# Can We Trust LLM-Generated Research Artifacts? A Freeze-and-Audit Protocol for LLM-Assisted Financial Research

Anonymous ACL submission

## Abstract

Can we trust research artifacts produced by LLM agents after repeated validation feedback? Even deterministic backtests become biased evidence when validation results steer what to try next. We study this reliability problem in quantitative financial research, a high-noise stress test with nonstationarity, leakage, transaction costs, and multiple testing. We propose SATS-QUANT, a Selection-Aware Testing and Scrutiny protocol for LLM-assisted adaptive research. Search-time screening may remain adaptive, but the evidence used to write the final trading claim is restricted to artifacts fixed before the final holdout, a predeclared backtest contract, and diagnostics over that fixed family. The contribution is disciplined claim formation, not a new forecasting architecture. In a retrospective score-family audit of fixed score mixtures, one $85/15$ Transformer/RD-Agent-generated expert mixture has higher descriptive holdout risk-adjusted information ratio than the Transformer anchor (5.042 vs. 3.317). In a prospectively locked five-seed targeted-search experiment, SATS-QUANT shows higher observed CV-selected holdout IR and better pre-holdout/holdout alignment, but uncertainty is large and endpoint dominance is not established. The audit maps adaptive-search findings to one of three outcomes: supported claim, downgrade, or abstain.

## 1 Introduction

LLM systems increasingly operate as *research agents*: they propose hypotheses, write code, run experiments, inspect logs, and adapt to select their next action from earlier outcomes. Agentic systems such as AIDE and AIRA treat machine-learning research as a search over executable artifacts rather than one-shot prediction (Jiang et al., 2025; Toledo et al., 2025). In quantitative finance, R&D-AGENT(Q) (Li et al., 2025) implements this pattern within a Qlib-based research loop. LLM modules propose factor or model hypotheses, generate executable code, and run it through Qlib (Yang et al., 2020), an open-source platform for deterministic backtests over historical market data. This raises a trust question: after an LLM-assisted loop has generated many artifacts and repeatedly observed validation results, which result, if any, is safe to claim?

The possible failures are broader than hallucination. RD-Agent-style systems can use deterministic validators such as Qlib backtests. But once validation feedback guides future hypotheses, code edits, or model choices, validation evidence is no longer independent. It has effectively selected among many tried candidates, like a test set repeatedly consulted during model development. Quantitative finance research is a useful stress test because time dependence, regime shifts, transaction costs, and multiple comparisons make attractive backtests easy to overstate.

**Core definitions and financial instance.** As a running example, suppose an agent has tried stock-ranking models and portfolio patches after seeing validation backtests. Before the final holdout, SATS-QUANT might freeze four Transformer/RD-Agent-generated expert score mixtures as the only artifacts allowed to compete for the final claim. This finite set is the *candidate family*. A *frozen* family is fixed before the final holdout is evaluated, so no new candidates, weights, dates, operators, or baselines may be added afterward. The frozen object may be model weights, family membership, or an exported score matrix. A *locked evaluation contract* fixes the financial backtest setup: data window, benchmark index, portfolio rule, transaction costs, metrics, and reporting code. A *claim-time audit* maps post-freeze evidence to *supported claim*, *downgrade*, or *abstain*.

Figure 1 shows an overview of the SATS-QUANT freeze-and-audit protocol. SATS-

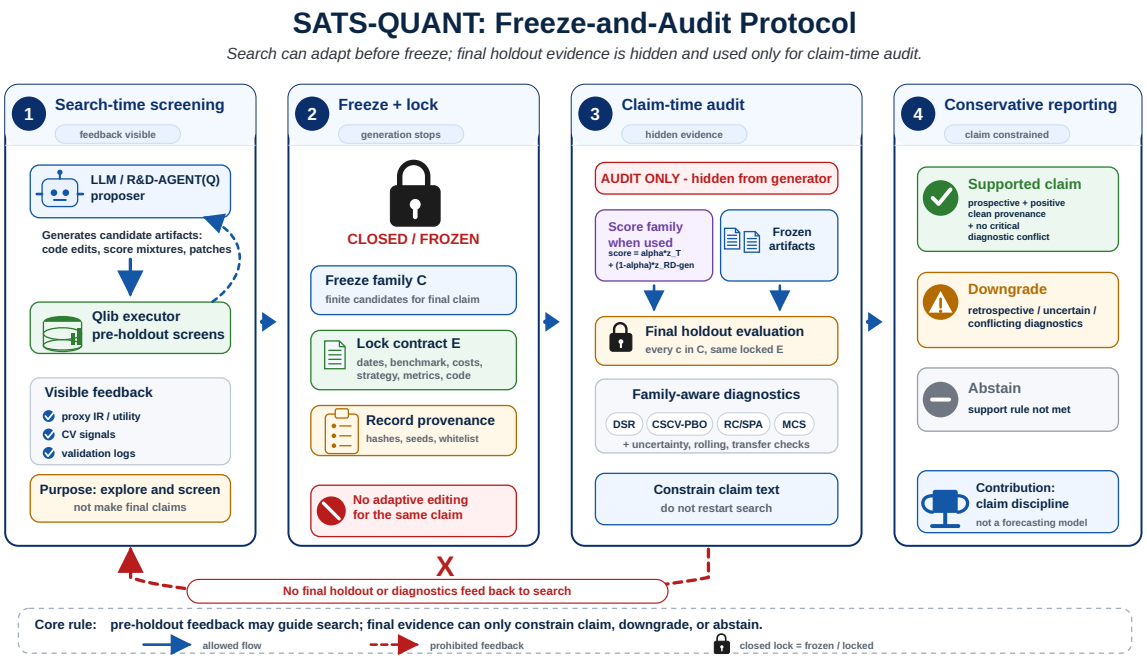


Figure 1: SATS-QUANT freeze-and-audit protocol. Search-time proxy/CV feedback may guide LLM/R&D-AGENT(Q) proposal generation before freeze. After the candidate family and evaluation contract are locked, final holdout evaluation and family-aware diagnostics are hidden from the generator and used only to constrain the reporting decision: supported claim, downgrade, or abstain. Section 4 states the main-body protocol; Algorithm 1 gives an appendix checklist.

QUANT addresses *claim formation*: search-time screening may remain adaptive, but final claims must be made against a frozen candidate family and locked evaluation contract. It is a claim-time audit protocol, not a forecasting architecture, not a replacement for Qlib or R&D-AGENT(Q), and not a claim that 85/15 is universal.

Table 1 states how the audit is applied in this paper. The two main evidence blocks have different candidate-generation mechanisms: one starts from fixed score mixtures, while the other uses an LLM patch proposer to generate portfolio candidates. The common part is what happens after freeze: the SATS-QUANT audit evaluates the locked holdout and diagnostics, then maps the evidence to an allowed claim. The claim rubric is conservative: a supported claim requires a prospective lock (a pre-holdout manifest fixing artifacts and reporting rules), a positive predeclared endpoint, clean provenance/leakage checks, and no critical diagnostic conflict. A critical conflict means predeclared diagnostics make the named winner unstable or non-confirmatory, such as high probability of backtest overfitting (PBO), low deflated Sharpe ratio (DSR), non-confirmatory familywise tests (RC/SPA), or rolling/transfer reversal.

**Contributions.** Beyond "use a holdout" or "run multiple-testing corrections," SATS-QUANT specifies the LLM-agent information boundary: when adaptive feedback stops, what remains hidden from the generator, which executable artifacts belong to the claim family, and how diagnostic conflicts change the claim text. We contribute: (i) a post-selection framing for trust in LLM-generated research artifacts; (ii) a freeze-and-audit contract separating Qlib execution, R&D-AGENT(Q) generation, and SATS-QUANT claim-time auditing; and (iii) a retrospective score-family audit plus a prospectively locked targeted-search experiment.

## 2 Preliminaries and System Stack

**Three-layer stack.** We organize the paper by a three-layer stack. R&D-AGENT(Q) is the *upstream generator*; Qlib is the *execution and backtesting platform*; and SATS-QUANT is the *claim-time audit layer*. RD-Agent's analysis and scheduler operate *inside* the search loop and can influence future experiments, whereas SATS-QUANT operates *after* freeze and decides whether to support, downgrade, or abstain from a claim. Table 2 summarizes the boundary; Appendix C gives the stage-by-stage mapping.

| Evidence block | Before freeze: candidate source | After freeze: SATS-QUANT audit and output |
|---|---|---|
| Retrospective score-family case study | No new search is run in this paper. The audited candidates are four fixed Transformer/RD-Agent-generated expert score mixtures: $\{50/50, 80/20, 85/15, 90/10\}$. | The audit evaluates the four mixtures under one locked Qlib contract and post-selection diagnostics. Because provenance is retrospective and diagnostics conflict, the output is downgrade to a descriptive conditional claim. |
| Prospectively locked targeted-search experiment | An LLM patch proposer generates portfolio candidates with three bounded patch operators: `CV_REFINE`, `STRATEGY_PATCH`, and `COST_PATCH` (bounded cost-field edits). The SATS-QUANT-style controller and RawIR control (raw information-ratio reward) differ only in pre-holdout reward. | The audit evaluates promoted candidates under pre-holdout manifests, the same locked holdout/reporting contract, and diagnostics. Because uncertainty is large and endpoint dominance is not established, the output is downgrade to a mixed selection-hygiene (alignment) claim. |
| Robustness diagnostics | No new candidates are generated. Rolling windows, stacking baselines, model-confidence-set (MCS) tests, transfer, and $2\times$-cost stress are post-hoc checks. | These checks test whether the two audited interpretations are stable. They are not used for selection, so they calibrate caveats rather than create a new claim. |

Table 1: How the same SATS-QUANT audit is applied to different evidence sources. Candidate generation differs before freeze; the shared method is the post-freeze audit that maps locked evidence to an allowed claim.

| System | When it operates | What feedback it sees | What it optimizes | Output | Failure mode controlled |
|---|---|---|---|---|---|
| R&D-AGENT(Q) | Inside the research loop | Intermediate metrics, prior proposals, and execution summaries | Generation of new quant artifacts | Qlib-compatible factor/model/workflow candidate | Search-space exploration; does not certify final claims |
| Qlib | During deterministic execution and backtesting | Fixed workflow, data, strategy, benchmark, and cost settings | Correct execution of one declared workflow | Daily score matrices, holdings, returns, and summary metrics | Not selection-aware by itself; execution substrate |
| SATS-QUANT | After candidate-family freeze | Frozen candidate pool, locked contract, pre-holdout summaries, final holdout results, and diagnostics | Credibility of the final claim | Supported claim / downgrade / abstain plus diagnostics and provenance | Post-selection overclaiming |

Table 2: Qlib, R&D-AGENT(Q), and SATS-QUANT boundary. The three systems operate at different times, see different feedback, and optimize different objects.

**Qlib workflow used here.** Qlib is the deterministic executor. A workflow specifies data, train/-validation/test segments, a model, prediction export, and portfolio-analysis records (Yang et al., 2020; Microsoft, 2026b,a). The Transformer anchor is the standard Qlib Transformer workflow over Alpha158, a hand-crafted price/volume feature set including lagged prices, returns, volume ratios, moving averages, and volatility-style indicators. Let $P_{t,i}^{\text{close}}$ be asset $i$'s closing price on day $t$. The supervised ranking target is

$$y_{t,i} = \frac{P_{t+2,i}^{\text{close}}}{P_{t+1,i}^{\text{close}}} - 1, \tag{1}$$

a one-period future close-to-close return. At inference time, the trained workflow exports one scalar score $\hat{s}_{t,i}^{(A)}$ per asset and day; larger scores mean the model ranks that asset as more attractive for the next holding period.

**R&D-AGENT(Q) WORKFLOW USED HERE.** R&D-AGENT(Q) is the upstream LLM-assisted research loop: LLM-facing modules propose hypotheses, convert them into Qlib-compatible workflow changes, run them in Qlib, summarize metrics, and schedule the next action (Li et al., 2025; Microsoft, 2026c). In the score-family case study, the *RD-Agent-generated expert* is a fixed Qlib-compatible neural workflow, not an interactive prompt at inference time. It uses a generic Qlib PyTorch wrapper (`GeneralPTNN`) with Alpha158 features and a 20-step feature sequence. The audit claim is about this frozen score-producing artifact, not every upstream R&D-AGENT(Q) trial; Appendix C states which stages are directly audited.

**Why combine the Transformer with an RD-Agent-generated expert?** This combination is an audit case study, not a repair mechanism for R&D-AGENT(Q). The Transformer is a strong non-LLM anchor; the RD-Agent-generated expert

is a fixed LLM-assisted artifact. Combining their normalized score matrices creates a small candidate family for asking whether the LLM-produced artifact adds *incremental ranking information* under one locked contract.

**From Qlib scores to audited claims.** The data path is: prices → features → expert scores → daily ranks → holdings → cost-adjusted returns → post-selection audit. In the score-family block, SATS-QUANT aligns the two frozen score matrices, standardizes each day's score vector within each expert, and forms a fixed mixture family. Qlib then evaluates every candidate under the same portfolio rule, benchmark, dates, and transaction-cost contract. The LLM or researcher may influence *candidate generation* before the freeze, but not the *evaluation contract* afterward. Appendix A gives a concrete toy walk-through and glossary for readers without quantitative-finance background.

## 3 Related Work

**LLM RESEARCH AGENTS AND AUTOMATED ML.** AIDE and AIRA frame LLM research as search over executable artifacts rather than one-shot prediction (Jiang et al., 2025; Toledo et al., 2025). MLE-bench evaluates agents on machine-learning engineering tasks (Chan et al., 2025), while AI-Scientist-style systems automate ideation, code execution, experiment analysis, and paper writing (Lu et al., 2024). Earlier AutoML systems searched over model and preprocessing pipelines (Feurer et al., 2015; Hutter et al., 2019). R&D-AGENT(Q) specializes this pattern to quantitative research by using LLM modules to manage Qlib-compatible factor/model workflows (Li et al., 2025). SATS-QUANT adds a claim-time audit boundary after such generation loops.

**Adaptive evaluation, leakage, and model-selection bias.** Repeated validation feedback connects this paper to adaptive data analysis and reusable-holdout work, which show how adaptively chosen analyses can invalidate ordinary holdout evidence (Dwork et al., 2015). The same reliability concern appears in model selection, where selection criteria can be overfit and downstream performance estimates biased (Cawley and Talbot, 2010; Varma and Simon, 2006). NLP benchmark-contamination studies emphasize a different leakage channel, but share the concern that benchmark scores can overstate generalization when evaluation evidence is not independent (Deng et al., 2024). Because the retrospective score-family result combines frozen expert scores, we also compare against forecast-combination (Bates and Granger, 1969) and stacking baselines (Wolpert, 1992; Breiman, 1996).

**Backtest overfitting and post-selection inference.** Established financial-econometrics tools for selection-aware backtest evaluation include DSR, CSCV-PBO, RC/SPA, and MCS (Bailey and López de Prado, 2014; Bailey et al., 2017; White, 2000; Hansen, 2005; Hansen et al., 2011). SATS-QUANT packages these tools as a protocol for LLM-assisted adaptive research: search may be adaptive, but the final claim must be made against a frozen family and locked contract.

**What SATS-QUANT adds.** Relative to ordinary holdout testing, SATS-QUANT fixes when adaptive feedback stops. Relative to preregistration or nested CV, it locks executable artifacts and action space, not only a split. Relative to standalone DSR/PBO/RC/SPA/MCS, it makes provenance and diagnostic conflicts binding on the reported claim rather than optional after-search commentary. Relative to R&D-AGENT(Q)'s analysis loop, it sits outside the scheduler: final holdout evidence cannot trigger more edits for the same claim.

## 4 Method

The protocol treats LLM proposals as hypotheses, not evidence. Quantitative improvement must be established through pre-holdout screening and post-freeze evaluation rather than through the agent's own assessment.

**SHARED FREEZE-AND-AUDIT PROTOCOL.** SATS-QUANT is the claim-time audit, not the candidate generator. Candidate generation can differ across studies; in this paper, the retrospective case audits fixed score mixtures, while the prospective case generates portfolio patches before the freeze. The shared audit has three phases. **(i) Upstream search-time screening:** an external generator, such as R&D-AGENT(Q) in our setting, may propose code edits, score mixtures, or portfolio patches. Qlib supplies pre-holdout proxy or cross-validation (CV) signals that may guide future search; a proxy is a cheap one-fold or partial backtest, while CV is a more expensive walk-forward evaluation. **(ii) Candidate-family freeze:** the candidate family, benchmark, holdout

dates, allowed operator scope, and reporting protocol are fixed. **(iii) Post-selection audit:** the frozen family is evaluated on the final holdout with ordinary baselines, rolling windows, and family-aware diagnostics. The audit returns one of three reporting decisions: supported claim, downgrade, or abstain; it abstains if no reliable winner remains. This paragraph, Figure 1, and Table 1 are the self-contained main-body protocol; Appendix B gives an implementation checklist.

**Instantiation 1: fixed score-family candidates.** For expert $e \in \{A, B\}$, let $X_{t,i}^{(e)}$ be the Qlib feature window for asset $i$ at decision date $t$. Let $f_e(\cdot;\theta_e)$ be the frozen trained predictor. The exported score is $\hat{s}_{t,i}^{(e)} = f_e(X_{t,i}^{(e)};\theta_e)$. Appendix D gives the Transformer and RD-Agent-generated expert score semantics. Because raw score scales differ, SATS-QUANT normalizes each expert's daily score vector across the active tradable universe $\mathcal{U}_t$,

$$z_{t,i}^{(e)} = \frac{\hat{s}_{t,i}^{(e)} - \mu_t^{(e)}}{\sigma_t^{(e)} + \epsilon}, \quad \mu_t^{(e)} = |\mathcal{U}_t|^{-1} \sum_{j \in \mathcal{U}_t} \hat{s}_{t,j}^{(e)}, \tag{2}$$

where $\sigma_t^{(e)}$ is the cross-sectional (across-asset) standard deviation and $\epsilon > 0$ prevents division by zero. Then it forms the frozen family:

$$\begin{aligned} \hat{s}_{t,i}^{(\alpha)} &= \alpha z_{t,i}^{(A)} + (1-\alpha) z_{t,i}^{(B)}, \\ \alpha &\in \Lambda = \{0.50, 0.80, 0.85, 0.90\}. \end{aligned} \tag{3}$$

The $50/50$ row is an equal-weight sanity check, while $80/20$, $85/15$, and $90/10$ sample an anchor-heavy region where the Transformer remains the dominant score and the RD-Agent-generated expert acts as a bounded auxiliary signal. The primary row uses $A =$ Qlib Transformer Alpha158, $B =$ RD-Agent-generated expert, and $\alpha = 0.85$. No model is retrained; only the inference-time ranking signal changes. A small auxiliary-expert weight can still matter because portfolio construction is nonlinear: small score changes near the top-$k$ or drop/add boundary can change which assets are bought and sold. We do not add a finer post-hoc grid after observing the primary holdout because that would define a new selection family rather than audit the fixed one.

**Portfolio rule and returns.** Qlib applies `TopkDropoutStrategy`. Despite the name, this is a portfolio replacement rule, not neural-network dropout. Let $q_t(i)$ be the rank induced by $\hat{s}_{t,\cdot}$, where rank 1 is best. With `topk`=50 and `n_drop`=5, the strategy holds the top-ranked assets and replaces up to five deteriorated holdings. Let $H_t$ be the post-update holding set. Selected assets are equally weighted: $w_{t,i} = 1/|H_t|$ for $i \in H_t$ and $w_{t,i} = 0$ otherwise. In the primary contract, $|H_t| \leq 50$. If $r_{t+1,i}$ is the close-to-close return after day $t$, the gross portfolio return is

$$R_{t+1}^{\text{gross}} = \sum_{i \in \mathcal{U}_t} w_{t,i} r_{t+1,i}, \tag{4}$$

and the net return subtracts turnover costs,

$$R_{t+1} = R_{t+1}^{\text{gross}} - K_{t+1}, \qquad E_{t+1} = R_{t+1} - B_{t+1}, \tag{5}$$

where $B_{t+1}$ is benchmark return and $K_{t+1}$ is the Qlib transaction-cost deduction from turnover. Subtracting $B_{t+1}$ makes the audit about active value beyond broad market movement. Let $A_y = 238$, the Chinese-trading-calendar convention used here. For a holdout of $T$ days, we report information ratio $\text{IR} = \sqrt{A_y}\,\bar{E}/\text{sd}(E)$, annualized return $\text{ARR} = (\prod_{t=1}^{T}(1+R_t))^{A_y/T} - 1$, and maximum drawdown $\text{MDD} = \min_t(V_t/\max_{\tau \leq t} V_\tau - 1)$ for cumulative wealth $V_t = \prod_{\tau \leq t}(1+R_\tau)$. Here $\bar{E}$ and $\text{sd}(E)$ are the mean and standard deviation of daily excess returns. MDD is a negative peak-to-trough drawdown; values closer to zero are less severe.

**Post-selection diagnostics.** Each diagnostic answers a different question. DSR asks whether a Sharpe/IR-style statistic remains impressive after correcting for repeated trials and non-normal returns (Bailey and López de Prado, 2014). CSCV-PBO asks how often an in-sample winner falls into the worse half out of sample. RC/SPA ask whether the best candidate in the frozen family beats the Transformer anchor after accounting for multiple tried candidates. MCS asks which candidates remain statistically plausible winners. These diagnostics are conservative by design: an attractive backtest may remain descriptive rather than claim-supporting. Exact formulas and settings appear in Appendix F.

**Instantiation 2: prospective targeted-search candidates.** The following operators are used only to generate candidates for the prospective targeted-search experiment; they are not part of the general SATS-QUANT audit rule and are not used in the retrospective score-family case study. Both

comparison controllers use the same LLM patch proposer and validator; only the pre-holdout reward used to rank and promote candidates differs. The LLM receives a candidate summary, proxy/CV metrics when available, an operator name, allowed file paths, and forbidden edits. The validator rejects out-of-scope proposals before backtesting, so action-space violations cannot silently change the locked contract.

This candidate-generation experiment defines three operators:

- **`CV_REFINE`**: a stability-oriented refinement request that changes at most two TopkDropout parameters (e.g., `topk`, `n_drop`).
- **`STRATEGY_PATCH`**: a direct bounded portfolio-strategy patch over the same parameter family.
- **`COST_PATCH`**: edits only `open_cost`, `close_cost`, and `min_cost` within predeclared finite ranges around the standard Qlib cost fields; the exact ranges and rationale are in Section 5.

For example, replacing `topk` is valid for the first two operators, while changing dates, benchmark, model class, data loading, metric code, or the reporting protocol is rejected. Patch collapse measures duplicate-patch concentration as $1-\#\text{unique patch fingerprints}/\#\text{candidates}$, where a fingerprint is a normalized representation of the edited fields and values; high collapse means the search is less diverse than the proposal count suggests.

In the targeted-search experiment, the SATS-QUANT-style controller ranks candidates by pre-holdout IR after penalizing fold dispersion and drawdown. After CV summaries are available, it can also use pre-holdout DSR/PBO-style screening quantities. The RawIR control instead ranks candidates by raw pre-holdout IR. These rewards steer only upstream promotion. Canonical CSCV-PBO, RC/SPA, MCS, and final-holdout results remain claim-time diagnostics and are never fed back into proposal generation. Appendix B gives the exact reward formulas used only for this targeted-search candidate-generation experiment.

## 5 Experimental Setup and Provenance

The case study uses Chinese daily Qlib data: the standard Transformer Alpha158 setup, the CSI300 instrument universe, and `SH000300`, Qlib's CSI 300 benchmark code. The trained experts are frozen before score mixing. The diagnostic/CV window is 2017-08-09 to 2020-07-31, and the primary holdout is 2020-08-03 to 2020-09-23 (38 trading days). The diagnostic/CV window may include dates in-sample for one or both frozen experts, so it characterizes selection risk within the four-row family rather than independent generalization. A disjoint COVID-crash transfer holdout runs from 2020-02-03 to 2020-03-25. The trading contract uses daily close execution, `topk`=50, `n_drop`=5, limit threshold 0.095, open cost $5e^{-4}$, close cost $1.5e^{-3}$, and minimum cost 5. The limit threshold is Qlib's filter for stocks near China's daily price limits.

The score-family audit evaluates the Transformer anchor, RD-Agent-generated expert, fixed mixtures, robustness rows ($2\times$ cost and `topk`=30/`n_drop`=10), and ordinary score-combination baselines: validation-selected convex grid search, rank averaging, information-coefficient (IC)-weighted averaging, ridge stacking, and non-negative least squares (NNLS).

The targeted-search experiment compares two matched candidate generators over the same operator family. Both use the same LLM patch proposer and validator; only the pre-holdout reward differs. One controller uses a stability-aware SATS-QUANT-style reward, and RawIR is the control reward using raw information ratio alone. The operational prospective endpoint is holdout IR of the candidate selected by pre-holdout CV reward. Alignment metrics are secondary selection-hygiene diagnostics, and best-promoted/top-3 IR are analysis-only holdout-oracle diagnostics that use final holdout outcomes.

Both controllers use the same `deepseek/deepseek-v3.2` patch proposer (temperature 0.2, maximum 800 completion tokens, 60s timeout). The checker parses JSON patches, enforces target files, allowed keys, and numeric ranges, attempts one schema/path repair, and rejects any remaining invalid patch before backtesting; no manual repair is used. Each controller runs five matched seeds, with 30 proposals and three promoted candidates per seed, so each controller has 150 proposed and 15 promoted candidates. One SATS-QUANT candidate failed before usable evaluation; RawIR had none.

The proxy screen is one pre-holdout chronological CV-fold-pair backtest of about 126 trading days. Promotion then uses up to six chronological CV folds with a two-trading-day purge (drop overlapping-label training observations) and em-

bargo (drop observations immediately after validation), reducing label overlap and short-range temporal leakage. For `COST_PATCH`, declared ranges are `open_cost` $\in [0.0001, 0.0015]$, `close_cost` $\in [0.0005, 0.0030]$, and `min_cost` $\in [1, 20]$. These guardrails surround Qlib's standard contract $(0.0005, 0.0015, 5)$ while forbidding zero, negative, or unbounded cost edits. Because bounded `COST_PATCH` changes declared cost fields, this experiment is matched operator-search evidence, not fixed-cost-contract evidence.

**Provenance boundary.** The score-family case study is a *retrospective audit*: the four-row candidate family is auditable, but there is no pre-holdout registration record proving that the $85/15$ row was fixed before primary-holdout access. Wider upstream exploration also cannot be ruled out, so the row is descriptive rather than prospective evidence. The targeted-search experiment is different: before holdout evaluation, local SHA-256 manifests record configuration files, seeds, operator whitelist, source/report scripts, and planned promotion/reporting procedure. This is not third-party timestamping, but it creates a reproducible local lock; only this experiment can support direct adaptive-search claims. Appendix E and Appendix J give the lock checklist and action boundary.

**Ordinary score-combination baselines.** We include these baselines to test whether the fixed SATS-QUANT row is merely a learned ensemble weight. All baselines are fit on the validation window and evaluated once on the locked holdout; Appendix G gives the formulas and fitting details.

**Reading the result sections.** Results I and II are not two different SATS-QUANT methods. They are two applications of the same post-freeze audit to the two candidate sources in Table 1. In each case, the result section answers the same audit questions: What was the holdout signal? What provenance or diagnostic conflicts appeared? Which claim outcome does the SATS-QUANT audit allow?

## 6 Results I: Frozen Score-Family Case Study

Table 3 summarizes selected rows; Figure 2 shows the full fixed-mixture family. The best fixed $0.85/0.15$ mixture has higher holdout IR than the Transformer anchor (5.042 vs. 3.317). It is not matched by the ordinary score-combination baselines considered: the strongest, validation-selected convex grid search, reaches IR 3.499, while rank averaging, IC weighting, ridge stacking, and NNLS remain below the fixed $85/15$ row. Ridge is negative because its validation-fitted map fails on the holdout. Repeated frozen rows are deterministic reruns, not independent retraining replicates.

| Method | Holdout IR | ARR | MDD | Role in claim |
|---|---|---|---|---|
| Qlib Transformer Alpha158 | 3.317 | 0.260 | -0.012 | Anchor |
| RD-Agent-generated expert | 3.001 | 0.187 | -0.009 | Auxiliary expert |
| SATS-QUANT score 50/50 | 3.712 | 0.246 | -0.007 | Fixed mixture row |
| SATS-QUANT score 85/15 | **5.042** | **0.347** | -0.010 | Best fixed row |
| Validation-selected convex grid | 3.499 | 0.235 | -0.008 | Strongest ordinary baseline |
| Rank averaging | 2.703 | 0.192 | -0.010 | Baseline |
| IC-weighted averaging | 2.824 | 0.211 | -0.010 | Baseline |
| Ridge stacking | -5.268 | -0.372 | -0.060 | Baseline |
| NNLS stacking | 3.317 | 0.260 | -0.012 | Baseline |

Table 3: Selected score-family rows and ordinary baselines on the same frozen expert artifacts. Figure 2 shows the full fixed family.

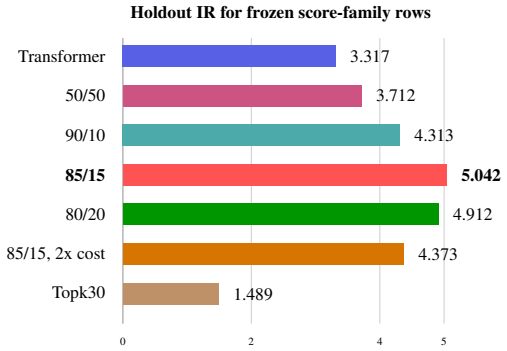


Figure 2: Full frozen score-family audit. The fixed $85/15$ row is strongest on the primary holdout, while neighboring rows and stress tests support a region-level, not universal-optimum, interpretation.

The primary holdout has only 38 trading days, so uncertainty is large. A paired moving-block bootstrap gives wide individual IR intervals: $[-0.627, 9.666]$ for the Transformer and $[0.112, 13.657]$ for $85/15$. The paired IR-gain interval is $[0.029, 4.926]$, while the paired ARR-gain interval crosses zero, $[-0.016, 0.201]$. We therefore use the score-family result as descriptive evidence with audit conflicts, not as a confirmatory performance claim.

The stronger $85/15$ row is consistent with partial complementarity rather than duplication. On the primary holdout, the Transformer and RD-Agent-generated expert daily excess returns have correlation $0.502$, and their mean daily cross-sectional score correlations are about $0.29$. Their excess-return signs disagree on 17 of 38 days. This supports the narrower claim that the auxiliary expert

contributes non-identical ranking information under the locked contract.

The audit constrains interpretation. CSCV-PBO is high (0.571), DSR is low (0.228 raw family size, 0.250 after correlation adjustment), and RC/SPA are non-confirmatory ($p = 0.225/0.177$). The best mixture changes across rolling windows, and the COVID transfer result reverses the primary ranking. **Audit outcome: downgrade.** Under the SATS-QUANT audit rule, the allowed claim is descriptive and conditional: under one locked contract, a Transformer-dominant fixed-mixture region can show higher holdout IR than the anchor, but the effect is power-limited and regime-sensitive. Appendix H gives supporting diagnostic tables.

## 7 Results II: Prospectively Locked Targeted Search

The direct adaptive-search test compares two matched candidate generators over the same portfolio-targeted operator family. Both generators are later audited by the same SATS-QUANT rule; before freeze, they differ only in whether promotion uses a stability-aware SATS-QUANT-style reward or RawIR. Because the operator family includes bounded `COST_PATCH`, the experiment is matched operator-search evidence rather than fixed-cost-contract evidence. Table 4 shows a mixed result.

Here *selection hygiene* means better alignment between pre-holdout selection signals and final holdout outcomes; *endpoint dominance* means clear superiority on final holdout endpoint metrics. SATS-QUANT does *not* improve best-promoted holdout IR or mean top-3 promoted IR, which are holdout-oracle diagnostics rather than deployable selection rules. On the operational CV-selected endpoint, observed holdout IR is higher ($1.371 \rightarrow 1.673$); CV top-1 accuracy rises from 0.000 to 0.400, proxy/holdout correlation from 0.200 to 0.400, and CV/holdout correlation from $-0.600$ to 0.300.

Correlations are Spearman rank correlations between pre-holdout screening scores and holdout IR over promoted candidates. CV top-1 accuracy is the fraction of seeds where the CV-selected candidate is also best by holdout IR. Promoted-set PBO is low-resolution over three promoted candidates. Its increase ($0.229 \rightarrow 0.383$), the slightly higher invalid rate, and lower proxy top-1 accuracy argue against a broad reliability win. The paired SATS-minus-RawIR mean difference on CV-selected IR is 0.302 with five-pair $t$ interval $[-0.447, 1.052]$ and sign-flip $p = 0.250$; seed 0 is an effective tie. **Audit outcome: downgrade to a mixed selection-hygiene claim.** Under the SATS-QUANT audit rule, endpoint dominance is not established.

| Metric | RawIR | SATS-QUANT |
|---|---|---|
| CV-selected IR | 1.371 | **1.673** |
| Proxy-selected IR | 1.945 | 1.683 |
| Best-promoted IR | 2.517 | 2.398 |
| Mean top-3 IR | 1.786 | 1.754 |
| Promoted-set PBO | 0.229 | 0.383 |
| Invalid rate | 0.000 | 0.007 |
| Patch collapse | 0.487 | 0.467 |
| Proxy/holdout $\rho$ | 0.200 | 0.400 |
| CV/holdout $\rho$ | -0.600 | **0.300** |
| CV top-1 accuracy | 0.000 | **0.400** |

Table 4: Prospective targeted-search summary. The operational endpoint is CV-selected IR. CV/proxy-selected metrics use pre-holdout selection; Best and Top-3 are holdout-oracle diagnostics, not deployable selection rules.

We therefore interpret this experiment as possible *selection hygiene under a CV-mediated promotion rule*, not broad adaptive-search superiority. Appendix I gives seed rows and Figure 3.

## 8 Discussion and Conclusion

The paper asks when an LLM-generated research artifact can be trusted enough to support a claim. The answer is not supplied by the LLM or by a single attractive backtest. Without SATS-QUANT, the fixed 85/15 row could be reported as a discovered improvement; with SATS-QUANT, it is downgraded because provenance, PBO, DSR, RC/SPA, rolling windows, and transfer evidence conflict. The prospective experiment is mixed: SATS-QUANT shows higher observed CV-selected holdout IR and alignment signals but not endpoint dominance. The reusable lesson is not "SATS finds better strategies." It is to separate adaptive search from claim formation, freeze the family before the final holdout, and report conflicting diagnostics.

The same boundary applies beyond finance: in prompt search, code-agent benchmarks, AutoML, and LLM-as-scientist loops, development scores are search feedback and hidden benchmark results should only constrain the final claim.

## Limitations

All empirical evidence comes from one domain, Chinese-equity quantitative research, and one stack,

Qlib plus RD-Agent-style generation. The primary holdout has only 38 trading days, and the rolling windows are also short. The diagnostics are therefore cautionary rather than decisive. The retrospective score-family result lacks prospective preregistration. It relies on frozen expert score matrices and deterministic reproducibility checks, not independent retraining variance. The direct adaptive-search experiment is prospectively locked but has only five matched seeds, includes a declared bounded cost-patch operator, and remains mixed. SATS-QUANT itself can fail in three ways: (i) the freeze happens after hidden holdout access, (ii) the action space silently alters the evaluation contract, or (iii) diagnostics are treated as proof rather than evidence constraints. Finally, the paper studies research-evaluation methodology, not deployable financial advice.

## Ethical Considerations

This work studies an evaluation methodology for LLM research agents. The main risk is that automated research systems can make overfit strategies appear credible. The purpose of SATS-QUANT is to expose, rather than hide, selection risk. Any public artifact release should include provenance information, configuration and result hashes, negative results, and clear warnings about the limits of backtests. Given a compatible Qlib data installation, the released report scripts and permitted frozen score/return artifacts are intended to regenerate the reported tables. Where redistribution is restricted, hashes and regeneration instructions should be provided instead. Artifact-use terms are handled as follows. Qlib and RD-Agent are cited existing software artifacts, the market data are used through a local Qlib data installation and are not redistributed, and the hosted LLM patch proposer is used through OpenRouter/DeepSeek access terms. Any released scripts or artifacts should include explicit license or access terms where release is permitted.

We have used an AI assistant for writing support (e.g., editing for clarity and suggesting phrasing/organization). All technical content, experiments, and reported results were produced and verified by the authors.

## References


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

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

John M Bates and Clive WJ Granger. 1969. The combination of forecasts. *Journal of the operational research society*, 20(4):451–468.

Leo Breiman. 1996. Stacked regressions. *Machine learning*, 24(1):49–64.

Gavin C Cawley and Nicola LC Talbot. 2010. On overfitting in model selection and subsequent selection bias in performance evaluation. *The Journal of Machine Learning Research*, 11:2079–2107.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Aleksander Madry, and Lilian Weng. 2025. Mle-bench: Evaluating machine learning agents on machine learning engineering. In *International Conference on Learning Representations*, volume 2025, pages 50466–50494.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2024. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8706–8719, Mexico City, Mexico. Association for Computational Linguistics.

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

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. 2015. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Peter R Hansen, Asger Lunde, and James M Nason. 2011. The model confidence set. *Econometrica*, 79(2):453–497.

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

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. 2019. *Automated machine learning: methods, systems, challenges*. Springer.

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. 2025. Aide: Ai-driven exploration in the space of code. *Preprint*, arXiv:2502.13138.

Yuante Li, Xu Yang, Xiao Yang, Xisen Wang, Weiqing Liu, and Jiang Bian. 2025. R&d-agent-quant: A multi-agent framework for data-centric factors and model joint optimization. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc.

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

Microsoft. 2026a. Qlib documentation: Portfolio strategy. https://qlib.readthedocs.io/en/stable/component/strategy.html. Accessed 2026-05-08.

Microsoft. 2026b. Qlib documentation: Workflow management. https://qlib.readthedocs.io/en/stable/component/workflow.html. Accessed 2026-05-08.

Microsoft. 2026c. Rd-agent documentation: Finance quant agent. https://rdagent.readthedocs.io/en/latest/scens/quant_agent_fin.html. Accessed 2026-05-08.

Edan Toledo, Karen Hambardzumyan, Martin Josifoski, Rishi Hazra, Nicolas Baldwin, Alexis Audran-Reiss, Michael Kuchnik, Despoina Magka, Minqi Jiang, Alisia Lupidi, Andrei Lupu, Roberta Raileanu, Tatiana Shavrina, Kelvin Niu, Jean-Christophe Gagnon-Audet, Michael Shvartsman, Shagun Sodhani, Alexander Miller, Abhishek Charnalia, and 6 others. 2025. Ai research agents for machine learning: Search, exploration, and generalization in mle-bench. In *Advances in Neural Information Processing Systems*, volume 38, pages 35309–35348. Curran Associates, Inc.

Sudhir Varma and Richard Simon. 2006. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7(1):91.

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

David H Wolpert. 1992. Stacked generalization. *Neural networks*, 5(2):241–259.

Xiao Yang, Weiqing Liu, Dong Zhou, Jiang Bian, and Tie-Yan Liu. 2020. Qlib: An ai-oriented quantitative investment platform. *Preprint*, arXiv:2009.11189.

## A Worked Example and Glossary

This appendix is illustrative: it does not introduce new evidence, and it does not replace the exact Qlib contracts used in the experiments. Its purpose is to make the finance and Qlib terminology concrete for readers whose background is NLP or ML evaluation.

**A toy end-to-end audit.** Suppose an upstream LLM-agent loop in the style of R&D-AGENT(Q) receives a bounded request:

```
Given the candidate Qlib workflow summary and
    pre-holdout CV metrics,
propose one JSON patch inside the allowed action
     space. You may edit
TopkDropout topk/n_drop or declared cost fields.
     Do not edit dates,
benchmark, data loading, metric code, or
    reporting code.
```

This request fragment illustrates the prospective targeted-search experiment: the agent output is a bounded portfolio patch. The separate score-family case study instead freezes already exported Transformer and RD-Agent-generated expert score matrices and audits mixtures of them. In both cases, Qlib executes the fixed artifact deterministically: it loads price/volume features, applies the trained predictor, exports one score per asset per day, forms holdings from ranks, subtracts transaction costs, and reports returns. SATS-QUANT does not ask the LLM whether the result is good; it freezes the candidate family and evaluates the final claim under the locked contract.

For a toy score-family day with three stocks, assume daily normalized Transformer scores are $(1.2, 0.3, -0.6)$ and normalized RD-Agent-generated expert scores are $(0.1, 1.0, -0.8)$. Equation 3 with $\alpha = 0.85$ gives mixed scores $(1.035, 0.405, -0.630)$, so the first stock ranks highest. The real experiment uses the full CSI300 universe, not three stocks, and Qlib's `TopkDropoutStrategy` holds 50 names and can replace up to five deteriorated holdings per day. The point is the same: a small auxiliary-expert weight can change ranks near the buy/sell boundary without retraining either model.

If the selected portfolio returns $0.4\%$ on the next day and the CSI300 benchmark returns $0.1\%$, the benchmark-relative excess return is $0.3\%$. This subtraction prevents the artifact from receiving credit for market-wide gains that the benchmark also captured. Repeating this over the holdout gives the excess-return series used for IR, RC/SPA, MCS, and the other diagnostics. If the attractive candidate has weak provenance, wide uncertainty, or conflicting diagnostics, SATS-QUANT reports a downgraded claim rather than treating the backtest as an independent discovery.

**Glossary for non-finance readers.**

**Artifact.** An executable research object: here, a Qlib model workflow, score matrix, score mixture, or bounded portfolio patch.

**Score matrix.** A table indexed by date and stock. Entry $\hat{s}_{t,i}$ is the model's scalar ranking score for stock $i$ on date $t$.

**Close-to-close return.** Price change from one market close to a later market close; for example, $101/100 - 1 = 1\%$.

**Qlib label.** The supervised target used to train the predictors; in this paper `Ref($close,-2)/Ref($close,-1)-1` denotes the one-period future close-to-close return in Eq. 1.

**Alpha158.** Qlib's hand-crafted 158-feature price/volume representation, including lagged prices, returns, volume ratios, moving averages, and volatility-style indicators.

**Transformer anchor.** The fixed non-LLM Qlib Transformer workflow used as the stable reference model.

**RD-Agent-generated expert.** The fixed LLM-assisted Qlib workflow audited as the auxiliary artifact; it exports scores like any other Qlib model.

**GeneralPTNN.** Qlib's generic PyTorch-neural-network wrapper for generated model code; it is an integration wrapper, not the audit method.

**TopkDropout.** Qlib portfolio rule that holds the top-ranked names and replaces up to `n_drop` weak holdings with better-ranked names.

**Turnover.** The change in portfolio holdings between consecutive days; higher turnover induces larger transaction-cost deductions.

**Benchmark return.** Return of the reference index, here Qlib's CSI300 benchmark code `SH000300`; it is the market baseline the strategy must beat.

**Excess return.** Portfolio return minus benchmark return; this measures active value beyond

market-wide movement and is the daily series audited for relative performance.

**Transaction costs.** Deductions for trading, including buy-side open cost, sell-side close cost, and minimum commission or fee.

**Limit threshold.** Qlib's approximation to Chinese daily price-limit behavior; it constrains trading when prices are near limit-up or limit-down states.

**IR.** Information ratio: annualized mean excess return divided by daily excess-return volatility, a signal-to-noise style metric.

**ARR.** Annualized return: the compounded portfolio return rescaled to a trading-year convention.

**MDD.** Maximum drawdown: the worst peak-to-trough wealth decline; in this paper it is reported as a negative number.

**Proxy.** A cheap pre-holdout screen, such as one chronological validation fold or partial backtest.

**CV.** Cross-validation; here, a more expensive pre-holdout walk-forward evaluation over chronological folds.

**RawIR.** A targeted-search control reward that ranks candidates by raw pre-holdout information ratio without the SATS-QUANT stability penalties.

**Holdout.** The final evaluation window hidden from adaptive search and used only after the family is frozen.

**Holdout-oracle diagnostic.** A diagnostic that uses final holdout outcomes, such as best-promoted IR; useful for analysis but not available for deployable selection.

**Patch fingerprint.** A normalized representation of the edited fields and values in a proposal, used to count duplicate or near-duplicate patches.

**Invalid patch.** A proposed edit that fails schema, path, range, or execution checks before producing usable evaluation results.

**Selection hygiene.** Evidence that pre-holdout selection signals better align with final holdout outcomes, e.g., higher CV/holdout rank correlation or CV top-1 accuracy.

**Endpoint dominance.** Clear superiority on final endpoint metrics; this paper explicitly does not establish it for the prospective targeted-search experiment.

**DSR.** Deflated Sharpe Ratio; asks whether an IR/Sharpe-like statistic remains impressive after repeated trials and non-normal returns.

**CSCV-PBO.** Combinatorially symmetric cross-validation Probability of Backtest Overfitting; asks how often an in-sample winner falls into the worse half out of sample.

**RC/SPA.** White's Reality Check and Hansen's Superior Predictive Ability tests; familywise tests of whether the best candidate beats the anchor after multiple tried candidates.

**MCS.** Model Confidence Set; returns a set of statistically plausible winners rather than forcing a single winner.

**COST_PATCH.** A declared targeted-search operator that edits only bounded cost fields. Because it changes declared candidate settings, the prospective experiment is matched operator-search evidence, not the fixed-cost score-family contract.

## B Protocol Algorithm

| Audit point | Used for | Can affect future search? |
|---|---|---|
| Proxy IR / proxy utility | Fast screening | Yes |
| Walk-forward CV utility | Promotion decision | Yes |
| Canonical CSCV-PBO | Post-selection overfit diagnostic | No |
| DSR sensitivity | Post-selection selection-risk diagnostic | No |
| RC/SPA familywise tests | Final anchor comparison | No |
| MCS | Candidate-region interpretation | No |
| Final holdout | Final reporting only | No |

Table 5: Search-time screening is separate from post-selection audit.

Table 6 maps the checklist to the two evidence blocks used in the paper.

| Audit component | Retrospective score-family block | Prospective targeted-search block |
|---|---|---|
| Candidate family $\mathcal{C}$ | Four fixed score mixtures: 50/50, 80/20, 85/15, 90/10 | Promoted patch candidates generated before holdout evaluation |
| Upstream search step | No new search is run in this paper; Step 1 is skipped | LLM patch proposer uses only proxy/CV feedback before freeze |
| Freeze evidence | Fixed score matrices, mixture weights, dates, benchmark, costs, metrics, and reporting code | SHA-256 manifests for seeds, operator whitelist, configs, scripts, and planned reporting |
| Locked evaluation | Evaluate every fixed mixture under the same Qlib contract | Evaluate promoted candidates under the same locked holdout/reporting contract |
| Post-freeze evaluation | PBO, DSR, RC/SPA, MCS, rolling windows, transfer, and uncertainty | CV-selected holdout IR, alignment diagnostics, promoted-set PBO, invalid rate, patch collapse, and uncertainty |
| Audit output in this paper | Downgrade to descriptive conditional claim | Downgrade to mixed selection-hygiene claim; no endpoint dominance |

Table 6: Concrete application of Algorithm 1 to the two evidence blocks. The candidate-generation mechanisms differ, but both blocks use the same post-freeze audit logic.

**Algorithm 1** SATS-QUANT freeze-and-audit protocol

**Require:** Candidate family $\mathcal{C}$ from frozen artifacts or a permitted upstream generator; Qlib-compatible locked evaluation contract; final holdout; diagnostics; claim rules.

If upstream search is used, the generator may use only permitted Qlib proxy/CV feedback (Table 5); candidate-generation rules must be fixed before final holdout evaluation.
Record the search/audit boundary: final holdout results and claim-time diagnostics are hidden from the generator.
Freeze $\mathcal{C}$, dates, benchmark, portfolio rule, costs, metrics, baselines, and reporting code.
**for** each $c \in \mathcal{C}$ **do**
    Evaluate $c$ under the locked contract; compute returns by Eq. 5 and headline metrics.
**end for**
Compute the predeclared post-selection diagnostics in Appendix F.
Do not feed final holdout results or claim-time diagnostics back into the same claim.
**if** prospective lock, positive predeclared endpoint, and no critical diagnostic conflict **then**
    **return** supported claim.
**else if** retrospective provenance, large uncertainty, or conflicting diagnostics **then**
    **return** downgraded descriptive or conditional claim.
**else**
    **return** abstain.
**end if**

**Targeted-search reward details.** Equations 6–7 are pre-holdout controller formulas for the prospective targeted-search experiment only; they are not the core SATS-QUANT audit rule:

$$U_{\text{proxy}}(a) = \mu_{\text{IR}}(a) - 0.25\sigma_{\text{IR}}(a) - 2DD95(a), \tag{6}$$

$$\begin{aligned} U_{\text{CV}}(a) = \mu_{\text{IR}}(a) &- 0.25\sigma_{\text{IR}}(a) - 2DD95(a) \\ &+ 0.5DSR_{\text{CV}}(a) \\ &+ 0.5(1 - PBO_{\text{CVpool}}(a)). \end{aligned} \tag{7}$$

Here $\mu_{\text{IR}}(a)$ and $\sigma_{\text{IR}}(a)$ are the mean and standard deviation of available fold IRs, and $DD95(a)$ is the 95th percentile across available folds of the fold maximum-drawdown magnitudes. $DSR_{\text{CV}}(a)$ and $PBO_{\text{CVpool}}(a)$ are best-effort pre-holdout quantities; if unavailable or constant, ordering is driven by the fold-level terms. The coefficients are predeclared constants that put dispersion and drawdown penalties on the same reward scale as IR; they are not learned weights or holdout-fitted parameters.

## C Qlib and R&D-Agent(Q) Stage Mapping

This appendix clarifies which parts of the audited workflow belong to R&D-AGENT(Q), Qlib, and SATS-QUANT. Table 7 gives the stage map. Phase (i) in Section 4 is the upstream R&D-AGENT(Q)/Qlib research loop being audited; the distinctive SATS-QUANT intervention is the freeze and claim-time audit boundary.

| Functional stage | RD-Agent-side role | Qlib-side role | Where used in this paper | SATS boundary |
|---|---|---|---|---|
| Hypothesis or artifact proposal | LLM-facing modules propose factors, model/workflow changes, or portfolio patches from previous summaries | No final evaluation; only supplies the executable environment | Fixed RD-Agent-generated expert; prospective targeted-search patch proposer | Search-time only; final holdout remains hidden |
| Implementation and wrapping | Generated code/config is converted into a Qlib-compatible workflow, e.g., a `GeneralPTNN` neural workflow | Loads data handler, model class, strategy, and configuration | Fixed RD-Agent-generated expert and bounded operator patches | Candidate identity, editable files, and operator scope must be recorded |
| Pre-holdout screening | Agent or scheduler may use summaries to choose the next action | Runs proxy or CV backtests and exports scores/metrics | Targeted-search proxy/CV rewards in Eqs. 6–7 | Proxy/CV feedback may guide search; final holdout and canonical diagnostics may not |
| Candidate-family freeze | Generation stops for the claim under audit | Data window, benchmark, strategy, costs, metrics, and report code are fixed | Four-row score family; prospective targeted-search manifests | No new candidates, weights, operators, dates, or baselines after freeze |
| Claim-time evaluation | No scheduler updates or new proposals for the same claim | Evaluates frozen candidates on the locked holdout and exports returns/metrics | Score-family audit and targeted-search holdout reporting | SATS-QUANT maps evidence to supported claim, downgrade, or abstain |

Table 7: Functional stage map for R&D-AGENT(Q), Qlib, and SATS-QUANT, separating the adaptive search loop from the post-freeze audit layer.

## D Expert Score Matrices

A score matrix is a table indexed by decision date $t$ and asset $i$. Entry $\hat{s}^{(e)}_{t,i}$ is the scalar output of frozen expert $e$ before daily ranking; its absolute scale is model-specific, so the audit uses the within-day normalization in Eq. 2. Larger values mean the expert ranks the asset as more attractive for the next holding period.

**Prediction-table semantics.** For both experts, Qlib turns a trained workflow into the same kind of prediction table: one scalar score for each tradable asset on each decision date. The time-series input is a 20-day window of Alpha158 price/volume features. The future close-to-close return in Eq. 1 is the supervised training target and is used later for signal analysis; it is not an input available to the model when the score is exported. Thus $\hat{s}^{(e)}_{t,i}$ means "the frozen workflow's predicted attractiveness score for asset $i$ at date $t$," before daily normalization, portfolio construction, costs, or post-selection diagnostics.

**Qlib Transformer anchor.** Qlib constructs Alpha158 price/volume features and trains the Transformer predictor $f_A(\cdot;\theta_A)$ against the forward close-to-close label $y_{t,i} = P^{\text{close}}_{t+2,i}/P^{\text{close}}_{t+1,i} - 1$. The Transformer maps the 20-day feature window through a neural sequence model and outputs one scalar score, $\hat{s}^{(A)}_{t,i} = f_A(X^{(A)}_{t,i};\theta_A)$. This score is a deterministic neural-model prediction from the fixed Qlib workflow, not an LLM judgment.

**RD-AGENT-GENERATED EXPERT.** R&D-AGENT(Q) generated a Qlib-compatible PyTorch workflow, wrapped by Qlib's `GeneralPTNN`, using Alpha158 features and a 20-step feature sequence. The frozen generated model for this audited expert is a two-layer gated recurrent neural network with a linear scalar output, so it exports the same kind of score, $\hat{s}^{(B)}_{t,i} = f_B(X^{(B)}_{t,i};\theta_B)$. The audit treats this exported score matrix as the fixed LLM-assisted artifact under test, rather than as evidence about the full upstream R&D-AGENT(Q) search process.

## E Lock and Audit Checklist

Table 8 records the retrospective score-family lock evidence and release boundary.

| Field | Value |
|---|---|
| Pre-holdout registration evidence | No pre-holdout registration record or repository commit hash is available for the retrospective score-family audit; primary configuration SHA-256 prefix `adf407db5ebbec50` |
| Prediction artifact hash(es) | score report `ad8d087ba3a0c23c`; holdout returns `8bcafa20f18a3a92`; CV returns `5ae9b1fec4002e06` |
| Candidate family evidence | The four fixed mixtures (50/50, 80/20, 85/15, 90/10) are identified in available audit records; the pre-holdout chronology and wider upstream exploration cannot be proven |
| Primary benchmark and trading contract | `TopkDropoutStrategy`, topk 50, n_drop 5, close execution, benchmark `SH000300` |
| Holdout dates | Primary: 2020-08-03 to 2020-09-23; COVID transfer: 2020-02-03 to 2020-03-25; rolling windows in Table 9 |
| Transaction-cost settings | open cost 0.0005, close cost 0.0015, minimum cost 5, limit threshold 0.095 |
| Robustness-only diagnostics | Stacking baselines, MCS, rolling holdouts, and transfer rows are used for interpretation and were not used to choose the fixed 85/15 row |
| Artifact release boundary | Report-generation scripts, generated tables, frozen daily return series, and frozen daily score matrices where redistribution is permitted; otherwise hashes plus regeneration instructions |

Table 8: Lock and audit table for the primary artifact family.

# F Audit Statistic Settings

This appendix gives the finance-specific diagnostic definitions used in the paper. Let $E_{t,c}$ denote daily benchmark-relative excess return for candidate $c$ on day $t$, $\mathcal{C}$ the frozen candidate family, $T$ the number of evaluated days, and $\bar{x}$ a sample mean over $t$.

**Deflated Sharpe Ratio (DSR).** DSR is a multiple-testing-adjusted Sharpe/IR diagnostic. For observed Sharpe- or IR-style statistic $SR$, skewness $\hat{\gamma}_3$, kurtosis $\hat{\gamma}_4$, and effective number of trials $N_{\text{eff}}$, we use

$$DSR = \Phi\left(\frac{(SR - SR_0(N_{\text{eff}}))\sqrt{T-1}}{\sqrt{1-\hat{\gamma}_3 SR + (\hat{\gamma}_4 - 1)SR^2/4}}\right),$$

where $\Phi$ is the standard normal cumulative distribution function and $SR_0(N_{\text{eff}})$ is the expected best null Sharpe/IR threshold after $N_{\text{eff}}$ trials. Following the usual DSR approximation, $SR_0(N) = (1-\gamma)\Phi^{-1}(1-1/N)+\gamma\Phi^{-1}(1-1/(Ne))$, where $\gamma \approx 0.5772$ is the Euler–Mascheroni constant. Larger DSR means the observed statistic is less easily explained by repeated trials and non-normal returns.

**Combinatorially symmetric cross-validation Probability of Backtest Overfitting (CSCV-PBO).** CSCV-PBO estimates how often the in-sample winner becomes a poor out-of-sample choice. For each chronological split $j$, let $I_j$ and $O_j$ be complementary in-sample and out-of-sample slices, $m_{I_j}(c)$ the in-sample ranking metric, and $c_j^* = \arg\max_{c\in\mathcal{C}} m_{I_j}(c)$. With rank 1 defined as worst out of sample,

$$\widehat{PBO} = J^{-1}\sum_{j=1}^{J} \mathbf{1}[\text{rank}_{O_j}(c_j^*) \le |\mathcal{C}|/2].$$

Higher PBO means the selection rule more often chooses a candidate that falls into the worse half out of sample.

**White's Reality Check (RC) and Superior Predictive Ability (SPA).** RC and SPA are familywise tests against the Transformer anchor. For each candidate, define daily differential $d_{t,c} = E_{t,c} - E_{t,\text{Tr}}$ and mean $\bar{d}_c = T^{-1}\sum_t d_{t,c}$. The familywise null is

$$H_0 : \max_{c\in\mathcal{C}} \mathbb{E}[d_{t,c}] \le 0.$$

RC uses the family maximum statistic $\max_c \sqrt{T}\bar{d}_c$; SPA uses a studentized variant $\max_c \sqrt{T}\bar{d}_c/\hat{\omega}_c$,

where $\hat{\omega}_c$ estimates the time-series standard deviation of $d_{t,c}$. Both are evaluated by a moving-block bootstrap to preserve serial dependence.

**Model Confidence Set (MCS).** MCS starts from a candidate universe and loss $\ell_{t,c} = -E_{t,c}$. For candidates $c, c'$, define loss differential $g_{t,c,c'} = \ell_{t,c} - \ell_{t,c'}$. The equal-predictive-ability null for a current set $\mathcal{M}$ is

$$H_{0,\mathcal{M}} : \mathbb{E}[g_{t,c,c'}] = 0 \quad \text{for all } c, c' \in \mathcal{M}.$$

The procedure iteratively removes the worst candidate until this null is not rejected, yielding a set of statistically plausible winners at the chosen significance level.

For the frozen score family, CSCV-PBO and DSR are computed on the common pre-holdout diagnostic/CV window from 2017-08-09 to 2020-07-31 (725 trading days) over the four fixed mixtures. CSCV uses eight chronological slices, all 35 four-slice/four-slice train-test pairings, and IR as the ranking metric. DSR is reported for the $85/15$ row under both the raw family size $|\mathcal{C}| = 4$ and a correlation-adjusted effective trial count $N_{\text{eff}} = 1.010$, using the observed skewness $-1.602$ and kurtosis $15.900$; $N_{\text{eff}}$ is the participation-ratio estimate $(\operatorname{tr} R)^2/\operatorname{tr}(R^2)$ from the candidate return-correlation matrix $R$. RC/SPA are computed on the 38-day primary holdout using daily excess-return differentials against the Transformer anchor, a moving-block bootstrap with block length 7, and 5000 bootstrap draws. MCS uses the same holdout and bootstrap, loss $\ell_{t,c} = -E_{t,c}$, and the universe consisting of the strongest fixed mixtures (80/20, 85/15, 90/10) plus the strongest ordinary baseline, the validation grid.

The targeted-search experiment reports a five-seed matched-budget comparison with 30 proposals per seed and three promoted candidates per run. Proxy and CV metrics may guide promotion; final holdout and canonical post-selection diagnostics are never fed back into proposal generation. Promoted-set PBO uses the same CSCV machinery on the three promoted candidates; with $|\mathcal{C}| = 3$, only the out-of-sample worst-ranked selected candidate counts as worse-half overfitting, and stable sorting uses candidate order to break exact ties. The alignment metrics in the main text are diagnostic only: they show whether pre-holdout screening scores track final holdout IR among promoted candidates.

## G Score-Combination Baseline Details

Validation-selected convex grid search chooses

$$\alpha^\star = \arg\max_{\alpha \in \Lambda_{\text{grid}}} m_{\text{val}}(\alpha),$$

then evaluates Equation 3 with $\alpha^\star$ on the locked holdout. Here $\Lambda_{\text{grid}} = \{0, 0.05, 0.10, \ldots, 1.00\}$ is the Transformer-weight grid and $m_{\text{val}}(\alpha)$ is validation-window IR under the same Qlib trading contract. Rank averaging replaces scores by within-day ranks. Information-coefficient weighting computes each expert's validation-window daily cross-sectional rank IC (the correlation between scores and the next-period Qlib return label). The daily ICs are averaged over days, negative mean ICs are clipped to zero, and positive ICs are renormalized to convex weights. Ridge and non-negative least squares (NNLS) fit a global validation-window linear map from the two normalized expert scores $(z_{t,i}^{(A)}, z_{t,i}^{(B)})$ to the same Qlib label used by the experts. NNLS solves

$$\min_{\beta_A, \beta_B \geq 0} \sum_{(t,i) \in \mathcal{V}} \left( y_{t,i} - \beta_A z_{t,i}^{(A)} - \beta_B z_{t,i}^{(B)} \right)^2,$$

where $\mathcal{V}$ is the validation set, and then normalizes positive coefficients to form a convex score combination. Ridge uses the same target with signed coefficients, no intercept, and selects its penalty from $\{0.001, 0.01, 0.1, 1, 10\}$ by validation IR. In this setup NNLS collapses to the Transformer-only solution, so its holdout metrics equal the Transformer anchor. Ridge is allowed signed weights, and its validation-selected map fails on holdout, producing negative IR. These baselines test whether the fixed SATS-QUANT row is merely a learned ensemble weight.

## H Extended Audit Tables

Tables 9, 10, and 11 give the robustness diagnostics summarized in the main text.

| Outer window | Transformer | RD-Agent | Best fixed mix | Grid | Winner | 85/15 rank | Verdict |
|---|---|---|---|---|---|---|---|
| Pre-COVID Q4-A | -1.290 | -3.658 | 90/10: -1.130 | -2.406 | 90/10: -1.130 | 3 | Fixed mix least-bad |
| Pre-COVID Q4-B | -0.699 | 2.270 | 50/50: 0.614 | 0.432 | RD-Agent: 2.270 | 4 | RD-Agent regime |
| COVID crash | 1.407 | 2.089 | 50/50: 1.654 | 1.501 | RD-Agent: 2.089 | 6 | RD-Agent regime |
| Post-COVID rebound | 1.761 | -2.131 | 85/15: 2.376 | 0.871 | 85/15: 2.376 | 1 | Fixed mix wins |
| Mid-2020 recovery | -2.191 | -2.158 | 50/50: -1.811 | -1.863 | 50/50: -1.811 | 3 | Fixed mix least-bad |
| Primary holdout | 3.317 | 3.001 (n=1) | 85/15: 5.042 | 3.499 | 85/15: 5.042 | 1 | Fixed mix wins |

Table 9: Rolling outer holdouts. The fixed-mixture family repeatedly improves the Transformer anchor, but the best weight changes by regime.

| Candidate | In MCS at 10% level? | In MCS at 5% level? | Mean loss rank |
|---|---|---|---|
| Fixed 80/20 | Yes | Yes | 2 |
| Fixed 85/15 | Yes | Yes | 1 |
| Fixed 90/10 | Yes | Yes | 3 |
| Best baseline: validation grid | No | No | 4 |

Table 10: Moving-block MCS over fixed 80/20, 85/15, 90/10, and the validation grid. Other anchors and baselines are excluded from this restricted comparison.

| Test | Adjusted $p$ | Reject 5%? |
|---|---|---|
| White RC (family max) | 0.225 | No |
| SPA-style (family max) | 0.177 | No |

Table 11: Conservative family-max tests on the frozen score family. The null is anchor-relative: no fixed mixture has positive expected daily excess-return differential relative to the Transformer anchor.

# I Targeted-Search Seed-Level Results

Table 12 reports the seed-level quantities summarized in Table 4. The promoted operators were: RawIR (6 `CV_REFINE`, 3 `STRATEGY_PATCH`, 6 `COST_PATCH`) and SATS-QUANT (5 `CV_REFINE`, 6 `STRATEGY_PATCH`, 4 `COST_PATCH`).

| Seed | Controller | CV-reward-sel. IR | Proxy-sel. IR | Best IR | Mean top-3 IR | Collapse | Promoted PBO |
|---|---|---|---|---|---|---|---|
| 0 | RawIR | 0.634 | 0.634 | 3.482 | 1.583 | 0.400 | 0.343 |
| 0 | SATS-QUANT | 0.634 | 0.634 | 3.482 | 1.583 | 0.467 | 0.514 |
| 1 | RawIR | 1.900 | 2.003 | 2.015 | 1.973 | 0.600 | 0.086 |
| 1 | SATS-QUANT | 2.003 | 2.003 | 2.003 | 1.836 | 0.500 | 0.257 |
| 2 | RawIR | -0.655 | 0.655 | 0.655 | -0.088 | 0.500 | 0.000 |
| 2 | SATS-QUANT | -0.708 | -0.655 | 0.070 | -0.431 | 0.400 | 0.514 |
| 3 | RawIR | 2.067 | 3.443 | 3.443 | 2.526 | 0.400 | 0.714 |
| 3 | SATS-QUANT | 3.443 | 3.443 | 3.443 | 2.807 | 0.400 | 0.629 |
| 4 | RawIR | 2.906 | 2.991 | 2.991 | 2.934 | 0.533 | 0.000 |
| 4 | SATS-QUANT | 2.991 | 2.991 | 2.991 | 2.974 | 0.567 | 0.000 |

Table 12: Per-seed targeted-search experiment results. CV-reward-selected and proxy-selected columns use pre-holdout selection among the three promoted candidates; Best IR is a best-promoted holdout diagnostic. Promoted PBO is the per-run CSCV-PBO diagnostic over the promoted set.

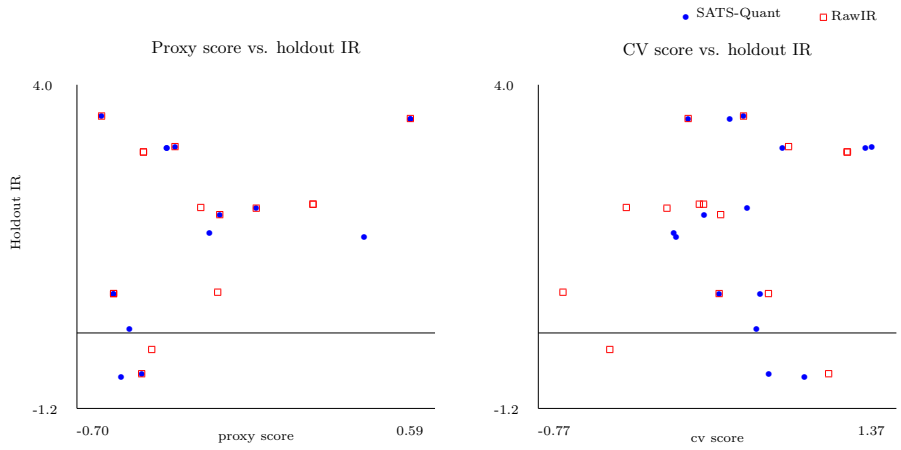


Figure 3: Targeted-search alignment diagnostic corresponding to Table 4. Each point is a promoted candidate; the axes compare pre-holdout screening scores with final holdout IR. The plot visualizes the rank-alignment signal only and is not evidence of endpoint dominance.

## J Agent Information Boundary and Action Space

This appendix documents the scientific information boundary used in the targeted-search experiment. Table 13 records what the agent could observe, what it could edit, and what was forbidden, so that hidden leakage is easier to rule out.

| Field | Constraint in the prospective targeted-search experiment |
|---|---|
| Observable information | Proxy metrics, chronological CV metrics, failure status, declared operator scope, and the files explicitly listed for the current operator |
| Editable objects | Only the files and fields declared by the active operator; the experiment used bounded `CV_REFINE`, `STRATEGY_PATCH`, and `COST_PATCH` operators |
| Forbidden edits | Holdout dates, benchmark, undeclared transaction-cost fields, data loading, metric code, reporting protocol, and any file outside the declared action space |
| Required output | Machine-checkable patch or bounded code diff that stays inside the declared scope |
| What remains hidden from search | Final holdout metrics, canonical CSCV-PBO, RC/SPA, MCS, and any post-selection reporting derived from the locked holdout |
| Scientific purpose | Ensure that search-time feedback is separated from final-report evidence and that the action space is auditable |

Table 13: Agent information boundary and action space for the targeted-search experiment. The aim is to document the leakage boundary, not to present project history.