

Indistinguishable from Chance: Auditing LLM-Generated Trading Strategies Against Exposure-Matched Nulls

Rajaa Filali

`r.filali@emsi.ma`

SMARTiLab, EMSI

Said El Qotbi

SMARTiLab, EMSI

Research Article

Keywords: large language models, algorithmic trading, backtest overfitting, null models, permutation tests, market timing

Posted Date: October 23rd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11272412/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Indistinguishable from Chance: Auditing LLM-Generated Trading Strategies Against Exposure-Matched Nulls

Rajaa Filali^{1*} and Said El Qotbi¹

^{1*}SMARTiLab, EMSI, Rabat, Morocco.

*Corresponding author(s). E-mail(s): r.filali@emsi.ma;

Abstract

Large language models will, on request, write a complete trading strategy and describe an impressive backtest. We measure what these strategies deliver. Five models from three providers generate 198 strategies under four prompt styles; 403 strategy-asset audits execute in a sandbox on a daily panel (S&P 500, gold, EUR/USD). The mean delivered Sharpe ratio is -0.09 (95% CI $[-0.12, -0.05]$), and 92% of audits fail to beat buy-and-hold. The strategies are statistically indistinguishable from symmetric random ones ($p = 0.16$). The central test compares each strategy with its own exposure-matched null, obtained by shuffling its blocks of constant position; oracles with a known edge show that the test detects modest timing skill. Gross of costs, the strategies earn $+0.07$ where random timing with the same exposure earns $+0.11$, and only 7 of 403 show significant timing, fewer than chance ($p < 0.001$). Every apparent improvement—a robustness prompt, a larger model, the gold market—comes from exposure, not timing. Asked to backtest their own strategy, no model reports a computed figure; three fabricate a results block. For investors who use these tools, an AI-produced backtest is a hypothesis, not evidence. We release the pipeline and all controls.

Keywords: large language models, algorithmic trading, backtest overfitting, null models, permutation tests, market timing

JEL Classification: G11 , G14 , G17 , C12 , C45

1 Introduction

Ask a large language model (LLM) for a profitable trading strategy and it will oblige within seconds: runnable code, a plausible rationale, and often a performance figure. The cost of producing a convincing backtest has collapsed. The cost of a wrong one has not, and the ways a backtest can mislead are long established (Bailey, Borwein, López de Prado, & Zhu, 2014; Harvey, Liu, & Zhu, 2016; López de Prado, 2018; White, 2000). Code written by a machine is clean and confident, which makes an empty strategy harder to recognise. The question is economic: retail and professional investors increasingly obtain trading logic from general-purpose assistants, and the evidence those assistants supply is the evidence on which they allocate savings. If that evidence is uninformative, AI assistance lowers the cost of producing a strategy without raising its expected return, and the cost is borne by the investors who act on it.

A growing literature studies language models as strategy generators. Capability benchmarks ask whether models can carry out a backtest correctly (Khoroshilov, Chernysh, Ekhtibarov, Kamkia, & Zmitrovich, 2026; Wang et al., 2026); reproducibility reviews ask whether the published LLM-trading literature documents its assumptions (Yao & Zheng, 2026); performance studies ask whether a given LLM approach beats the market (Li, Kim, Cucuringu, & Ma, 2025). We ask a question downstream of all three: taking the strategies that models produce on demand, do they contain any skill once executed on documented data, and does what they present match what they deliver?

Answering requires care about what “no skill” means. An audit that rejects every strategy cannot by itself distinguish weak strategies from a harsh auditor. We therefore measure skill against controls that state what chance looks like:

- a *positive control*, graded oracles whose true edge is known, which calibrates what the audit can detect;
- a *negative control*, symmetric random strategies;
- a *human baseline*, textbook trading rules run through the same pipeline;
- an *exposure-matched null*, the central test: each LLM strategy is compared with itself after its blocks of constant position are shuffled in time, which keeps its exposure and its blocks of positions and removes only its timing.

We make five contributions. (i) A reproducible generation-and-audit pipeline with its controls. (ii) The finding that LLM-generated strategies contain no timing skill: they are indistinguishable from random strategies and do no better than their own exposure-matched nulls, a test we show can detect modest skill. (iii) The finding that every apparent improvement—from prompting for robustness, from a larger model, and on a trending asset—is explained by exposure. (iv) The *reporting illusion*: asked to backtest their strategy, models return the form of a performance report without its substance. (v) A record of four artefacts of our own harness and data, caught by self-audit, including a data misalignment that had produced our best-looking strategy.

2 Related literature

Backtest overfitting and data snooping. The classical literature on spurious performance provides our diagnostics and our standard of evidence: the Deflated Sharpe Ratio (Bailey & López de Prado, 2014) and the Probability of Backtest Overfitting (Bailey, Borwein, López de Prado, & Zhu, 2017); reality-check and data-snooping tests (Sullivan, Timmermann, & White, 1999; White, 2000); multiple-testing corrections in asset pricing (Harvey & Liu, 2020; Harvey et al., 2016); the post-publication decay of anomalies (McLean & Pontiff, 2016); and the backtesting protocol of Arnott, Harvey, and Markowitz (2019). These works establish *how* a backtest can mislead; we ask how often a modern generator produces such backtests.

Random and permutation benchmarks. Comparing a trading rule with randomised versions of itself is an old idea in technical-analysis research (Brock, Lakonishok, & LeBaron, 1992; Sullivan et al., 1999). Bootstrap and permutation nulls preserve chosen properties of a strategy while destroying the information it claims to use. Our exposure-matched null applies this idea to generated strategies: it preserves exposure and the blocks of positions and removes timing.

LLMs as strategy generators. BacktestBench (Wang et al., 2026) and QuantCode-Bench (Khoroshilov et al., 2026) measure whether an LLM can carry out a backtesting task competently. We take competence as given where the code runs, and ask whether the resulting strategy has any skill. Chen, Chen, Chen, and Sra (2025) argue that standard benchmarks miss the risks of LLM finance agents; we share that spirit for generated strategies. Yao and Zheng (2026) audit thirty published LLM-trading studies for their execution assumptions; their unit of analysis is the literature, ours the generated strategy. A growing body documents hallucination of financial figures (Lopez-Lira & Tang, 2023; Zhang et al., 2025); we measure what models do when asked to report performance they cannot have computed.

3 Method

3.1 Generation

Five models from three providers—GPT-5.4-mini (OpenAI), Gemini 2.5 Flash and Gemini 3.5 Flash (Google), Llama 3.3 70B and Llama 3.1 8B (Meta, open weights)—receive four prompt conditions, ten times each, at temperature 0.7 (Table 1). P0 asks plainly for a profitable strategy; P1 adds an instruction to avoid overfitting and to validate out of sample; P2 asks for a strategy on a basket; P3 asks the model to write *and* backtest a strategy, reporting its Sharpe ratio and win rate, without supplied data. All raw outputs are archived with a timestamped manifest. Of 200 requested generations, 198 succeeded; two were lost to free-tier rate limits.

3.2 Execution

A parser extracts each strategy’s signal function; a sandbox (static analysis plus an isolated subprocess with a timeout) executes it on each panel asset. Positions are applied to the next bar’s return, and every unit change of position costs 0.05% of notional: $r_t = w_{t-1} (P_t/P_{t-1} - 1) - 0.0005 |w_t - w_{t-1}|$. The delivered Sharpe ratio is

Table 1 Experimental design.

Factor	Levels
Models (5)	GPT-5.4-mini; Gemini 2.5 / 3.5 Flash; Llama 3.3 70B / 3.1 8B
Prompts (4)	P0 naive; P1 robustness; P2 basket; P3 write-and-backtest
Repetitions	10 per cell, temperature 0.7, raw outputs archived
Panel	S&P 500, gold (COMEX futures), EUR/USD (broker feed); 3,000 daily bars each, 2014–2026
Execution	positions applied next bar; cost 0.05% per unit change of position

$\sqrt{252}$ times the mean-to-volatility ratio of daily returns, without a risk-free rate. The *unit of analysis* is the strategy–asset audit: each of the 150 P0–P2 generations is run on the three assets, giving 450 attempts, of which 403 execute and take positions. The 48 P3 generations produce no extractable signal function and are analysed separately (Section 4.5).

Prices are the last 3,000 daily bars per asset, ending on 8 July 2026. The S&P 500 index ($\sim$ GSPC) and COMEX gold futures as a continuous front-month series (GC=F) come from Yahoo Finance; gold is thus a futures price, not spot. EUR/USD comes from the daily bars of a retail broker feed (IC Markets, MetaTrader 5). We first used Yahoo’s EURUSD=X series, but its daily bars are misaligned: its returns match the broker’s returns of the previous day (correlation 0.95), and the high and low of a bar extend beyond its close, so any rule that uses the daily range sees part of the next day (Section 5). We checked all three series against official ones: the S&P 500 closes coincide with FRED over the ten years it publishes, and gold and EUR/USD deviate from the LBMA afternoon fix and the ECB reference rate by a median of 0.24% and 0.19%. Daily returns correlate less (0.81 and 0.47) because the fixings are taken at other times of day, but 20-day returns correlate at 0.97–0.99. The sandbox rejected code attempting filesystem or system access. Executing model-written code is inherently risky: manual review of rejected code is feasible at this scale, and a disposable virtual machine is advisable for larger runs.

3.3 Audit and controls

Each executed strategy is scored by a composite credibility auditor, described in a companion manuscript by the same authors that is currently under review. It combines causality, drift, cost-sensitivity, multiplicity, seed-stability and benchmark diagnostics—including the Deflated Sharpe Ratio (Bailey & López de Prado, 2014) and a fabricated-data detector based on stylised facts and two-sample tests (Cont, 2001; Gretton, Borgwardt, Rasch, Schölkopf, & Smola, 2012; Lopez-Paz & Oquab, 2017)—through a calibrated logistic fusion model trained on labelled synthetic cases with twenty configurations each. Its role in this study is limited by the calibration result of Section 4.2. Our conclusions rest on three auditor-independent quantities: the delivered Sharpe ratio, the comparison with buy-and-hold on the same asset, and the comparison with the controls below. All controls run through exactly the same sandbox, panel, execution and cost.

E1, positive control. A *daily oracle* takes the sign of the next bar’s return with probability $p \in \{0.50, 0.51, 0.52, 0.53, 0.55, 0.60, 0.70, 0.80, 0.90\}$ and the opposite position otherwise (10 seeds for $p \leq 0.60$, 5 above). A *slow oracle* holds for 20 days the direction of the next 20-day move with probability $p \in \{0.55, 0.60, 0.70, 0.80, 0.90\}$ (5 seeds), so that its edge is not consumed by costs. Oracles are not tradable strategies; they are gauges with a known, graded edge.

E2, negative control. One hundred symmetric random strategies per asset (300 audits), long or short with equal probability for holding periods drawn uniformly between 1 and 20 days.

E3, human baseline. Buy-and-hold and six textbook rules: a 50/200-day moving-average crossover (long-only and long-short), twelve-month time-series momentum, a 55-day breakout, RSI(14) mean reversion with thresholds 30/70, and Bollinger(20) mean reversion.

E4, exposure-matched null. For each LLM audit, the position series is cut at every change of position, and the resulting blocks are concatenated in a random order, 200 times. Each shuffle has exactly the time spent long, short and flat of the original strategy and the same blocks; only the timing differs. Because two shuffled blocks of equal sign can become adjacent and merge, a shuffle may trade less than the original and pay less in costs. We therefore run the test on gross returns, which merging leaves unchanged, and report costs separately. The one-sided p -value is $(1 + \#\{\text{shuffles with Sharpe} \geq \text{strategy Sharpe}\}) / (1 + 200)$. Under the hypothesis of no timing skill, p is uniform and about 5% of audits have $p < 0.05$. To measure the power of this test, we run through it oracles with a known timing edge that hold their position for 1, 5 or 20 days (10 seeds on each asset).

Statistics. The three audits of a generation share the same code and are not independent. We therefore treat the generation as the unit of resampling: means are reported with 95% cluster-bootstrap intervals (5,000 resamples of the 135 executed generations), and group comparisons use Mann–Whitney tests on generation means. Proportions carry Wilson intervals, paired strategy–null differences are tested with the Wilcoxon signed-rank test, and the number of significant audits with a binomial test against 5%.

3.4 Use of AI tools in preparing this study

The language models evaluated in this study are its object. Separately, the authors used Claude (Anthropic) for code development and manuscript drafting; no Anthropic model was among the models evaluated. The authors reviewed and edited all content, every reported number can be recomputed from the released scripts, and the authors take full responsibility for the content of the article.

4 Results

4.1 Delivered performance

Execution is reliable: 90% of P0–P2 audits run and take positions, from 73% for Llama 3.1 8B to 100% for Llama 3.3 70B. Performance is not. The mean delivered

Sharpe ratio is -0.087 (95% CI $[-0.123, -0.051]$), negative for four of the five models (Table 2). Only 32 of 403 audits beat buy-and-hold on their asset (7.9%, Wilson CI $[5.7\%, 11.0\%]$), against buy-and-hold Sharpe ratios of 0.74 for the S&P 500, 0.67 for gold and -0.06 for EUR/USD.

Table 2 Execution and delivered performance by model (P0–P2 audits). *Sharpe* is net of costs. The timing test is gross of costs: *Gross* is the strategy’s gross Sharpe ratio, *Null* the mean gross Sharpe ratio of its exposure-matched null, *Timing* their difference, and $p < 0.05$ counts audits with significant timing.

Model	Executed	Sharpe	Beat B&H	Gross	Null	Timing	$p < 0.05$
GPT-5.4-mini	85/90	-0.124	5.9%	$+0.054$	$+0.097$	-0.043	1
Gemini 2.5 Flash	75/90	$+0.068$	16.0%	$+0.144$	$+0.162$	-0.018	2
Gemini 3.5 Flash	87/90	-0.041	4.6%	$+0.083$	$+0.188$	-0.105	0
Llama 3.3 70B	90/90	-0.096	10.0%	$+0.068$	$+0.086$	-0.017	1
Llama 3.1 8B	66/90	-0.263	3.0%	$+0.007$	$+0.018$	-0.011	3
All	403/450	-0.087	7.9%	$+0.072$	$+0.113$	-0.041	7

4.2 What the audit can detect

The positive controls answer the question an audit without controls cannot: would a strategy with real skill be recognised? We ask it of both instruments.

The credibility score. Figure 1 plots the score against the delivered Sharpe ratio of the oracles. The score rises with the edge, but slowly. A daily oracle with a Sharpe ratio of 1.21 scores 0.14; a score of 0.30 requires a Sharpe ratio of about 5, and a score of 0.50 one of about 9. The slow oracle reaches a Sharpe ratio of 1.81 and scores 0.16; buy-and-hold scores 0.09. The cause is a mismatch between training and use: the fusion model was calibrated on cases with twenty configurations per strategy, for which multiplicity statistics such as the Probability of Backtest Overfitting are defined, whereas a generated strategy is a single configuration. On such inputs the score is a compressed function of the Sharpe ratio. The highest score of any LLM strategy, 0.13, corresponds to an oracle Sharpe ratio of about 1.0; the score adds no information to the Sharpe ratio, and any survival threshold of the kind we first used (0.30) would reject strategies with a Sharpe ratio below about 5. We do not use the score as a survival criterion; the tests below do not depend on it.

The exposure-matched test. The test that carries our conclusions does recognise skill (Table 3). Oracles whose timing contributes a gross Sharpe ratio of about 0.5–0.6 are detected in 63–80% of runs at the 5% level, and those contributing about 1.0 in every run, whether they hold positions for one, five or twenty days. A contribution of 0.2–0.3 is detected in 10–23% of runs: single audits have limited power against small edges, but across 403 audits even a small average skill would raise the share of significant cases well above 5%.

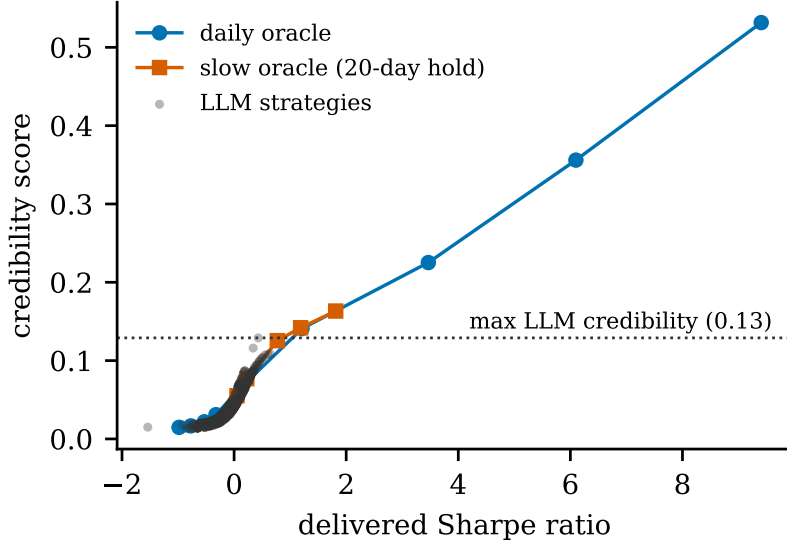


Fig. 1 Positive control. Credibility score against delivered Sharpe ratio for oracles of graded accuracy (daily, and 20-day holding), with the LLM strategies in grey. The highest LLM score (0.13, dotted line) corresponds to an oracle Sharpe ratio of about 1.0; a score of 0.30 requires about 5

Table 3 Power of the exposure-matched test. Oracles hold each block of 1, 5 or 20 days in the direction of that block’s return with probability p (10 seeds on each of the three assets, 200 shuffles each). *Timing* is the oracle’s gross Sharpe ratio minus that of its null; *Detected* is the share of runs with $p < 0.05$.

Holding (days)	p	Gross Sharpe	Timing	Detected
1	0.51	0.18	0.18	10%
1	0.52	0.39	0.39	37%
1	0.53	0.61	0.61	80%
1	0.55	1.04	1.04	100%
5	0.55	0.63	0.62	70%
5	0.60	1.12	1.10	100%
20	0.55	0.29	0.27	23%
20	0.60	0.52	0.48	63%
20	0.70	0.99	0.93	100%

4.3 Against chance and against the textbook

The LLM strategies and the symmetric random strategies deliver almost the same distribution of Sharpe ratios (Figure 2): -0.087 against -0.089 , Mann–Whitney $p = 0.16$ on generation means ($p = 0.43$ on audits), Kolmogorov–Smirnov $p = 0.20$. They beat buy-and-hold less often: 7.9% against 13.3% (Fisher $p = 0.02$). The textbook rules

fare no better. Their Sharpe ratios range from -0.17 (Bollinger reversion) to $+0.34$ (long-only moving-average crossover), and none beats buy-and-hold on the S&P 500 or gold. What the models generate is the textbook folklore of technical trading, and on this panel this folklore is indistinguishable from chance.

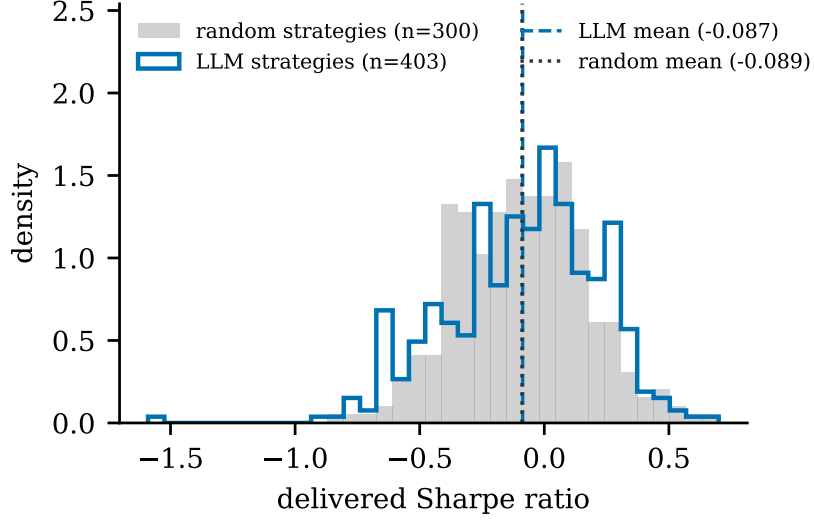


Fig. 2 Negative control. Delivered Sharpe ratios of the LLM strategies and of symmetric random strategies run through the same pipeline; dashed and dotted lines mark the two means

One asset is an exception at first sight. On gold, the LLM strategies earn $+0.02$ against -0.08 for the random strategies (Mann–Whitney $p = 0.004$). The next test shows why.

4.4 Exposure, not timing

Symmetric random strategies are long half of the time; the LLM strategies are not. On the S&P 500 they are long 46% of the time, short 18% and flat 36%. On a market that trends up, a long bias earns the drift without any timing skill. The exposure-matched null removes this confound by comparing each strategy with shuffles of its own positions.

The result is clear (Figure 3). Gross of costs, the LLM strategies earn a mean Sharpe ratio of $+0.072$; their exposure-matched nulls earn $+0.113$ (Wilcoxon signed-rank test on generation means, $p = 10^{-4}$). Only 7 of 403 audits (1.7%) show significant timing at the 5% level, fewer than the 5% expected by chance alone (binomial $p < 0.001$). No asset shows positive timing. On gold and EUR/USD the timing component is small and not significant ($+0.029$, $p = 0.06$, and $+0.017$, $p = 0.21$); on the S&P 500 it is significantly negative (-0.168 , $p < 10^{-10}$), and 77% of strategies do worse than shuffles of their own positions. The advantage on gold is thus exposure: of its gross

Sharpe ratio of +0.140, random timing with the same exposure earns +0.112. After costs, the gap between strategies and nulls widens (-0.085 against $+0.001$), partly because shuffles trade less; this is why the test uses gross returns.

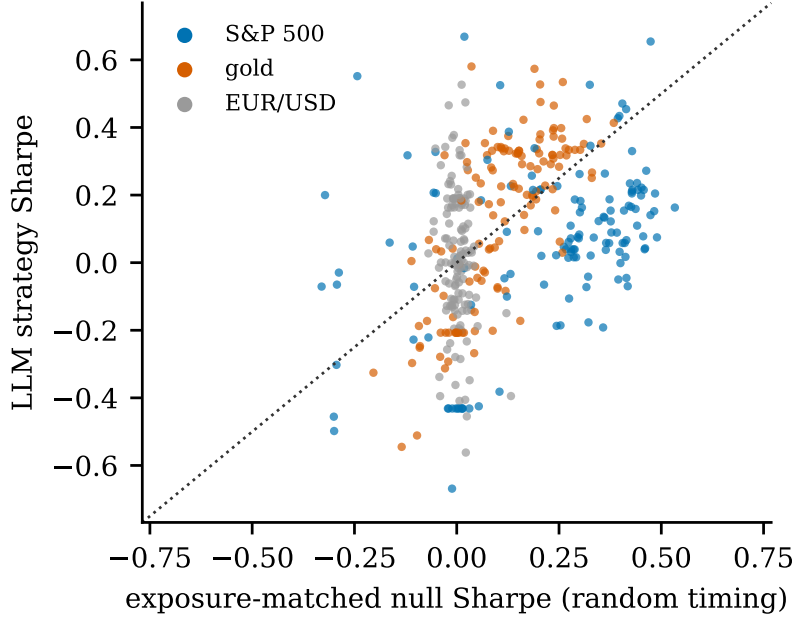


Fig. 3 Exposure-matched null, gross of costs. Each point is an LLM audit: its Sharpe ratio against the mean Sharpe ratio of 200 shuffles of its own blocks of constant position (same exposure, random timing). Points below the diagonal have worse-than-random timing

The same decomposition explains the two apparent improvements. The robustness prompt raises the delivered Sharpe ratio from -0.136 under P0 to -0.032 under P1, a difference that is significant across audits (Mann–Whitney $p = 0.009$) but not on generation means ($p = 0.09$); the larger Llama model outperforms the smaller one (-0.096 against -0.263 , $p = 0.010$ on generation means). In both cases the gain is exposure. Strategies written under P1 are long 42% of the time against 34% under P0, their exposure-matched nulls earn more ($+0.142$ against $+0.094$, $p = 0.016$), and their timing is no better ($p = 0.23$). The two Llama models differ through their nulls ($+0.086$ against $+0.018$, $p = 0.027$), not through timing ($p = 0.99$). Prompting and scale change what a strategy holds; they do not make it choose its moments better.

4.5 The reporting illusion

The P3 condition asks each model to write a strategy *and* backtest it, reporting its Sharpe ratio and win rate, without supplying data. None of the 48 outputs reports a genuinely computed figure, and none can, since nothing is executed at generation

time (Table 4, Figure 4). Twenty-seven outputs (56%) return an unexecuted script whose print statements would display metrics if someone ran it: the form of an answer without its substance. Sixteen (33%) add pedagogical thresholds (“a Sharpe above 1 is considered good”) without making a claim. Five commit to numbers: two forecast their own performance optimistically, and three present an entirely fabricated results block, formatted as the output of a run that never took place. Fabrication of *data* is rare: 46 of 48 outputs load genuine market prices through a data API. The failure these outputs share is a simulation of due diligence: deliverables that look like a completed backtest report and are not one.

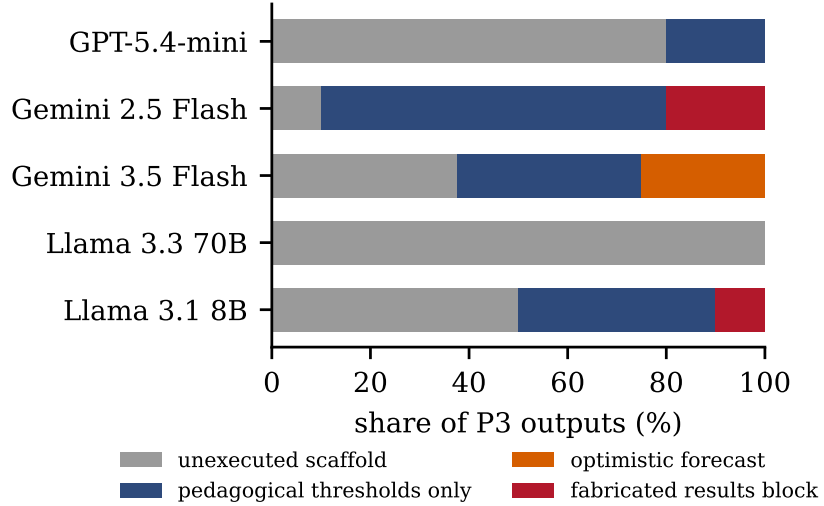


Fig. 4 Reporting behaviour of the 48 P3 outputs. No model reports a computed figure; most return an unexecuted scaffold or pedagogy

Table 4 What P3 outputs do instead of reporting a computed result (48 outputs).

Model	Scaffold	Pedagogy	Forecast	Fabricated	Real data
GPT-5.4-mini	8/10	2/10	0	0	10/10
Gemini 2.5 Flash	1/10	7/10	0	2/10	9/10
Gemini 3.5 Flash	3/8	3/8	2/8	0	8/8
Llama 3.3 70B	10/10	0	0	0	9/10
Llama 3.1 8B	5/10	4/10	0	1/10	10/10
All	56%	33%	2/48	3/48	96%

5 Discussion

An auditor must audit itself. Four artefacts of our own harness and data would have produced confident, wrong findings had they not been caught. First, our initial evaluation rejected about two thirds of one model’s strategies for a missing data column—a column our prompt had promised and our data files did not supply. Second, our first extraction of “claimed Sharpe ratios” from P3 outputs read the format specifiers of Python f-strings (`{sharpe:.2f}` yielding “2”) and manufactured a striking headline, models “claiming Sharpe 2–4”, which dissolved under manual inspection. Third, an earlier version of this study rested its headline on the credibility score; only the positive control revealed that, on single-configuration strategies, the score compresses a Sharpe ratio of 1.2 into 0.14. Fourth, a draft of this paper reported a best LLM strategy with a Sharpe ratio of 2.14 on EUR/USD, a plain stochastic-oscillator rule. Its code was causal; the data were not. In the Yahoo daily FX series the high and low of a bar extend beyond its close, so a rule built on the daily range sees part of the next day. On broker data the same rule earns -0.29 , and we replaced the series. The four episodes are the disease this paper studies, one level up: a plausible measurement is not a correct one until its assumptions are checked. A positive control is the cheapest such check, and any audit that reports “nothing survived” should show that something could.

The economic stakes. An investor who replaced holding the S&P 500 with the average LLM strategy would have exchanged a Sharpe ratio of 0.74 for one of -0.05 over 2014–2026; on gold, 0.67 for $+0.02$. The cost of AI-generated trading logic is not a failure to beat the market by a small margin but the loss of most of the return that passive holding provides.

What the null models change. Comparing a strategy with buy-and-hold answers whether it is worth holding; it does not answer whether it contains skill. A long-biased strategy on a rising market can beat cash and lose to buy-and-hold while containing no information at all. The exposure-matched null isolates the one thing a timing strategy claims to contribute, and finds it absent. Exposure, not timing, also explains every improvement we observed, which suggests a simple diagnostic for practitioners and for platforms that distribute AI-generated strategies: before crediting a strategy, compare it with shuffles of its own positions.

Why the illusion matters. The practical danger of AI-generated strategies is not that they crash—most run—and not, on this evidence, that they fabricate data. It is that they return artefacts with every appearance of completed due diligence. A user cannot tell an unexecuted scaffold from a validated strategy by looking. An AI-produced backtest artefact should be treated as unexecuted until it is re-run on documented data and compared with buy-and-hold and with its own exposure-matched null.

Limitations. The study covers five models from three providers at the API tiers accessible to us: smaller, faster variants rather than the largest flagships, which are also the tiers that retail users reach for free or at low cost. Within the Llama family, the larger model writes runnable code more reliably and holds a more favourable exposure, but shows no better timing. The corpus is 198 strategies, so rates are indicative rather than population estimates. The panel is three daily assets over 2014–2026, with a

uniform cost of 0.05% per unit change of position; intraday strategies and other asset classes are not covered. Part of the generated code is not deterministic, for example models fitted without a fixed random seed: re-executing the corpus changes the Sharpe ratio of 57 of 403 audits by more than 0.01 and of 7 by more than 0.05, without changing any aggregate reported here. The models were trained on text that covers most of the sample period, so their strategies could in principle be fitted to known history; that they still show no skill strengthens the conclusion. The P3 classification uses regular-expression rules with manual verification. We have not tested whether the models can exploit an edge planted in synthetic data and described to them; that experiment would separate an inability to find skill from the absence of skill to find.

6 Conclusion

Large language models generate trading strategies fluently and confidently. On documented data, those strategies deliver a negative mean Sharpe ratio, beat buy-and-hold in fewer than one case in ten, and are statistically indistinguishable from random strategies. Compared with their own exposure-matched nulls, they show no timing skill, and fewer significant cases than chance would produce. What prompting for robustness and larger models improve is exposure, not timing. Asked to backtest and report, the models return the form of a performance report without its substance. An AI-produced backtest is a hypothesis, not evidence, until it is re-run and compared with a null that matches its exposure—and an audit that finds nothing must first show that it could find something.

Statements and Declarations

Competing interests. The authors have no relevant financial or non-financial interests to disclose.

Funding. The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Author contributions. Rajaa Filali: conceptualization, methodology, software, formal analysis, investigation, data curation, visualization, writing – review and editing. Said El Qotbi: conceptualization, methodology, formal analysis. All authors read and approved the final manuscript.

Data availability. The generation-and-audit pipeline, the archived model outputs, the control strategies and the scripts that reproduce every table and figure are available at Zenodo, <https://doi.org/10.5281/zenodo.23107268>. Price data are not redistributed: S&P 500 and gold prices were obtained from Yahoo Finance (tickers `^GSPC` and `GC=F`) and EUR/USD from the IC Markets MetaTrader 5 feed, available with a free demonstration account. The repository provides download scripts together with the checksums, row counts and date ranges of the files used. The official series used for validation are publicly available from FRED, the LBMA and the ECB.

Ethics approval. Not applicable: the study involves no human participants or animals.

ORCID. Rajaa Filali: <https://orcid.org/0009-0009-5191-5343>

References

- Arnott, R.D., Harvey, C.R., Markowitz, H. (2019). A backtesting protocol in the era of machine learning. *Journal of Financial Data Science*, 1(1), 64–74, <https://doi.org/10.3905/jfds.2019.1.064>
- Bailey, D.H., Borwein, J.M., López de Prado, M., Zhu, Q.J. (2014). Pseudo-mathematics and financial charlatanism: The effects of backtest overfitting on out-of-sample performance. *Notices of the American Mathematical Society*, 61(5), 458–471, <https://doi.org/10.1090/noti1105>
- Bailey, D.H., Borwein, J.M., López de Prado, M., Zhu, Q.J. (2017). The probability of backtest overfitting. *Journal of Computational Finance*, 20(4), 39–69, <https://doi.org/10.21314/jcf.2016.322>
- Bailey, D.H., & López de Prado, M. (2014). The deflated Sharpe ratio: Correcting for selection bias, backtest overfitting, and non-normality. *Journal of Portfolio Management*, 40(5), 94–107, <https://doi.org/10.3905/jpm.2014.40.5.094>

- Brock, W., Lakonishok, J., LeBaron, B. (1992). Simple technical trading rules and the stochastic properties of stock returns. *Journal of Finance*, 47(5), 1731–1764, <https://doi.org/10.1111/j.1540-6261.1992.tb04681.x>
- Chen, Z., Chen, J., Chen, J., Sra, M. (2025). *Standard benchmarks fail – auditing LLM agents in finance must prioritize risk*. arXiv preprint arXiv:2502.15865, <https://doi.org/10.48550/arXiv.2502.15865>.
- Cont, R. (2001). Empirical properties of asset returns: Stylized facts and statistical issues. *Quantitative Finance*, 1(2), 223–236, <https://doi.org/10.1080/713665670>
- Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, 13, 723–773, Retrieved from <https://jmlr.org/papers/v13/gretton12a.html>
- Harvey, C.R., & Liu, Y. (2020). False (and missed) discoveries in financial economics. *Journal of Finance*, 75(5), 2503–2553, <https://doi.org/10.1111/jofi.12951>
- Harvey, C.R., Liu, Y., Zhu, H. (2016). ... and the cross-section of expected returns. *Review of Financial Studies*, 29(1), 5–68, <https://doi.org/10.1093/rfs/hhv059>
- Khoroshilov, A., Chernysh, A., Ekhtibarov, O., Kamkia, N., Zmitrovich, D. (2026). *QuantCode-Bench: A benchmark for evaluating the ability of large language models to generate executable algorithmic trading strategies*. arXiv preprint arXiv:2604.15151, <https://doi.org/10.48550/arXiv.2604.15151>.
- Li, W., Kim, H., Cucuringu, M., Ma, T. (2025). *Can LLM-based financial investing strategies outperform the market in the long run?* arXiv preprint arXiv:2505.07078, <https://doi.org/10.48550/arXiv.2505.07078>.
- López de Prado, M. (2018). *Advances in financial machine learning*. Wiley.
- Lopez-Lira, A., & Tang, Y. (2023). *Can ChatGPT forecast stock price movements? Return predictability and large language models*. arXiv preprint arXiv:2304.07619, <https://doi.org/10.48550/arXiv.2304.07619>.
- Lopez-Paz, D., & Oquab, M. (2017). Revisiting classifier two-sample tests. *International Conference on Learning Representations (ICLR)*.
- McLean, R.D., & Pontiff, J. (2016). Does academic research destroy stock return predictability? *Journal of Finance*, 71(1), 5–32, <https://doi.org/10.1111/jofi>

- Sullivan, R., Timmermann, A., White, H. (1999). Data-snooping, technical trading rule performance, and the bootstrap. *Journal of Finance*, 54(5), 1647–1691, <https://doi.org/10.1111/0022-1082.00163>
- Wang, Z., Yang, W., Wu, Q., Ma, L., Zhang, Y., Jia, W. (2026). *BacktestBench: Benchmarking large language models for automated quantitative strategy back-testing*. arXiv preprint arXiv:2605.17937, <https://doi.org/10.48550/arXiv.2605.17937>.
- White, H. (2000). A reality check for data snooping. *Econometrica*, 68(5), 1097–1126, <https://doi.org/10.1111/1468-0262.00152>
- Yao, J., & Zheng, Z. (2026). *Beyond agent architecture: Execution assumptions and reproducibility in LLM-based trading systems*. arXiv preprint arXiv:2606.08285, <https://doi.org/10.48550/arXiv.2606.08285>.
- Zhang, M., Fu, J., Warrier, T., Wang, Y., Tan, T., Huang, K.-w. (2025). *FAITH: A framework for assessing intrinsic tabular hallucinations in finance*. arXiv preprint arXiv:2508.05201, <https://doi.org/10.48550/arXiv.2508.05201>.