IC of about 0.0005; because every candidate is built from the same price and volume fields as the planted signals, about two-thirds of the candidates in every regime clear it, and a rule's false discovery rate is the share of what it accepts that carries no effect at all. The length of the oracle decides what is measured. With an oracle of the search panel's length, an earlier version of this analysis read an 80% false discovery rate for Benjamini–Hochberg on the non-adaptive arm, where the procedure is valid; that number was the oracle's own miss rate, not the rule's. With the ten-times oracle, in the dense-weak regime—five weak signals—the non-adaptive arm's false discovery rate is at or near zero for every rule (uncorrected 3%, Benjamini–Hochberg 3%, Benjamini–Yekutieli 1%, Bonferroni and null calibration 0%), as validity on that arm requires, and the rules differ in power alone: uncorrected 85%, Benjamini–Hochberg 82%, Benjamini–Yekutieli 66%, null calibration 16%, Bonferroni 11%. On the evolutionary arm the ordering is the same (95%, 95%, 93%, 66%, 67%). Null calibration is Bonferroni's class in power, and that is the price of the guarantee it carries: it controls the probability that any null candidate clears the hurdle for an arbitrary searcher, which the corrections do only for a fixed design. The other regimes say the same thing at different amplitudes (Figure 5): in sparse-weak, where two signals at IC 0.01–0.02 leave every rule little to find, the corrected rules accept 8–14 candidates with no false discovery and power of 5–9% (null calibration 7%); in the two strong regimes every rule's power is between 75% and 92% with FDR at most 3.3%, and null calibration trails Benjamini–Hochberg by 5 to 12 points, the narrowest the gap gets, because a real signal clears the hurdle easily. What separates the rules is the weak, dense regime, and there null calibration pays for its guarantee. The comparison that would show the repair most directly—the same power analysis on the adaptive arm, where Benjamini–Hochberg is invalid rather than merely conservative—is not reported here. What we measure is the price of null calibration against rules that are working, which is the harder case for it: on the arm where the alternatives are broken, a valid rule can only look better by comparison.

H.6 The second domain, extended

The stopping rule, and what it says before any statistic. A search stops when a round returns no new parseable candidate, and on this grammar the rule fired on 6 of 303 first-pass runs—five of them in the `llm` arm and none in the zero-shot arm, the real `llm` run stopping at 179 of 400. Shown its own scores, the model converges on a few forms. Such runs are analysed as they ended. Benjamini–Hochberg gives the same ordering as SPA (10% for `llm`, paired against the control

arm	adaptive	R	$\lvert t\rvert \geq 2$	Bonf.	Holm	BH	BY	RC	SPA
`random`	no	100	99.0*	2.0	2.0	2.0	0.0	2.0	2.0
`gp`	yes	100	96.0*	4.0	4.0	9.0	1.0	5.0	5.0
`llm`	yes	100	99.0*	4.0	4.0	10.0*	1.0	12.0*	12.0*
`llm_shuffled`	no (unpaired)	100	100.0*	4.0	4.0	4.0	0.0	5.0	5.0
`llm_zeroshot`	no	100	99.0*	4.0	4.0	5.0	0.0	1.0	1.0

Table 9: Empirical size on the symbolic-regression domain, where the null permutes the target across samples and nothing of the market remains (Section 5.1). Same five arms, same statistics, same nominal 5%; layout as Table 1.

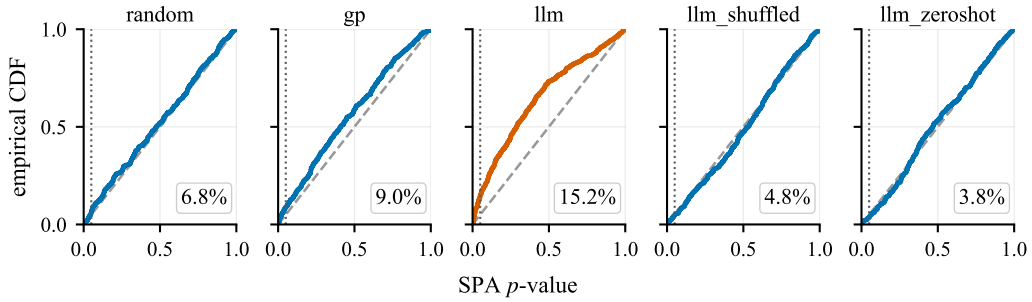


Figure 6: Hansen's SPA p-value across the 400 within-date-permutation replicates of Table 1, one panel per arm: empirical CDF against the $U[0,1]$ diagonal, with the dotted line at $\alpha = 0.05$. A valid test lies on the diagonal; one that over-rejects bows above it at the left, as `llm` does. The number in each panel is that arm's empirical size at 5%. Kolmogorov–Smirnov p-values against uniformity are in Table 1.

$p = 0.02$). A robustness set conditioned on reaching the full budget (six fresh draws over the 297 shared files) moves nothing beyond one replicate: `llm` SPA 11% [6.3, 18.6], BH 11%, Bonferroni 3%; paired against zero-shot $p = 0.002$, against `random` $p = 0.023$, against `llm_shuffled` $p = 0.18$.

Section 5.2, transfer and early-stop paragraphs (moved verbatim). The apparatus transfers unchanged: the score is a correlation whose per-sample series is what SPA and CSCV consume, and the null permutes the target. What the domain removes is every explanation specific to markets—factor structure, cross-sectional premia, serial dependence, the alpha prompt's calibration advice. If the departure of the feedback arm from nominal is a property of the loop rather than of finance, it should reappear here with the same two controls at nominal.

One property of the feedback loop shows up here before any statistic is computed. A search stops when a round returns no new parseable candidate, a rule that is part of the procedure and applies to real and null runs alike. On this grammar it fired on 6 of 303 first-pass runs, five of them in the `llm` arm and none in the zero-shot arm, and the real `llm` run itself stopped at 179 of its 400 evaluations: shown its own scores, the model converges on a few forms and stops producing new ones, which the same model without the feedback table never does. Runs that stopped are analysed as they ended; replacing them with fresh draws would condition the null set on reaching the budget, and the replicates most like the real run are exactly the ones that stopped (Appendix G).

H.7 Real markets, extended

The holdout, in full. Of the candidates clearing each arm's own null-calibrated hurdle on period A, the fraction still significant at $\lvert t\rvert \geq 2$ on period C:

arm	accepted on A	still $\lvert t\rvert \geq 2$ on C	fraction	sign kept
`random`	84	57	0.68	0.97
`gp`	130	107	0.82	1.00
`llm`	352	325	0.92	0.99
`llm_shuffled`	273	189	0.69	0.92
`llm_zeroshot`	297	257	0.87	0.96

The permutation null, as stated before condensing (moved verbatim). Table 1 reports, for each arm, the fraction of 400 permutation-null replicates on which each procedure made at least one discovery. The non-adaptive control comes in at nominal on every row: Hansen's SPA 6.8% (Wilson 95% interval $[4.7, 9.6]$), Bonferroni 4.0%, Benjamini–Hochberg 5.8% ($[3.9, 8.5]$). The two arms that are non-adaptive under Definition 1 but run the identical loop—`llm_shuffled`, which receives every score with the pairing destroyed, and `llm_zeroshot`, which receives none—are at or below nominal on every row as well (SPA 4.8%, $[3.1, 7.3]$, and 3.8%, $[2.3, 6.1]$). The evolutionary arm `gp` sits at 9.0% on SPA ($[6.6, 12.2]$), an interval that excludes 5% by a point and a half, and at 12.8% on Benjamini–Hochberg; its p-values are not uniform (Kolmogorov–Smirnov $p = 5.4\times10^{-4}$). Two arms therefore depart from nominal, one mildly and one strongly. The strong one is `llm`, the arm whose proposer reads its own backtest results, at 13% on SPA ($[7.8, 21.0]$), 12% on Bonferroni ($[7.0, 19.8]$), 10% on Benjamini–Yekutieli ($[5.5, 17.4]$) and 26% on Benjamini–Hochberg ($[18.4, 35.4]$). Because every arm is run on the same 400 null panels, the arm contrast is paired by panel, and the exact paired test (Table 1, caption) is the right comparison. On this null it separates `llm` from its two causal controls—from `llm_shuffled` at $p = 0.003$ and from `llm_zeroshot` at $p = 0.002$—but not from `random` ($p = 0.14$): the feedback arm and the non-adaptive control reject on only one shared panel, so the paired test is close to the unpaired one, and the evidence that `llm` departs from nominal on the within-date null rests on its own interval and on the uniformity test rather than on a head-to-head with `random`. The head-to-head is decisive on the simulated null.

Why the bootstrap is correct for some arms and not others (moved verbatim). The bootstrap that Hansen's test performs—resample the rows of the evaluation set, hold the realised candidate columns fixed—produces a correct null for every searcher that did not choose its columns using those rows, and an incorrect one for the searchers that did, in proportion to how much they did.

The simulated null, as stated before condensing (moved verbatim). On the simulated null market (Table 10), the control is nominal on the per-candidate procedures (Bonferroni 5.0%, Benjamini–Hochberg 6.2%) but mildly above on the bootstrap tests (SPA 7.8%, $[5.5, 10.8]$), and so are the two unpaired arms (`llm_shuffled` SPA 7.5%, `llm_zeroshot` 7.8%). No arm's SPA p-values are uniform under this construction (Kolmogorov–Smirnov $p = 2.3 \times 10^{-4}$ for `random`, 1.0×10^{-6} and 3.5×10^{-7} for the unpaired arms), so the permutation null of Table 1 is the only construction on which every control passes its own uniformity test, and on this one it is the arm contrast rather than the absolute level that carries the result (Appendix E). Meanwhile `llm` breaks every correction except the deflated Sharpe ratio: Bonferroni 14.8% ($[11.6, 18.6]$), Benjamini–Yekutieli 12.2% ($[9.4, 15.8]$), SPA 26.2% ($[22.2, 30.8]$), Benjamini–Hochberg 36.0% ($[31.4, 40.8]$). Because the inflation is common to every arm and all five run on the same panels, it cancels in the paired test, which is decisive: `llm` against `random` at $p = 3.8 \times 10^{-13}$ and against `llm_shuffled` at $p = 3.3 \times 10^{-14}$—92 and 90 panels on which `llm` rejects and the control does not, against 18 and 15 the other way—while `gp` against `random` is $p = 0.002$. The deflated Sharpe ratio, told the true trial count $2M$, is conservative on every arm and every null.

The acceptance funnel, as stated before condensing (moved verbatim). On the real market the uncorrected rule $|t| \geq 2$ accepts 115 of `random`'s 283 unique candidates and 408 of `llm`'s 484; the practitioner rule, which adds IC ≥ 0.02 and an annualised Sharpe of at least one, accepts 29 and 89. Against each arm's own null-calibrated hurdle at $R = 100$—the 95th percentile of the null maximum, 3.76 for `random` and 4.10 for `llm`—84 and 352 survive. The effective trial count per unique candidate, N_{eff}/M, is 0.32 for `random`, 0.46 for `gp`, 1.08 for `llm`, and 0.19 for both unpaired arms. These ratios are an ordering, not a scale: the estimator's absolute level moved with the reference (the `random` anchor read 1.07 at $R = 30$ and 0.32 at $R = 100$, Section 4.3), so what the funnel establishes is that the paired-feedback arm carries more implicit multiplicity per candidate than the non-adaptive search, which carries more than the same loop with feedback destroyed. Every

procedure-level p-value is at its floor of $1/(R+1) = 0.0099$; the probability of backtest overfitting, computed on both signs of every candidate, is below 0.02 for every arm.

The seed spread, as stated before condensing (moved verbatim). A single run of a stochastic search is one draw, so the comparison between `llm` and `llm_shuffled` in Table 11 is repeated at five seeds per arm. The seeds are independent draws in the relevant sense: the pairwise Jaccard overlap of the candidate sets they propose is 0.045 for `llm`, 0.033 for `llm_shuffled` and 0.211 for `llm_zeroshot`, the last being higher because a prior with no feedback to move it resamples its favourites. The maximum $|t|$ is not the statistic to compare on. Three of five `llm_zeroshot` seeds and two of five `llm_shuffled` seeds return an identical value of 13.53, because the same expression, `ts_rank(volume, 10)`, wins each time—it is also `random`'s winner on this market, and `gp`'s winner is another volume expression at a nearly identical $|t|$ of 13.66. An extreme value saturates at whatever the prior's favourite formula scores and carries no information about the search.

The count of candidates above each arm's own null-calibrated hurdle does. At the 95th percentile of the null maximum over 100 replicates (hurdles 4.10, 3.56 and 3.47 for the three arms), `llm` retains 373 ± 45 candidates per seed $(352, 429, 405, 315, 364)$, `llm_shuffled` 212 ± 50 $(273, 199, 203, 141, 243)$ and `llm_zeroshot` 286 ± 9. Every feedback seed exceeds every shuffled seed (minimum 315 against maximum 273); the Mann–Whitney p-value is 0.0079, the smallest attainable at five against five, and Welch's t gives 7×10^{-4}. The same separation holds on the uncorrected count $|t| \geq 2$ (429 ± 33 against 325 ± 38). The gap is visible in the formulas themselves: on four of five seeds the feedback arm's winner composes the prior's volume term with a short-horizon price-change term—`add(rank(delta(open, 5)), ts_rank(volume, 3))` and its variants—at $|t|$ between 16.1 and 16.9, where the prior alone reaches 13.5. The loop's gain is reproducible; the one seed that missed it (13.10) is why the spread on the maximum is four and a half times the shuffled control's (standard deviations 1.49, 0.33 and 0.66).

The shift null, as stated before condensing (moved verbatim). Table 12 repeats the permutation picture for the per-candidate procedures on a null that keeps every serial dependence of the real panel: the control sits at nominal (Bonferroni 6%, Benjamini–Hochberg 6%, Benjamini–Yekutieli 0%), the two unpaired arms at 5–7% on Bonferroni and 10–11% on Benjamini–Hochberg (a mild departure with dependent offsets), and `llm` departs further than on the permutation null—24% on Bonferroni ($[16.7, 33.2]$), 62% on Benjamini–Hochberg ($[52.2, 70.9]$), 22% on Benjamini–Yekutieli, against 12.5%, 28.5% and 8.5% in Table 1—with `gp` in between (7%, 18%, 4%). Because the offsets are dependent, these percentages rest on far fewer than 100 effective draws and are read as an ordering, not a scale. The bootstrap and Sharpe-based tests cannot be read on this construction at all: with a fixed feature–return pairing, any feature whose cross-sectional ranking is persistent selects a nearly fixed basket of stocks, and a fixed basket carries the other era's idiosyncratic premium, so their long-short returns have non-zero mean under the null (SPA rejects on 96% of the control's replicates; a fixed relabelling of stocks does not remove it, Appendix E). The within-date permutation re-draws the basket every day, which is why it is exact for portfolio statistics and why the shift null is a robustness check for the per-candidate procedures only.

The second universe, as stated before condensing (moved verbatim). The second universe is a replication, not a transfer. Evaluating CSI 300 survivors on CSI 500 would confound whether the phenomenon reproduces with whether particular alphas generalise, so the full pipeline was run again from scratch on CSI 500 period A with the same five arms, the same budget and $R = 100$ permutation nulls (Table 13). The non-adaptive control is at or below nominal on every row (Hansen's SPA 5%, Bonferroni 1%, Benjamini–Hochberg 3%), `llm_zeroshot` likewise (5%, 0%, 4%). `llm` departs on every correction except the deflated Sharpe ratio, and by almost exactly the CSI 300 margins: SPA 13%, Bonferroni 15%, Benjamini–Yekutieli 9%, Benjamini–Hochberg 26% (against 13, 12, 10 and 26 on CSI 300 at the matched $R = 100$, which is the comparison to make here; Table 1 reports CSI 300 at $R = 400$, where the same four cells are 15.2, 12.5, 8.5 and 28.5). Two differences are worth stating rather than smoothing, and both weaken what the second universe establishes. First, `llm_shuffled`, which was at 2% on CSI 300, is at 10% on SPA here, an interval $[5.5, 17.4]$ that excludes 5%, with non-uniform p-values (KS $p = 0.0013$); the evolutionary arm is likewise non-uniform (8%, KS $p = 0.0003$). Only `random` (5%, KS 0.47) and `llm_zeroshot` (5%, KS 0.17) are cleanly at nominal on this universe. Second, the paired test over the 100 shared panels does not separate `llm` from `llm_shuffled` here ($p = 0.63$; against `random` $p = 0.06$ and against

arm	adaptive	R	$\lvert t\rvert \geq 2$	practitioner	Bonf.	BH	BY	RC	SPA	DSR	median p_{SPA}	KS p vs $U[0,1]$
`random`	no	400	100.0*	0.0	5.0	6.2	1.2	7.5*	7.8*	2.8	0.411	**2.3e-04**
`gp`	yes	400	99.2*	0.0	7.0	13.8*	2.8	13.8*	14.2*	1.2	0.316	**4.8e-14**
`llm`	yes	400	99.5*	0.0	14.8*	36.0*	12.2*	26.0*	26.2*	0.2	0.182	**6.0e-44**
`llm_shuffled`	no (unpaired)	400	99.8*	0.0	2.8	4.0	1.0	7.5*	7.5*	1.0	0.376	**1.0e-06**
`llm_zeroshot`	no	400	99.0*	0.0	2.8	5.2	1.0	6.8	7.8*	1.5	0.406	**3.5e-07**

Table 10: Empirical size under the simulated null, which preserves the cross-sectional covariance and fat tails of the real panel and removes every feature–return link. Matched $R = 400$, nominal 5%; layout as Table 1. Every arm including the non-adaptive controls is mildly above nominal on the two bootstrap tests here, so on this null it is the arm contrast rather than the absolute level that carries the result.

arm	cand.	$\lvert t\rvert \geq 2$	practit.	Bonf.	BH	BY	null-cal.	N_{eff}/M [95% CI]
`random`	283	115	29	84	110	93	84	0.32 [0.25, 0.39]
`gp`	202	156	71	130	153	136	130	0.46 [0.37, 0.55]
`llm`	484	408	89	363	404	382	352	1.08 [0.57, 2.37]
`llm_shuffled`	498	379	27	249	376	334	273	0.19 [0.14, 0.22]
`llm_zeroshot`	489	363	56	278	360	323	297	0.19 [0.16, 0.21]

Table 11: Acceptance funnel on the real market (CSI 300, period A): candidates accepted by each rule, and the effective trial count relative to unique candidates with its bootstrap interval. The practitioner rule is IC ≥ 0.02, $|t| \geq 2$, annualised Sharpe ≥ 1.

`llm_zeroshot` $p = 0.10$), where on CSI 300 it did. What CSI 500 replicates is therefore the marginal result—the feedback arm departs from nominal, by the CSI 300 margins, on a universe it never saw—and not the causal contrast against the shuffled control, which is established on CSI 300 and on the simulated null. The effective-trial ordering is unchanged: N_{eff}/M of 2.54 for `llm` against 1.06 for `gp`, 0.63 for `random`, 0.49 for `llm_shuffled` and 0.19 for `llm_zeroshot`. CSI 500 is a small-cap universe with far more raw cross-sectional predictability than CSI 300—the uncorrected rule accepts 248 of `random`'s 279 candidates on the real market against 115 of 283—and the size distortion of the feedback arm is the same size on both.

The excluded block permutation, as stated before condensing (moved verbatim). A third construction, block permutation of the date index, preserves serial dependence by resampling contiguous blocks of dates, and fails the non-adaptive control as badly as every other arm (Hansen's SPA 100%, Bonferroni 54%). It is not a null. A block whose offset falls inside a feature's lookback window hands the candidate a "forward" return the feature already contains, which is a genuine feature–target relation on a panel labelled null; it hits every arm equally, and no bootstrap survives a leak. A second defect compounds it: the block structure induces long-range serial dependence in the daily IC series that an i.i.d. standard error ignores, so that a Newey–West standard error with twenty lags brings the non-adaptive arm's $|t| \geq 2$ rate from 27% to 4% while leaving the real market's unchanged (40.0% to 40.5%)—the real market's own IC autocorrelation, lag-1 0.07, is small enough that the i.i.d. statistic is adequate there.

H.8 Limitations, in full

Limitations, as stated before condensing. Proposition 1 is an existence result: the standard corrections carry no guarantee under adaptivity, not that every adaptive searcher breaks them, and the evolutionary arm is at most weakly affected at budget 500, though it breaks Benjamini–Hochberg outright by budget 10,000 (Table 7). The real-data evidence rests on one grammar and one model family, and the second market, domain and continent are robustness checks rather than independent replications. Two qualifications carry particular weight. Under the shift null the portfolio-return tests cannot be read at all, because a persistent characteristic selects a nearly fixed basket carrying the other era's premium and SPA then rejects on 96% of the *control's* replicates. And on CSI 500 the shuffled control is itself mildly non-nominal (SPA 10%) and the paired test does not separate it from `llm` ($p = 0.63$), so that universe replicates the marginal result but not the causal contrast. We did not run the two adaptive-data-analysis mechanisms on the language-model arm, and we attempted two larger models without success: one was not servable on the software stack every other number

arm	adaptive	R	$\lvert t\rvert \geq 2$	practitioner	Bonf.	BH	BY
`random`	no	100	100.0*	0.0	6.0	6.0	0.0
`gp`	yes	100	97.0*	0.0	7.0	18.0*	4.0
`llm`	yes	100	100.0*	0.0	24.0*	62.0*	22.0*
`llm_shuffled`	no (unpaired)	100	100.0*	0.0	5.0	11.0*	0.0
`llm_zeroshot`	no	100	98.0*	0.0	7.0	10.0*	0.0

Table 12: Empirical size of the per-candidate procedures under the `shift` null, which preserves all serial dependence in both features and returns by re-aligning the return panel forward in time by a single offset $o \in [231, 483]$ trading days, larger than any lookback window, with no wrap-around (Section 4.1, Appendix E). Offsets are dependent across replicates, so $R = 100$ shift draws are far fewer effective replicates than $R = 100$ permutations. The portfolio-return tests (Reality Check, SPA, deflated Sharpe) are omitted: under a fixed feature–return pairing a persistent characteristic defines a nearly fixed basket that carries the other era's premium, so they reject on 96% of the control's replicates (Appendix E). Nominal 5%; layout as Table 1.

arm	adaptive	R	$\lvert t\rvert \geq 2$	practitioner	Bonf.	BH	BY	RC	SPA	DSR	median p_{SPA}	KS p vs $U[0,1]$
`random`	no	100	100.0*	0.0	1.0	3.0	0.0	5.0	5.0	3.0	0.443	0.474
`gp`	yes	100	97.0*	0.0	4.0	11.0*	1.0	8.0	8.0	2.0	0.355	**2.6e-04**
`llm`	yes	100	100.0*	0.0	15.0*	26.0*	9.0	13.0*	13.0*	1.0	0.221	**1.7e-08**
`llm_shuffled`	no (unpaired)	100	100.0*	0.0	7.0	7.0	1.0	10.0*	10.0*	3.0	0.426	**0.001**
`llm_zeroshot`	no	100	99.0*	0.0	0.0	4.0	0.0	5.0	5.0	1.0	0.429	0.170

Table 13: Empirical size on CSI 500, the second universe, under the permutation null at matched $R = 100$; layout as Table 1. The entire pipeline—search, null calibration, correction—was run independently on CSI 500 period A; no CSI 300 candidate was transferred.

was produced on, and the other's outputs failed the grammar's syntax asymmetrically between arms, which would have made a scale comparison a syntax comparison. The remaining limitations follow.

Of the candidate mechanisms in Table 8, two are excluded and two hold only as orderings of the arms rather than as the monotone relations predicted, so the question of *which* adaptive searchers break which corrections is answered empirically and not yet mechanistically.

The smallest attainable calibrated p-value is $1/(R+1)$, so at $R = 100$ we do not distinguish $p = 0.005$ from $p = 0.01$. Because every arm is calibrated on the same 100 panels, the arm-wise Wilson intervals in Table 1 are not independent of one another; this is why the arm contrasts are reported through the exact paired test on the per-panel rejection indicators rather than by comparing intervals.

The within-date permutation destroys the serial dependence of the daily IC series, so the t-statistic's i.i.d. standard error is exact on that null but not on the real market. On the real market the median ratio of the i.i.d. to the Newey–West t-statistic across the control's candidates is 1.05, with a lag-1 autocorrelation of the IC series of 0.07, so the inflation is small; the forward-shift null preserves that dependence and reproduces the ordering.

The multi-seed evidence uses five seeds per arm, which is the smallest number at which the Mann–Whitney test can reach $p < 0.01$. The period boundaries were re-specified once, before any holdout evaluation, because the public data source ends earlier than originally assumed.

Corollary 1: derivation and numerical check. Take the construction of Proposition 1 with $N = K$, so $M = 2K$, and write $Z_k = S_D(g_k)$ and ε_n for the variants' independent parts, all standard normal under H_0. Extreme-value theory for the Gaussian maximum gives $\max_{k \leq K} Z_k = \sqrt{2 \log K}\,(1 + o_P(1))$ and likewise for $\max_{n \leq N} \varepsilon_n$.

What the search realises. The block is anchored on the selected representative, whose score *is* the maximum of the K draws, so

$$\max_{i \leq M} S_D(f_i) \;\geq\; \rho \max_{k \leq K} Z_k + \sqrt{1-\rho^2} \max_{n \leq N} \varepsilon_n \;=\; \big(\rho + \sqrt{1-\rho^2}\big)\sqrt{2 \log K}\,(1 + o_P(1)).$$

What a fixed-design correction computes. Any critical value that is a functional of $(M, \hat{\Sigma})$ takes as its null the maximum of a mean-zero Gaussian field with the realised covariance. That field has K independent coordinates and one block of N loading ρ on a *fresh* coordinate $Z \sim N(0,1)$, because resampling the rows re-draws the parent rather than re-selecting it. Its maximum is

$$\max\Big\{\sqrt{2\log K},\ \rho Z + \sqrt{1-\rho^2}\sqrt{2\log N}\Big\}(1+o_P(1)) \ = \ \sqrt{2\log K}\,(1+o_P(1)),$$

since $\sqrt{1-\rho^2} < 1$ and $Z = O_P(1)$. The $(1-\alpha)$ quantile adds $O(1/\sqrt{2\log K})$, the Gumbel scale.

The gap. Dividing, the ratio of realised statistic to critical value tends to $c(\rho) = \rho + \sqrt{1-\rho^2}$, which exceeds 1 on $(0,1)$ and is maximised at $\rho = 1/\sqrt{2}$ with $c = \sqrt{2}$. The difference is $(c(\rho) - 1)\sqrt{2\log M}(1+o_P(1))$ and the null's fluctuations are $O_P(1/\sqrt{2\log M})$, so the rejection probability tends to one.

Numerical check (`tools/verify_budget_corollary.py`, 4,000 replicates per cell, output in `results/budget_corollary.txt`):

K	10	10^2	10^3	10^4
size, $\rho = 0.5$	0.14	0.46	0.87	1.00
size, $\rho = 1/\sqrt{2}$	0.16	0.58	0.98	1.00
size, $\rho = 0.9$	0.15	0.49	0.89	1.00

Convergence of the *ratio* to $c(\rho)$ is slow—at $K = 10^4$ and $\rho = 1/\sqrt{2}$ the measured ratio is 1.23 against an asymptotic 1.414—because the correction is $O(1/\log K)$. The *size*, which is what matters, is already at 1. Two honest caveats: the result is asymptotic in K and stated for this cluster family under Gaussian scores, so it bounds what a fixed-design correction can promise rather than describing every searcher; and a real grammar's cluster structure is neither exactly this nor stationary in the budget, which is why Section 5.3 measures the sweep instead of predicting it.

H.9 Buying the hurdle with fewer runs

That, however, is a fact about the *empirical quantile* rather than about the information in R draws. Each replicate contributes a maximum over many dependent candidate scores, and such a maximum is asymptotically Gumbel, so a two-parameter fit uses the shape of the whole sample instead of only its top order statistic. Substituting a moment-matched Gumbel fit changes the arithmetic (held-out size, mean over four arms on the US panel, $4{,}000$ random subsets per cell):

null markets used to calibrate	10	20	30	50	75
empirical quantile	11.9%	9.1%	7.6%	6.6%	6.0%
Gumbel fit	6.9%	5.6%	5.1%	4.7%	4.6%

On this panel the Gumbel hurdle at $R = 20$ is better calibrated than the empirical quantile at $R = 100$, at a fifth of the cost. It buys that by being slightly conservative—its hurdle sits above the reference and it accepts fewer candidates, $72 \to 66$ on the evolutionary arm and $48 \to 38$ under the reusable holdout, with the ladder arm unchanged at 24.

On CSI 300 the improvement over the empirical quantile holds on every arm and every budget—at $R = 20$ the Gumbel fit's held-out size is 4.5–7.5% against the empirical quantile's 8.9–9.4%—but its absolute calibration depends on the arm. It is conservative for the feedback arm (4.5% at $R = 20$, 4.0% at $R = 30$) and runs one to two points liberal for the non-adaptive control and the evolutionary arm (7.0% and 6.0% at $R = 30$, where the empirical quantile at $R = 100$ gives 6.0%). The fit is only as good as the match between its tail and each arm's, and candidate sets that differ in how concentrated they are have tails that differ too.

The property that matters for a remedy meant to be applied by someone who does not know how adaptive their searcher is, is that none of this should depend on the searcher. It does not. Across the simulation's adaptivity dial, from a non-adaptive control to a proposer whose coupling to the evaluation data is 0.95, the empirical quantile's held-out size at $R = 20$ is 8.8–9.4% and the

Gumbel fit's is 4.5–5.0%; neither moves with adaptivity at any budget, because the rule uses only the exchangeability of the null draws and never the searcher.

Gumbel is, strictly, the wrong family. With 300 replicates the fit is marginally rejected on the four adaptive cells (Kolmogorov–Smirnov $p = 0.044$–0.059) and not rejected on the non-adaptive one ($p = 0.18$), and a three-parameter GEV fits far better ($p \geq 0.90$ throughout) with a shape near -0.2: the null maxima have a *bounded* upper tail. That is exactly why the two-parameter fit errs safely—forcing an unbounded tail onto a bounded one over-estimates the upper quantile—and it makes the obvious improvement worth testing. It does not work. Fitting the correct family is worse at every budget we would use: at $R = 20$ the GEV's held-out size is 7.3–8.0% against Gumbel's 4.5–5.0%, and at $R = 10$ the GEV hurdle's standard deviation across subsets is 0.7 to 1.8 on three of the five adaptivity levels and diverges by six orders of magnitude on the other two, against Gumbel's 0.40 at every level. The third parameter has to be estimated from the same ten or twenty points, and on a bounded tail it is barely identifiable there, so the variance it adds costs more than the bias it removes. The deliberately mis-specified estimator is the better one here, and it is conservative when it is wrong.

What survives every market and arm is the comparison at equal cost: at any budget below 50, the Gumbel fit is closer to nominal than the empirical quantile, usually by several points. What does not survive is a promise of exact nominal size at small R, since that depends on how well each arm's tail matches the family. We therefore recommend the Gumbel fit as the default at $R = 20$–50, with the caveat that a practitioner who needs size within a point of nominal on an unfamiliar candidate set should either check the tail or spend the replicates. Every experiment in this paper uses the empirical quantile at $R \geq 100$, because that is what they were run at and we will not re-derive results from an estimator adopted afterwards.

H.10 Passages moved from the main text for length

Remark 3 (Multiplicity is not validity). Replacing M with an effective trial count N_{eff}—the number of independent trials the search "is worth", as the Harvey–Liu–Zhu and deflated-Sharpe families implicitly do—does not rescue the fixed design. The searcher in the proof reports every column: its multiplicity is fully visible to the bootstrap, which computes the right null for those columns anchored on the wrong one. Empirically (Section 5.3) N_{eff} saturates at $3.6\times$ the control while every bootstrap and false-discovery-rate procedure keeps rising, and Bonferroni told N_{eff} in place of M rejects 8.7% where nothing is adaptive (Appendix H.1).

How many null markets are needed is an empirical question, and the answer is more than a practitioner would guess—held-out size against R runs 12.1%, 9.1%, 7.7%, 6.7%, 6.3%, 6.0% at $R = 10$, 20, 30, 50, 75, 100. Ten null runs, the number one would try first, give a rule whose true size is more than double nominal, because the estimated 95th percentile of ten points sits near their maximum and a fresh draw exceeds it about $1/(R+1)$ of the time. The cost of under-calibrating is paid in validity rather than in yield: at $R = 10$ the feedback arm's hurdle is 4.02 ± 0.34 against 4.10 at $R = 100$, while accepted candidates move only from 357 to 352.

Recommended practice. For a discovery loop that scores candidates on a fixed evaluation set and conditions later proposals on the result, a multiple-testing correction computed on the evaluated candidates communicates a guarantee that does not exist (Proposition 1). The alternative is to freeze the procedure by hash, construct R datasets on which the null holds by transformation, run the procedure unchanged on each, and take the $(1-\alpha)$ quantile of the R discovery statistics as the hurdle and (2) as the p-value (Proposition 2)—with a non-adaptive proposer of matched budget that must sit at nominal, and, for a language-model proposer, the shuffled-feedback arm that isolates the information channel. The null runs pack onto CPU nodes since only the proposer needs an accelerator, and a cheaper proxy proposer must not be substituted for them: the guarantee comes from re-running the procedure itself.