Trials-Adjusted Out-of-Sample Gating for LLM-Proposed Trading Strategies

Subomi Olagoke *Preprint (draft), 2026. Correspondence: solagoke21@gmail.com*

Abstract

A language model can propose trading strategies faster than a human can read them. That capability is close to worthless on its own, because the binding constraint in systematic trading was never idea generation; it is not being fooled by ideas that only worked in the past. A strategy selected as the best of many candidates carries selection bias: its in-sample performance is inflated by luck in proportion to how many candidates were tried, and that inflation is invisible to the naive backtest that produced it. This paper describes and evaluates a validation gate built to be un-foolable by construction, and reports what happens when it is pointed at real markets.

The system, ÀYÒ, separates a nondeterministic proposer (an LLM that authors strategies as typed, content-hashed data artifacts, never as executable code) from a deterministic verifier that trades only the frozen artifact. The verifier enforces three defenses against backtest overfitting: strict out-of-sample evaluation via a walk-forward split, a *trials-adjusted* acceptance bar whose required out-of-sample Sharpe rises with the number of candidates tested, and robustness sweeps across rolling windows and parameter settings. We run all three on 2–4 years of real U.S. equity data (Alpaca daily bars, 32–83 symbols, ending 2026-07-17) and report every number the gate produced. A single rule-based strategy selected for its in-sample Sharpe of 3.07 collapsed to an out-of-sample Sharpe of $\mathbf{-4.90}$ and was rejected. A cross-sectional momentum book that appeared to beat buy-and-hold out-of-sample by a Sharpe edge of $+0.13$ in one configuration was exposed by the robustness sweep as a fluke: it beat the benchmark in only 2 of 5 rolling windows and in **0 of 12** parameter settings. Under honest validation, nothing survived — which is the expected and correct result on efficient large-cap markets, and precisely the outcome the gate exists to enforce. We argue that the contribution is not an alpha but a methodology: a reproducible gate that reliably refuses to deploy mirages, and that generalizes to any setting where a nondeterministic proposer must be held to a deterministic guarantee. We are explicit throughout about which claims the evidence supports and which remain open.

1. Introduction

Suppose you can generate trading strategies at will. You describe a market intuition, and moments later you have a fully specified, backtestable strategy with entry rules, exits, and risk parameters. Large language models make this not just possible but cheap. The natural reaction is to generate many strategies, backtest them all, and deploy the best one.

This is exactly the wrong thing to do, and it is the single most common way retail systematic traders lose money. The problem is *selection under multiplicity*. If you test enough strategies on one dataset and keep the best, that winner's backtest performance is inflated by luck, and the inflation grows with the number of candidates tried [BacktestOverfitting; White; STW]. The backtest that selected the winner cannot see its own optimism. A strategy with a beautiful in-sample track record and no real edge is not a rare pathology; on efficient markets it is the *default* output of an unconstrained search.

The asymmetry this creates is the premise of the present work. Generating candidate strategies is easy and getting easier; *not being fooled by them* is the hard part, and it does not get easier when the proposer gets faster. If anything it gets harder, because a faster proposer means more trials, and more trials mean more selection bias. A system that pairs a powerful generator with a weak validator is worse than one with no generator at all, because it manufactures confident, well-dressed mistakes.

ÀYÒ is built around that asymmetry. It treats strategy generation as cheap and untrusted, and concentrates its engineering in a validation layer that a plausible-but-wrong strategy cannot pass. Concretely:

- **The proposer is untrusted and quarantined.** An LLM authors each strategy as a typed data *artifact* — a set of declarative rules — not as code. The artifact is frozen and content-hashed. The model's output never executes; only the deterministic interpreter that reads the frozen artifact does. Remove the model entirely and the system still trades its existing rules safely; it simply stops acquiring new ones.
- **The verifier is deterministic and adversarial to its own inputs.** It evaluates every artifact out-of-sample, raises the acceptance bar in proportion to how many candidates were tried, and demands that any apparent edge survive perturbation of both time window and parameters.

This paper focuses on the verifier and reports its behavior on real data. Our contributions are:

1. **A gate design** that composes three standard-but-rarely-combined defenses against backtest overfitting — walk-forward out-of-sample evaluation, a trials-adjusted acceptance bar, and robustness sweeps — behind the discipline of a frozen, content-hashed artifact that keeps a nondeterministic author out of the execution path (§3–4).
2. **A real-data evaluation** (§5) in which the gate rejects every candidate it was given, including (a) a single strategy whose in-sample Sharpe of 3.07 became an out-of-sample -4.90, and (b) a momentum book whose lone positive out-of-sample number ($+0.13$ Sharpe edge) is shown by the robustness sweep to be a one-configuration fluke (2/5 windows, 0/12 settings).
3. **An honest accounting** (§5.5, §7) of what "the gate rejected everything" does and does not establish, including the ways our own trial-counting understates the true multiplicity of the search.

We want to be clear about the shape of the result. We did not find a profitable strategy. On efficient large-cap equities, over these windows, with price-only signals, we did not expect to, and the honest-validation literature predicts we should not [BacktestOverfitting; AFML]. The result we do report is that the gate behaves as designed: it takes strategies that look excellent and refuses them for concrete, reproducible reasons. A method whose headline demonstration is that it kills its author's best-looking ideas is, we argue, exactly the kind of method the field is short on.

2. Related Work

2.1 Backtest overfitting and multiple testing

That data-snooping inflates apparent performance is old news in finance. White's Reality Check [White] and the data-snooping study of Sullivan, Timmermann, and White [STW] gave bootstrap procedures for asking whether the best of many trading rules is better than chance. Harvey and Liu [HarveyLiu] argue that with hundreds of factors and strategies tested across the literature, conventional significance thresholds are far too lenient, and that the required hurdle should rise with the number of trials. Bailey, Borwein, López de Prado, and Zhu [BacktestOverfitting] show formally that with enough trials one can construct a strategy with an arbitrarily high in-sample Sharpe and zero true edge, and that in-sample optimization tends to *select against* out-of-sample performance. The Deflated Sharpe Ratio [DSR] operationalizes this by discounting an observed Sharpe for the number of trials, the length of the track record, and non-normality. López de Prado's *Advances in Financial Machine Learning* [AFML] consolidates these into walk-forward and combinatorially-purged cross-validation, and the Probability of Backtest Overfitting.

Our verifier is squarely in this tradition; it does not invent a new statistical test. Its trials-adjusted bar (§4.4) is a deliberately transparent, monotonic *heuristic* in the spirit of the Deflated Sharpe Ratio rather than a computation of it — the design choice is to make the penalty legible and hard to game, and we treat replacing it with a calibrated DSR as future work (§7).

2.2 Walk-forward and out-of-sample discipline

Out-of-sample evaluation is the standard defense: fit or select on one slice of history, judge on a later slice the strategy never influenced. Its weakness in practice is that a single split is a single draw. A strategy can clear one out-of-sample window by luck, especially after the researcher has, consciously or not, iterated against that window. This motivates evaluating stability across *many* windows and parameterizations [AFML], which our robustness sweep (§4.5) does directly, and which turns out to be decisive in §5.

2.3 Machine- and LLM-proposed strategies

Automated strategy search predates language models — genetic programming and AutoML for trading are decades old, and each is a well-known overfitting hazard precisely because they search a large space. Recent work uses LLMs to propose or code trading strategies from natural-language intuitions. The risk profile is the same but amplified: cheaper proposals mean more trials and thus more selection bias, and an LLM that emits runnable code puts a nondeterministic, potentially unsafe author directly in the execution path. ÀYÒ's response is to make the proposer emit *data*, not code, and to freeze and content-hash it (§4.1), so that the deployed behavior is fully reproducible even though the author is not.

2.4 Deterministic gates around nondeterministic models

The broader pattern — wrap a nondeterministic model in a deterministic layer that supplies a guarantee the model alone cannot — recurs across the author's systems: a governed memory engine that reconciles

and audits what an agent may store [Cortex], and a citation verifier that blocks fabricated quotations in an agentic workflow. The present paper is the empirical, single-domain instance of that pattern; a cross-domain synthesis is deferred to future work (§6).

3. The gate, formally (light)

We frame the system as a pair (P, V):

- P is a **proposer**: a nondeterministic map from a prompt to a candidate strategy artifact a. In ÀYÒ, P is an LLM. Nothing in the guarantees below depends on P being trustworthy, correct, or even non-adversarial.
- V is a **verifier**: a deterministic predicate $V(a, D) \in {\text{accept}, \text{reject}}$ evaluated against a fixed historical dataset D. Only artifacts for which V accepts may be executed, and execution reads only the frozen artifact.

The design aims for four properties. We state them as targets, not theorems; §4 describes the mechanisms and §7 the gaps.

1. **Reproducibility.** An artifact is content-addressed: identical trading logic yields an identical hash and therefore identical behavior. The executor recomputes the hash before running and refuses a mismatch. Validating the *artifact* rather than the *model* is what lets a nondeterministic author produce perfectly reproducible deployed behavior.
2. **Falsifiable acceptance.** V is a decidable predicate over measured quantities (out-of-sample Sharpe, drawdown, trade count, decay, robustness counts), not a model's judgment. Whether a given artifact passes is checkable by re-running the gate.
3. **Fail-safe degradation.** Removing P leaves a system that still trades its already-accepted artifacts under their frozen rules. Loss of the proposer degrades *improvement*, not *safety*.
4. **Non-gameability under search.** Because the acceptance bar rises with the number of candidates tested and requires stability across windows and parameters, a search cannot mine its way to acceptance simply by trying more variations — the very act of trying more raises the bar it must clear.

Property 4 is the one that does real work against the failure mode of §1, and it is what §5 stresses.

4. Method

4.1 Strategy artifacts: content-hashed, non-executable

A strategy is a frozen dataclass with an explicit universe, a list of entry rules (all must hold to enter), exit rules (any triggers an exit), a hard stop loss, a take profit, and a maximum holding period. Each rule is a declarative triple `(indicator, op, threshold)` — e.g. `(rsi_14, <, 35)` — evaluated by the executor against precomputed features. No text from the proposer is ever `exec`'d.

The artifact's content hash is a SHA-256 over its trading logic, excluding the pass/fail flag and the hash field itself. An artifact is deployable only if it was marked passed by the gate **and** its stored hash still equals a fresh recomputation. This is the mechanical basis for reproducibility (Property 1): the object that trades is exactly the object that was validated.

4.2 The base backtest gate

Every artifact is first simulated over the full history and must clear a fixed floor: at least 20 trades, Sharpe ≥ 1.0, maximum drawdown $\leq 15\%$, and win rate $\geq 45\%$. This floor screens obviously non-viable strategies but is *not* the honesty layer — a strategy can clear it purely by in-sample luck. The floor exists so that the out-of-sample test is spent only on candidates that are at least coherent.

4.3 Walk-forward out-of-sample evaluation

We split each symbol's history into an in-sample portion (the first 70%) and an out-of-sample portion (the remaining 30%, prefixed with a 50-bar warm-up so indicators are valid at the boundary but trades fire only inside the out-of-sample window). Candidate *selection* happens on in-sample data only; the winner is then judged on the out-of-sample slice it never influenced. In addition to clearing the floor out-of-sample, the winner must not have *decayed*: we require the out-of-sample Sharpe to retain at least half of the in-sample Sharpe. A strategy whose performance halves out-of-sample is treated as fragile even if it clears the raw threshold.

4.4 The trials-adjusted acceptance bar

The core anti-overfitting device is that the required out-of-sample Sharpe rises with the number of candidates tested:

```
required_oos_sharpe(N) = 1.0 + 0.15 * log2(N)
```

Testing 1 candidate requires an out-of-sample Sharpe of 1.0; 6 candidates requires 1.39; 64 candidates requires 1.90. The functional form is a transparent stand-in for the Deflated Sharpe Ratio's selection-bias correction [DSR]: monotonic in the number of trials, cheap to reason about, and impossible to satisfy by simply enumerating more variants. It is a heuristic, and we flag two honest weaknesses in §7 — the constants are not calibrated to a null distribution, and N counts only the candidates in a single scripted search, undercounting the true researcher degrees of freedom.

4.5 Robustness sweeps

For cross-sectional strategies we add two stress tests, because a single out-of-sample split is a single draw (§2.2):

- **Rolling windows.** Partition the post-warm-up period into 5 sequential windows and ask, in each, whether the strategy beats a buy-and-hold benchmark on a Sharpe basis. A real edge should appear in most windows, not one.

- **Parameter sensitivity.** Vary the strategy's free parameters (here, momentum look-back over {126, 189, 252} days and portfolio breadth `top_k` over {8, 12, 16, 20}, 12 combinations) and count how many settings beat the benchmark. A real edge should survive across the grid, not appear at one lucky point.

A strategy is treated as having a robust edge only if it beats the benchmark in most windows *and* most settings. Scattered signs indicate a fluke.

5. Experiments

5.1 Data and universe

All results use real daily bars retrieved from Alpaca on 2026-07-20. The single-strategy walk-forward (§5.2) uses a pre-declared universe of 32 large-cap U.S. equities over roughly four years. The cross-sectional experiments (§5.3–5.4) use a broader pre-declared universe of 83 large-cap equities; after alignment and warm-up this yields a panel of **959 trading days spanning 2022-09-20 → 2026-07-17**. Universes and features were fixed before results were inspected. The exact run logs backing every number below are included in the reproducibility appendix (§A); note the caveat there about the retrieval window.

5.2 The single-strategy overfit trap

Six rule-based "trend dip-buy" strategies were ranked on in-sample data. All looked strong in-sample; several posted in-sample Sharpes above 2 and in-sample returns in the hundreds of percent. The in-sample winner was `trend_dip_30` (buy oversold names in an uptrend), with an **in-sample Sharpe of 3.07** over 46 trades, a 57% win rate, and a 23.5% maximum drawdown.

Judged out-of-sample against a bar of 1.39 (6 trials), it did not merely underperform — it inverted:

```
IS Sharpe 3.07  ->  OOS Sharpe -4.90   (need >= 1.39)
OOS: 14 trades, win 36%, maxDD 30.4%, return -25.5%, decay_ok = False
VERDICT: REJECTED — not deployed
```

An in-sample Sharpe of 3.07 that becomes −4.90 out-of-sample is the canonical shape of an overfit backtest: the selection process found a configuration tuned to the noise of the in-sample period, and that tuning was actively harmful once the noise changed. This is the trap that empties accounts, and the gate's entire job is to see it and say no.

5.3 A single out-of-sample split can still fool you

Cross-sectional momentum — hold the strongest recent performers, rebalance periodically — is one of the most documented anomalies in the literature, so it is a fair test of whether the gate is merely pessimistic. We evaluated it in-sample and out-of-sample, with and without a market-regime filter, against a buy-and-hold benchmark of the same universe:

Configuration	Split	Strategy Sharpe	Benchmark Sharpe	Edge vs B&H
With regime filter	in-sample	0.40	0.91	−0.51
With regime filter	out-of-sample	0.41	1.08	−0.67
No regime filter	in-sample	0.58	0.91	−0.33
No regime filter	out-of-sample	1.21	1.08	**+0.13**

Three of the four cells lose to simply buying the universe. The fourth — momentum with no regime filter, out-of-sample — beats the benchmark by a Sharpe edge of +0.13. Taken alone, that single positive out-of-sample number is exactly the kind of result a hopeful researcher publishes: an anomaly that "works out-of-sample." A naive out-of-sample check would pass it.

5.4 The robustness sweep exposes the fluke

The +0.13 does not survive contact with §4.5. Across 5 sequential rolling windows, momentum beat buy-and-hold in only **2 of 5**, and the wins were marginal while the losses were large:

```
window 1 (2023-10-03 -> 2024-04-23): edge -0.40  lost
window 2 (2024-04-24 -> 2024-11-11): edge -1.76  lost
window 3 (2024-11-12 -> 2025-06-05): edge +0.05  BEAT
window 4 (2025-06-06 -> 2025-12-24): edge -1.17  lost
window 5 (2025-12-26 -> 2026-07-17): edge +0.13  BEAT
-> beat buy&hold in 2/5 windows
```

Across the 12-point parameter grid, the edge was positive in **0 of 12** settings — every look-back / breadth combination lost to buy-and-hold, by margins from −0.22 to −0.65:

```
lookback in {126,189,252} x top_k in {8,12,16,20}
edges ranged -0.22 to -0.65 ; positive in 0/12 settings
```

The reading is unambiguous. The +0.13 of §5.3 was a single lucky configuration on a single split, not an edge. Robustness across windows and parameters is what distinguishes the two, and it is why an out-of-sample split — necessary as it is — is not sufficient on its own.

5.5 Summary, and what it does and does not establish

Every candidate the gate was given was rejected: the single-strategy winner on decay and sign inversion, and cross-sectional momentum on robustness. On efficient large-cap equities, over these windows, with price-only signals, this is the outcome honest validation predicts [BacktestOverfitting; AFML], and producing it is the point.

What this **establishes**: the gate's mechanisms behave as designed on real data. It quarantines the proposer behind a frozen, hash-checked artifact; it converts a spectacular in-sample Sharpe (3.07) into a concrete rejection (−4.90 out-of-sample); and it correctly reclassifies a plausible out-of-sample "edge"

(+0.13) as a fluke once stability is required (2/5, 0/12). The negative results are reproducible from the included logs.

What this does **not** establish: that these strategies have no edge in any universe, timeframe, or cost regime; that the specific constants in the bar are optimal; or, most importantly, that the gate has been stress-tested against a strategy that *deserves* to pass. A rejecter is only fully validated once it also *accepts* something true. We have shown the gate says no to mirages; we have not yet shown it says yes to a real edge, because we have not yet found one to offer it.

6. Discussion: the pattern generalizes

The device at the center of this paper — quarantine a nondeterministic proposer behind a deterministic, falsifiable verifier, and validate the frozen artifact rather than the model — is not specific to trading. The same shape appears wherever an LLM's fluency outruns its trustworthiness: a governed memory engine that admits a fact to long-term storage only after deterministic reconciliation and records why [Cortex]; a citation verifier that lets an agent's answer through only if each quoted span is found verbatim in the cited source. In each case the LLM proposes and a deterministic gate disposes, and in each case the engineering value is concentrated in the gate, not the proposer. Trading is the sharpest test of the idea because the cost of a false accept is measured directly in money and the environment is adversarial by default. A cross-domain formalization of the proposer/verifier pattern and its invariants is the subject of planned follow-on work; this paper is its trading instance.

7. Limitations and future work

- **No accepted strategy yet.** The gate has been shown to reject, not to accept correctly. The most important next experiment is to construct or discover a strategy with a genuine edge (likely from signals not contained in price alone) and confirm the gate passes it. Until then, "the gate works" means "the gate reliably refuses mirages."
- **The bar is a heuristic, not a calibrated test.** `1.0 + 0.15·log2(N)` is monotonic and legible but its constants are not derived from a null distribution. Replacing it with a properly Deflated Sharpe Ratio [DSR] and reporting the Probability of Backtest Overfitting [AFML] would put the acceptance decision on firmer statistical ground.
- **Trial-counting undercounts researcher degrees of freedom.** N counts candidates in one scripted search. The true multiplicity across the whole research program — every universe, feature, and threshold the author has ever considered — is far larger, so the effective bar is, if anything, too lenient. This is the honest hardest problem in the area and we do not claim to have solved it.
- **Single data snapshot and daily bars.** Results are from one retrieval window of daily U.S. equity bars. Combinatorially-purged cross-validation, multiple snapshots, other asset classes and frequencies, and an explicit transaction-cost, borrow, and capacity model would strengthen every claim.
- **Reproducibility window.** The data fetch is anchored to the retrieval date, so re-running on a later date shifts the window. We treat the committed run logs and a pinned data snapshot as the frozen

reference (§A); making the fetch fully date-pinned is a small, planned change.

8. Conclusion

The scarce resource in systematic trading is not strategy ideas; it is not being fooled by them, and a fast proposer makes that scarcer, not more abundant. ÀYÒ concentrates its engineering accordingly: it treats an LLM as a cheap, untrusted proposer of frozen data artifacts, and spends its rigor on a deterministic gate that validates the artifact out-of-sample, raises its bar with the number of trials, and demands robustness across windows and parameters. On real markets the gate did what it was built to do — it turned an in-sample Sharpe of 3.07 into a -4.90 rejection, and unmasked a $+0.13$ out-of-sample "edge" as a 0-of-12 fluke — and it deployed nothing, which is the correct answer on efficient markets. The result is not an alpha but an artifact the field is short on: a validation layer that reliably refuses to fool itself, and a pattern for holding nondeterministic proposers to deterministic guarantees.

References

[BacktestOverfitting] D. Bailey, J. Borwein, M. López de Prado, Q. Zhu. "Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-of-Sample Performance." *Notices of the American Mathematical Society*, 61(5), 2014.

[DSR] D. Bailey, M. López de Prado. "The Deflated Sharpe Ratio: Correcting for Selection Bias, Backtest Overfitting, and Non-Normality." *Journal of Portfolio Management*, 40(5), 2014.

[HarveyLiu] C. Harvey, Y. Liu. "Backtesting." *Journal of Portfolio Management*, 42(1), 2015. See also C. Harvey, Y. Liu, H. Zhu, "…and the Cross-Section of Expected Returns," *Review of Financial Studies*, 29(1), 2016.

[White] H. White. "A Reality Check for Data Snooping." *Econometrica*, 68(5), 2000.

[STW] R. Sullivan, A. Timmermann, H. White. "Data-Snooping, Technical Trading Rule Performance, and the Bootstrap." *Journal of Finance*, 54(5), 1999.

[AFML] M. López de Prado. *Advances in Financial Machine Learning*. Wiley, 2018.

[Cortex] S. Olagoke. "Cortex: A Governed Long-Term Memory Engine for LLM Agents." Preprint (draft), 2026.

Appendix A. Reproducibility

The three experiments are reproducible from the ÀYÒ repository:

```
python -m tests.find_strategy     # §5.2 single-strategy walk-forward
python -m tests.research_xsec      # §5.3 cross-sectional momentum IS/OOS
python -m tests.robustness_xsec    # §5.4 rolling-window + parameter sweep
```

Verbatim stdout from the runs backing this paper is committed under `data/runs/` (`find_strategy_run.log`, `research_xsec_run.log`, `robustness_run.log`). Bars are real Alpaca daily data retrieved 2026-07-20; the cross-sectional panel spans 2022-09-20 → 2026-07-17 (959 trading days, 83 symbols). Because `get_bars` / `get_price_panel` anchor the fetch to the retrieval date, exact reproduction is against the committed logs and the pinned snapshot rather than a fresh fetch on a later date.