Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Coding-Agent Numerical Reference Reconstruction and Abstention: A 32-Case KPI Audit

Meshaal Obaid Mouawad

mm4922@msstate.edu

Mississippi State University

Research Article

Keywords: Large language models, coding agents, test oracles, evidence grounding, numerical reference reconstruction, software measurement

Posted Date: November 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11824644/v1

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

Additional Declarations: No competing interests reported.

Coding-Agent Numerical Reference Reconstruction and Abstention: A 32-Case KPI Audit

Meshaal Obaid Mouawad
Department of Electrical and Computer Engineering
Mississippi State University
mm4922@msstate.edu

Abstract

Large language model (LLM) coding agents can assess software metrics only after establishing expected numerical values from applicable documentation. We investigate when two role-conditioned sessions of one coding model reconstructed or withheld such references from 32 version-pinned financial key performance indicator (KPI) packets across eight software projects. Each session used the same 15-field, source-cited assessment instrument and was instructed to exclude implementations and native outputs. Eleven cases produced concordant numerical point references, 20 were unresolved in both sessions, and one produced different resolution decisions. All 11 concordant points were independently recalculated from archived inputs and matched stored native results within the recorded absolute tolerances. A 32-case status index and the source-linked condition register record outcomes and the reported choices concerning input preparation, numerical scale, estimator divisor and time basis. Five archived conditional recalculations showed how alternative conventions changed the resulting scalar. The different-resolution RISKFOLIO_03 case exposed a choice between literal and peak-relative drawdown interpretations. Separate source-execution examples are retained in a scientific supplement. The contribution is a reproducible, evidence-linked corpus of numerical-reference submissions and explicit withholding decisions for studying the grounding of coding agents before implementation-conformance judgments.

Keywords: Large language models; coding agents; test oracles; evidence grounding; numerical reference reconstruction; software measurement

1 Introduction

Large language model (LLM) coding agents increasingly support software review, testing, and requirements interpretation. A numerical conformance judgment requires an expected value justified by the applicable specification before the observed program output is examined. We study that upstream reference-construction decision through 32 version-pinned financial key performance indicator (KPI) documentary packets from eight software projects. Two separately executed, differently role-conditioned sessions of the same coding model each returned a source-cited assessment using the same 15-field instrument. Eleven cases produced matching points, 20 were marked unresolved in both sessions, and one produced different resolution decisions. These retained outcomes provide a directly inspectable record of how the specified model configurations handled evidence available before the native-output comparison.

Consider an annualized-return metric whose input prices are fixed but whose available excerpt does not select a 365-day or 365.25-day year basis. Either convention yields a reproducible number; the documented rule must select the relevant one before that number can serve as a conformance reference. When a coding agent instead reports an unresolved choice and cites the relevant passage, it produces an observable withholding decision. The research question is what the actual two agent sessions returned under those conditions, rather than whether withholding itself was independently adjudicated as correct.

Prior software-engineering work establishes the test-oracle problem and methods for turning documentation into executable checks (Barr et al. 2015; Hossain et al. 2025; Kiecker et al. 2026). Contemporary research further investigates evidence-governed abstention in agentic software issue generation (Wang et al. 2026) and errors in LLM-generated requirement-conformance decisions (Jin and Chen 2026). These studies motivate an upstream, numerical question: which fixed documentary KPI packets led the two recorded coding-agent sessions to submit a scalar, withhold one, or disagree about resolution, and which documentary choices accompanied those decisions?

We answer this question using the recovered assignment constraints, 64 archived model-session forms for 32 cases, linked documentary excerpts, source-span verification summaries, the stored native result archive and retrospective reanalysis. The C28 case-level index uses exact original case identifiers and links both submitted assessments, uncertainty notes and citation spans to the archived evidence. We first classify the submitted documentary decisions; only then do we compare concordant numerical points

with native outputs. Eleven point references matched both independent raw-input recalculations and the retained native scalars within their pre-specified tolerances. Five conditional calculations document the numerical effect of selecting different conventions. An accompanying source-execution supplement preserves why reference decisions must remain separate from the counted entity, producer query and executed path.

The study contributes an instrumented, version-pinned reference-decision corpus for trustworthy software-engineering research: paired source-cited coding-agent outputs with explicit unresolved states, original assignment and provenance records, a case-level classification ledger, and an independently reproducible set of numerical checks. Its central finding is the recorded distribution of submitted and withheld numerical references under two role instructions to one model, with a source-linked 32-case decision index and five numerical-choice examples. The manuscript uses the archive to make evidence grounding and withholding inspectable at the reference-construction stage, before a pass/fail conformance verdict is produced.

2 Established research and the numerical-reference question

2.1 Explaining implemented numerical outputs

A numerical implementation can be inspected without knowing what output its specification requires. Müller and Engel (2022) demonstrate fine-grained how-, where- and why-provenance that connects SQL output cells to contributing input cells and expressions, including aggregation and bag semantics. Haroon et al. (2024) instrument SQL execution to expose intermediate subquery results, evaluating debugging overhead and fault-localization time. Gathani et al. (2020) synthesize query-debugging techniques and investigate user practices. These studies explain where a computed value came from and why a query produced it; their demonstrated outcomes concern query interpretation and debugging rather than the resolution of a numerical expectation from a fixed documentary packet. Research on statistical summarizability and SQL measurement expressions also characterizes when arithmetic operations combine compatible quantities (Lenz and Shoshani 1997; Thanisch et al. 2019). In Automated Software Engineering, Fantino et al. (2026) evaluate execution-derived data-change descriptions as inputs for automated code comments, reporting improvements for no-change cases without corresponding gains in semantically richer cases. That study uses observed data transfor-

mations to explain source behavior; it does not establish the documented numerical expectation for a separately chosen financial-metric convention. Together these contributions make the source-counting and execution-boundary examples in this manuscript intelligible without requiring those examples to carry its primary scientific claim.

2.2 Constructing expectations from documentation

The expected result is itself an object of research. Barr et al. (2015) identify the test-oracle problem and describe the difficulty of extracting precise expectations from partial or ambiguous textual specifications. Hossain et al. (2025) test documentation-based oracle generation using Java method documentation, contrasting the information available from different comment components and evaluating resulting oracles on executable tasks. Their study shows that documentation can supply useful oracle information, including when the implementation under test is excluded from the generation input. It therefore establishes documentary-only oracle construction as an existing capability. The remaining question for this audit concerns the recorded decision to submit or withhold a specific numerical reference when financial-metric excerpts leave a calculation convention unsettled.

Methods for detecting documentation–code disagreement evaluate different downstream outcomes. Lee et al. (2025) use tests and metamorphic queries to classify inconsistencies between program documentation and behavior. Kiecker et al. (2026) generate tests from documentation and cross-check them against an implementation generated from the same description, evaluating consistent and inconsistent source pairs and testing the approach on repositories. Tafese et al. (2026) formalize documentary business rules and implementation representations, then apply verification to cost-calculation consistency. These methods provide concrete mechanisms for testing whether a chosen description and executable behavior agree. The present cases record an earlier decision: whether the supplied material supports a determinate scalar in the first place, and which numerical alternatives were retained when it did not.

2.3 Evidence sufficiency and conformance judgments

Recent papers in Automated Software Engineering directly address the reliability of decisions supported by external records. Wang et al. (2026) govern the retrieval and use of official documentation, issue history and other sources when constructing software issue specifications; the system can obtain more

evidence or abstain and is evaluated on Firefox and Visual Studio Code data. Jin and Chen (2026) evaluate requirement-conformance judgments on executable code benchmarks under multiple prompting arrangements, reporting incorrect rejection of correct implementations and testing an execution-based verification filter. Both studies establish evidence selection and judgment reliability as measurable research problems. Their evaluation objects are, respectively, issue-specification decisions and conformance verdicts against available executable reference outcomes.

Tasnim et al. (2026) further demonstrate that reusable oracle research requires a documented test collection, executable evaluation conditions and explicit distinctions among oracle categories. Their benchmark measures generated oracles within software test suites. The present 32-case archive addresses a different measurement unit: a version-pinned financial-metric packet paired with two documentary numerical-reference submissions and their recorded source justifications. The evidence does not establish the general correctness or reliability of either assessment; it permits direct inspection of convergence, withholding and conflicting resolution under the archived conditions.

2.4 Empirical position of the 32-case audit

Research on query provenance and execution-aware documentation establishes how observed numerical outputs are produced, including query dependencies, intermediate results and data transformations (Müller and Engel 2022; Haroon et al. 2024; Fantino et al. 2026). Documentation-derived test-oracle research establishes techniques for constructing executable expectations from textual descriptions (Barr et al. 2015; Hossain et al. 2025). Documentation–behavior checking and formalized business-rule conformance then assess a selected reference against execution (Lee et al. 2025; Kiecker et al. 2026; Tafese et al. 2026). These are complementary parts of a verification pipeline rather than directly comparable accuracy results.

The closest agent-oriented evidence comes from Wang et al. (2026), who evaluated evidence acquisition, sufficiency checking and abstention in agent-governed software issue specification, and Jin and Chen (2026), who studied LLM conformance judgments against executable reference outcomes. The present audit measures an earlier numerical decision under a different task: whether the same coding model, conditioned by two roles, returned a source-cited expected scalar from an implementation-excluded KPI documentation packet. Its records make the specific choice of numerical convention available for inspection. The observed outcomes do not estimate model calibration

or multi-model reliability; they provide a reproducible empirical object on which those questions can subsequently be tested.

The selected 32-case corpus therefore connects documentation-derived oracle formation, evidence provenance and explicit reference withholding. Its contribution is measured as case-specific submissions and documented uncertainty, with independently recalculated numerical points and separate native-output checks. The finding concerns evidence-grounded numerical decisions, source-cited withholding and reproducible agent evaluation.

Table 1: Comparison with closely related oracle, evidence-sufficiency and program-analysis research. The evaluation targets differ; no cross-paper performance ranking is inferred.

Prior work	Evaluated object	Present study object
SQL provenance; DeSQL; Fantino et al. (2022–2026)	Dependencies, intermediate execution results and execution-informed comments	A source-cited expected numerical value established before comparison with observed output
Barr et al.; Doc2OracLL (2015, 2025)	Test-oracle problem; documentation-derived oracle generation	Whether paired documentary submissions converge or record unresolved numerical choices
MetaMon; CASCADE (2025–2026)	Documentation–behavior inconsistency tests on executable cases	The reference decision and its documentary passages, preceding a conformance verdict
SIRNA / Tafese et al. (2026)	Formalized business-rule equivalence to implemented functions	Alternative numerical conventions recorded from supplied source excerpts
Wang et al.; Jin–Chen (2026)	Evidence-grounded issue reporting and code conformance against executable references	Paired scalar-reference status and citation trace from fixed packets
TOGBench (2026)	Categories and runtime effectiveness of generated software-test oracles	Descriptive, source-linked numerical-reference reconstruction records

3 Materials and assessment method

3.1 Research question and observation units

The study asks: Which numerical point references did two separately executed, role-conditioned sessions of one model submit from identical version-pinned KPI documentary packets, and which semantic choices did their notes record as unresolved? The primary observation unit is one packet with its two reference submissions. The 32 packets belong to eight named software projects, four metrics per project. The supporting source-execution archives use different observational units (Table 2).

A point reference is a scalar submitted from documentary evidence and fixed inputs. For packet D and inputs x, let Theta(D,x) denote the set of admissible interpretations of the available specification. A point is determinate to tolerance epsilon when all relevant admissible interpretations produce values within epsilon of one another:

$$\max_{\theta_1,\theta_2\in\Theta(D,x)} |f_{\theta_1}(x) - f_{\theta_2}(x)| \leq \varepsilon. \tag{1}$$

The retained assessment forms do not enumerate the complete interpretation set. The empirical outcome is therefore the observed pair of documentary submissions; the conditional calculations illustrate selected unresolved conventions. Equation (1) defines the conceptual criterion, not a claim of formally adjudicated document completeness.

Table 2: Evaluation units. Denominators represent different retained objects.

Evidence set	Unit/count	Analytical role
Documentary-reference cohort	32 KPI packets; two same-model session returns per packet	Primary outcomes: submitting or withholding a numerical reference
Four Keys/PVWatts source archive	13 constructed-input observations	Supporting source-behavior and documented-reference contrasts
Matched incident records	Two dependent executions	Controlled test of incident identity and join multiplicity
dbt source-model comparisons	199 output identities	Supporting execution-path and nullability comparison

3.2 Version-pinned packets and retained case selection

The research archive includes 32 preselected financial-metric cases from eight projects, with source revision, metric, price or allocation inputs, parameters, documentary excerpts and a separately retained source-native scalar. The families cover growth, volatility, drawdown, downside measures, risk ratios and portfolio outputs. Table 3 gives the full fixed case distribution. The original selected cases and evidence links are preserved for this secondary analysis.

Table 3: Native-library metric cases. Each project supplies four fixed metrics at its pinned revision.

Project	`Revision prefix`	Included metrics
QuantStats	`d1a214fb906e`	CAGR, maximum drawdown, Sortino ratio, volatility
ffn	`ab34c72687b6`	CAGR, maximum drawdown, Sortino ratio, total return
Riskfolio-Lib	`344b899eb72e`	Historical conditional VaR, mean absolute deviation, relative maximum drawdown, historical VaR
PyPortfolioOpt	`a6638d2e06da`	Annualized volatility, expected return, Sharpe ratio, semideviation
skfolio	`ba8e39401040`	Maximum drawdown, annualized Sharpe ratio, standard deviation, mean
vectorbt	`9ac1b697166b`	Maximum drawdown, annualized volatility, total return, Sharpe ratio
empyrical-reloaded	`f937c0f3ae8e`	Maximum drawdown, Sharpe ratio, annual volatility, annual return
FinQuant	`55112a9b3d51`	Sortino ratio, value at risk, expected return, downside risk

Each packet contains the applicable available documentation or docstring excerpts, source attribution, raw input context and case-specific parameters. The two assigned archives retained the same 360 case files. Instructions excluded implementation files and native values from the material assigned for numerical-reference reconstruction and prohibited the sessions from consulting implementation results. The archived study record also establishes that personnel constructing the packets had inspected native outputs during preparation. This earlier exposure is part of the study provenance; later comparison of reference points with native outputs occurred after the returned decisions were classified. The original archive includes 32 native baselines and 64 constructed-input perturbation probes; the present primary analysis uses the two stored reference submissions and the corresponding original native baselines. The case-level source-span references and decision notes are retained in the companion documentary-condition register (Mouawad 2026b). The core replication deposit is denoted DOI1 (10.7910/DVN/W2RJ51); the companion documentary-condition deposit is denoted DOI2 (10.7910/DVN/4Q8BFP).

3.3 Recovered two-session assessment protocol

The recovered coordinator history and original assignment archives identify REF_A and REF_B as two separately executed Codex sessions, both using gpt-6.1-sol at low reasoning effort. They shared a coordinator parent context, workspace and underlying model, but received different requested professional-role instructions: procurement-KPI interpretation for A and corporate-finance

interpretation for B. The stated years of experience in those instructions were role descriptions, not verified qualifications. The coordinator history links the two session returns to the final archived CSVs and notes; their substantive fields match the retained submissions. The assessment outcomes consequently describe two configured model-session outputs, not agreement between distinct expert populations or independent model architectures.

Both sessions received the same case materials and 15-field assessment instrument. The fields cover case identity, assessment identity, release applicability, counted entity, observation identity, time window, eligibility, numerator, denominator, unit, allowed output, source spans, uncertainty and two timing fields. The recovered instructions required a documentary-derived numerical point, permissible value or reproducible formula where sufficiently specified, and an explicit UNRESOLVED outcome with competing interpretations when no unique point could be supported. They permitted computation from supplied prices and prohibited consulting the other submission, source implementations and native results. Both returned separate, source-cited files to their coordinator. Their notes report compliance with the specified source restrictions, but independent access-control monitoring was not retained. Complete per-case analytical durations cannot be inferred from validation or finish timestamps. The original assignment and return files are deposited under DOI1. The accompanying *Verification and Protocol Supplement* records the recoverable instrument schema, source-to-outcome mapping, integrity counts and the citation-endpoint correction procedure. The verification package reproduces both original assignment instructions verbatim and the original 15-column schema; the complete per-case evidence index retains each role-specific submission and citation.

3.4 Status coding and documentary-condition register

Each case was classified from the unmodified allowed_output fields. Concordant point requires both submissions to supply a scalar within the smaller stated absolute tolerance (10^{-12} or 10^{-10}). Jointly unresolved requires both to record an UNRESOLVED decision. Different resolution means one supplies a point and the other records UNRESOLVED. In this manuscript, "withholding" or "abstention" denotes that observed UNRESOLVED submission; it does not assume the choice was independently judged correct. Only the eleven concordant cases were subsequently compared with the retained native scalars, at full stored precision.

For every case we linked the recorded uncertainty fields and output statements to the cited documentary packet paths. The resulting 32-row register

(Mouawad 2026b) gives the paired status, original uncertainty statements, cited spans and a selected, non-exclusive classification of the reported calculation choices. The classifications are non-exclusive and summarize what was recorded: numerical scaling or sign, estimator divisor, return/input preparation, period basis, linked definition, producer defaults, portfolio configuration or a metric-name/formula conflict. A missing selector in a cited excerpt is a packet-level statement; it does not establish absence from every source document that could exist for the project. The categories describe selected topics recorded by the sessions, rather than an exhaustive count of all uncertainty mentions; the observed A/B status is the primary result.

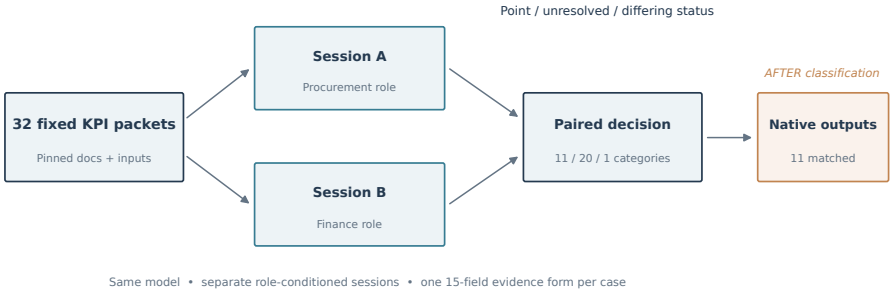


Figure 1: Recorded documentary-reference assessment: two role-conditioned Codex sessions received the fixed packets, returned source-cited decisions, and were compared with native outputs only after classifying the 11/20/1 statuses. Packet constructors had inspected native values earlier.

Five archived alternative calculations are retained as examples of consequences when a stated convention changes, using the original recorded values and inputs. No reference is resolved retrospectively by selecting the candidate that happens to match the native implementation. The original CSVs, source packets and secondary analysis scripts are deposited under DOI1 (Mouawad 2026c); the case-specific documentary-condition register is recorded under DOI2 (Mouawad 2026b).

3.5 Supporting source-execution evidence

The archived source-witness stream pins Four Keys at revision 6cc642e037a9 and PVWatts at revision 5d7c3e2e8f1 (DORA Team 2024; System Advisor Model and SAM Simulation Core 2024). Four Keys SQL was adapted for an archived local query-engine replay; PVWatts uses extracted source

assignments and unit conversions. The original thirteen archived observations comprise eight agreements and five differences, with two dependent incident-identity executions recorded separately. An earlier eight/four/one interpretation was reconciled to eight/five using the same thirteen records. The dbt archive separately compares 199 output identities for one demonstration dataset. These observations illustrate the need to identify the execution path and counting entity when interpreting a numerical reference.

4 Results

4.1 Primary 32-case documentary-reference outcomes

The 32-case reference audit produced eleven concordant numerical point reconstructions, twenty cases unresolved by both same-model sessions and one case with different resolution status (Figure 2). Both forms contain the same 32 case IDs without duplicate rows. Assessment A marked 21 references unresolved and assessment B marked twenty; the one-case difference accounts for their distinct totals. The eleven concordant points are distributed across five of the eight project families.

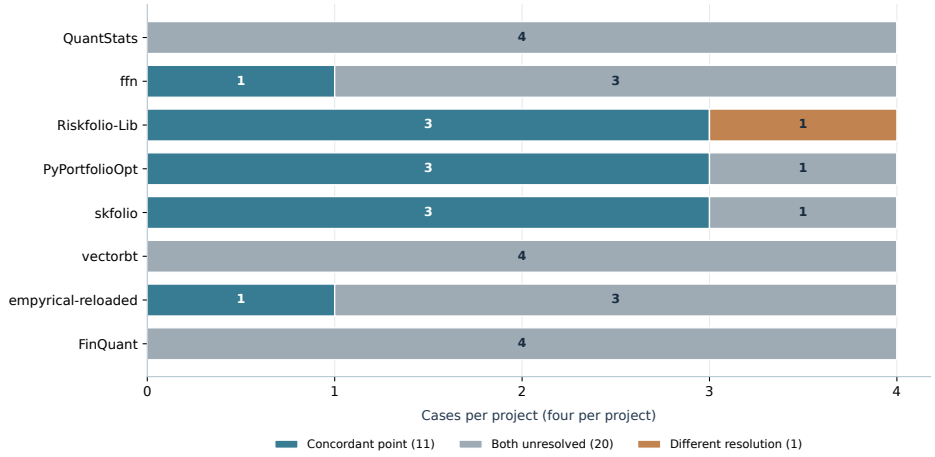


Figure 2: Recorded reference-decision outcomes across eight projects, four cases each. The bars represent eleven concordant points, twenty cases unresolved by both sessions and one different resolution.

The eleven concordant point submissions were independently recomputed from archived price inputs and match their stored native outputs within

the case-specific absolute tolerances. Table 4 summarizes the uncertainty topics reported in the nonconcordant submissions, and Table 5 displays the concordant points and their native comparisons.

Table 4: Selected, non-exclusive semantic-choice topics among 21 nonconcordant cases. Counts refer to recorded questions, not independently adjudicated causes.

Topic	Cases	Example question recorded in forms
Input or return preparation	9	First return; missing-price handling
Numerical scaling or sign	7	Drawdown sign; fraction versus percentage
Estimator divisor	7	Sample versus population dispersion
Period or year basis	5	365 versus 365.25 day count
Linked definition absent from extract	4	Referenced return/statistic definition
Portfolio configuration	4	Allocation and normalization defaults
Execution or producer defaults	4	Holding/order acquisition settings
Metric-name/formula conflict	1	RISKFOLIO_03

Table 5: Paired documentary and native numerical evidence for eleven concordant-point cases. Ref. A and Ref. B are separately submitted values; comparisons use full stored precision and absolute tolerances.

Case	Ref. A	Ref. B	Native	Unit	Tol.
ffn:04	0.0166402535658	0.0166402535658	0.0166402535658	fraction	10^{-12}
Risk:01	0.0257741954044	0.0257741954044	0.0257741954044	daily frac.	10^{-12}
Risk:02	0.0077759073069	0.0077759073069	0.0077759073069	daily frac.	10^{-12}
Risk:04	0.0169242089772	0.0169242089772	0.0169242089772	daily frac.	10^{-12}
PPO:01	0.206944579991	0.206944579991	0.206944579991	ann. frac.	10^{-10}
PPO:02	0.195326123948	0.195326123948	0.195326123948	ann. frac.	10^{-10}
PPO:04	0.137768587642	0.137768587642	0.137768587642	ann. frac.	10^{-10}
skf:02	1.03525375136	1.03525375136	1.03525375136	ratio	10^{-10}
skf:03	0.280187506865	0.280187506865	0.280187506865	ann. frac.	10^{-10}
skf:04	0.290065167568	0.290065167568	0.290065167568	ann. frac.	10^{-10}
emp:04	0.259785325644	0.259785325644	0.259785325644	ann. frac.	10^{-10}

For twenty cases, both submissions documented unresolved convention choices. We linked each reported choice to its original form and cited packet extract without assigning a causal diagnosis of the source material. Table 4 shows the non-exclusive categories; the full 32-case register identifies each status, recorded uncertainty, citation path and output. All five conditional alternatives in Table 6 were independently recalculated from the archived inputs. They illustrate how a different rule changes the scalar on the same prepared input; these are calculations under stated alternatives, not additional observed implementation differences.

Table 6: Five archived conditional calculations. Values are consequences of stated alternatives on fixed inputs, rather than independently authorized expected outcomes.

Case	Convention	Alternative 1	Alternative 2
QuantStats 03	All observations / loss-only downside divisor	0.1482551128	0.1018159504
ffn 01	ACT/365 / ACT/365.25 for CAGR	0.0041428837	0.0041457271
vectorbt 02	Omit / retain initial zero-return row	0.2394820565	0.2393236200
empyrical 02	Sample / population standard deviation	1.1389763970	1.1397334436
Riskfolio 03	Literal wealth difference / peak-relative magnitude	0.3078446910	0.2434210526

The only different-resolution case is RISKFOLIO_03. Assessment A retained two interpretations: the literal wealth-difference expression gives approximately 0.307844690966720, while peak-relative normalization gives approximately 0.243421052631579. Assessment B submitted the literal value, approximately 0.3078446909667185, as its point reference. The native output is approximately 0.2434210526315783. The contrast identifies the unresolved selection of documentary interpretation: native agreement with one alternative does not establish that it is the governing reference.

Every case links the submitted decision to cited documentary excerpts in the companion 32-case condition register (Mouawad 2026b). The source-span audit located all 128 case–path references (127 distinct files). Of 212 submitted source-span occurrences, 180 were fully within the relevant file. The other 32 were A's case_context.json ranges, each extending one to three lines beyond the file's end; every documentary-text range was in bounds. The supplementary 212-entry ledger records both original and corrected endpoints without changing either original form. Eighteen additional file-only source citations across 16 cases were checked separately and are retained in a second ledger. The uncertainty categories describe questions recorded in the submissions. Only the different-resolution case permits a direct comparison of two choices made from the same printed formula; causal explanations for other nonconcordant submissions remain unassigned.

Table 7: Previously verified reproduction and source-linkage checks from the archived study record.

Check	Retained verification result
32-case outcomes	32 unique IDs; 11 concordant; 20 unresolved by both; 1 differing
Concordant scalar recomputation	11/11 recalculated from archived prices and matched native values under original absolute tolerance
Conditional alternatives	5/5 pairs recalculated to ten decimal places
Source paths	128 case–path references to 127 distinct existing files
Cited source spans	212 occurrences: 180 within file bounds; 32 context-file endpoints corrected separately; 18 additional path-only citations checked
Archived reanalysis	Stored primary and documentary-condition outputs reproduced byte-for-byte in prior verification

4.2 Cross-case patterns in the recorded unresolved choices

The original documentary-condition register assigns one or more selected, non-exclusive semantic-choice codes to each of the twenty-one cases without a concordant point. Six cases have one recorded code, ten have two and five have three; thus fifteen of the twenty-one carry multiple recorded topics. These counts describe the original assessment notes, not a judgement of reference correctness. Table 8 keeps each project's jointly unresolved and different-resolution cases separate and counts cases with at least two recorded codes. The complete code-by-case membership, exact original identifiers, original A/B comments and source citations are provided in the C28 index and verification evidence compendium.

Table 8: Distribution of the twenty-one nonconcordant cases and co-recorded documentary-choice topics. A case with at least two codes is counted once in the last column.

Project	Both unresolved	Different status	$\geq$2 topics
QuantStats	4	0	4
ffn	3	0	2
Riskfolio-Lib	0	1	1
PyPortfolioOpt	1	0	0
skfolio	1	0	0
vectorbt	4	0	3
empyrical-reloaded	3	0	1
FinQuant	4	0	4
Total	20	1	15

4.3 Case-level decisions, documentary choices and trace checks

The companion *C28 32-Case Decision Index* enumerates all 32 original case identifiers, with their previously used readable display aliases, project, metric and paired status. For each case it preserves both original `allowed_output` submissions, the original role-specific `uncertainty` notes, the `source_spans` strings and the recorded documentary-condition codes. The `original_case_id` column is the authoritative join key to the unmodified REF_A and REF_B forms and the source-packet directory. For example, display label `Riskfolio-Lib_03` maps to original `RISKFOLIO_03`; display `PyPortfolioOpt_01` maps to `B_pyportfolioopt_01`; and `FinQuant_01` maps to `finquant_1`. The supplementary case-evidence compendium reproduces the returned descriptions and cited source paths, not retrospective documentary truth labels.

Five cases expose a reported convention choice whose numerical consequence is preserved (Table 9). Four were jointly unresolved; RISKFOLIO_03 had different resolution. The first four examples concern the year basis, downside divisor, first-return treatment and estimator divisor; the fifth contrasts a literal drawdown formula against peak-relative normalization. The same fixed inputs yield different candidate numbers under the named conventions (Table 6). The index preserves the exact source spans given by each role and links them to DOI1 source packets and the DOI2 condition register, retaining original citation text unchanged.

Table 9: Five source-linked conditional examples identified by the archived assessment notes. The status is the observed paired decision, not an adjudicated quality label.

Case	Paired status	Recorded documentary choice
QuantStats_03	Both unresolved	Downside divisor: all observations versus loss-only
ffn_01	Both unresolved	CAGR year basis: ACT/365 versus ACT/365.25
vectorbt_02	Both unresolved	Initial zero-return row: omit versus retain
empyrical_02	Both unresolved	Dispersion divisor: sample versus population
RISKFOLIO_03	Different resolution	Literal wealth difference versus peak-relative magnitude

The recorded citation audit provides a second case-level check: 128 case–file links refer to 127 distinct existing packet files, and 212 submitted source-span occurrences were checked. All documentary-text ranges were in bounds. The 32 out-of-bounds spans were references from assessment A to `case_context.json`, each extending one to three lines beyond the file end;

those require the preserved endpoint-correction rule rather than a change to an agent's submitted decision. The C29 verification package contains the exact original 15-field schema and instructions, the 212-entry original/corrected span ledger, and 18 separate file-level citations; the original 32 submissions remain unchanged. Counts of linked files and citation bounds characterize evidence inspectability; they do not adjudicate whether either submitted numerical interpretation was required by the full project specification.

The archived cases therefore expose three distinct observable outputs that prospective coding-agent studies can evaluate against the same frozen packets: (i) submission of a scalar or explicit `UNRESOLVED` outcome; (ii) the specificity and file validity of the documentary citations; and (iii) the reproducible numerical consequences of a selected convention. A future accuracy or calibration score requires an independent documentary reference standard that was not created by the two original sessions. The contribution here is the case-level decision evidence and its traceable evaluation inputs.

4.4 Supporting source-execution comparisons

These archived comparisons examine how an executable KPI output can depend on its counting identity and source path. They are separate from the 32 documentary-reference decisions. Table 10 reports their preserved results. The original source contrasts, controlled incident intervention, DevLake denominator query comparison and three vector figures are retained in the separate *Supporting Source-Execution Evidence* supplement.

Table 10: Selected supporting source-execution contrasts. Counts and denominators belong to their own source experiments, not to the 32 coding-agent cases.

Source case	Recorded comparison	Mechanism or boundary
Four Keys	25% affected deployments vs 50% joined-row result	One affected deployment; two joined contributions among four deployments
PVWatts	20% DC vs 25% AC capacity factor; 0% controls	Same generation; different nameplate basis; documented unit conversion
dbt Jaffle Shop	161 matching and 38 different values of 199	NULL from full left-join model versus zero from a reduced expression
Apache DevLake	0.6667 documentation SQL vs 1.0000 panel-99 SQL	Joined-row versus distinct-PR denominators under aligned filters
Four Keys restoration	3 h direct versus 4 h with upstream producer	Source of start timestamp changes

5 Discussion

The primary finding concerns the construction of a numerical reference before source-code conformance is judged. Eleven of the 32 version-pinned packets produced matching scalar submissions from the two recorded Codex sessions. Twenty produced UNRESOLVED decisions in both sessions, and one produced different resolution decisions. Both sessions used the same model with different requested professional roles, and their source-cited forms make the actual output of those configurations inspectable. The 11 concordant scalars were recomputed from archived prices and matched their retained native values within recorded tolerances. The result is a reproducible distribution of reference-decision outcomes, not an inferred measure of general model accuracy.

The content of the withheld decisions matters. Across the 21 nonconcordant cases, the archived forms mention input or return preparation in nine cases, scaling or sign in seven, estimator divisor in seven and period or year basis in five; categories overlap and additional topics appear in the original register. Five original conditional recalculations show that choosing one documented alternative rather than another changes the numeric scalar on fixed input data. Their evidential role is to demonstrate numerical consequence, while the paired assessments record whether an agent submitted a point under the available excerpts.

RISKFOLIO_03 exposes the distinction within one packet. One assessment retained a literal wealth-difference result and a peak-relative interpretation; the second submitted the literal quantity as a point. The archived native output matches the peak-relative interpretation. Agreement with a native implementation does not itself select which available documentary convention is normative. The source-linked ledger allows subsequent researchers to examine that interpretation choice without silently replacing the documentary record with the observed implementation.

For trustworthy LLM-assisted software engineering, this identifies a distinct stage requiring its own evaluation. A model can cite a source passage, produce a number or explicitly withhold one; these outcomes are directly measurable. Whether a specific withholding decision was necessary requires an adjudicated reference standard and independently monitored access conditions. Keeping the reference decision separate from the later implementation comparison prevents a matching executable result from being treated as the source of the expectation. The recovered protocol records the model, role instructions, 15-field form, permitted materials and coordinator linkage for a reproducible audit of the actual two-session procedure.

The closest prior studies give the finding its context. Wang et al. (2026) investigate evidence-aware selection and abstention in issue-generation workflows; Jin and Chen (2026) identify erroneous LLM conformance judgments against executable references. Hossain et al. (2025) examine documentation-only oracle generation. The present audit adds a concrete numerical-reference decision corpus with paired role-conditioned outputs, case-specific source spans and unreconciled calculation choices. It supplies material for evaluating future agent strategies for obtaining, selecting and withholding expected numerical outputs in software measurement workflows.

The supporting source studies address the later execution stage. Their concise comparison in Table 10 demonstrates how counting units and producer paths alter source-derived outputs. The full original analyses, including the Four Keys intervention and DevLake documentation-versus-panel SQL contrast, are in the companion source-execution supplement; no supporting example is counted in the 32-case coding-agent denominator.

The 32-case original-identifier index, verbatim role-specific submissions, citation ledgers and five conditional calculations make the empirical result inspectable beyond the aggregate 11/20/1 totals. A future study can apply its own model or prompt to the same fixed documentary packets and report three quantities separately: whether it returned a point or an explicit unresolved decision, whether cited sources and spans are valid, and whether any submitted scalar follows its stated convention on the supplied data. With an independently adjudicated documentary oracle, that future study could then evaluate the appropriateness of those decisions. This separation is the operational contribution of the released record to trustworthy coding-agent evaluation.

6 Conclusion

This 32-case audit measured numerical-reference reconstruction by two role-conditioned sessions of one coding model from version-pinned KPI documentation. Eleven packets yielded concordant point references, twenty were unresolved in both submissions, and one produced different resolution decisions. All eleven matching point submissions were independently recalculated and agreed with the archived native scalars within tolerance. Source-linked case notes and five conditional computations document the numerical conventions implicated in nonconcordant decisions.

The study establishes a reusable record of coding-agent reference submissions and withholding decisions before implementation conformance checking.

Its case-level decision index, documentary-choice examples, recorded protocol and source-link audits support direct inspection of the grounding step. The companion execution supplement preserves the different question of how source identity and executed paths affect observed results.

7 Limitations and reproducibility

The primary corpus is a purposive selection of 32 financial KPIs from eight projects. A complete original candidate-screening and exclusion record was not retained. Cases share input histories and related estimators, so the observed 11/20/1 distribution is descriptive of the archived packets rather than a population-level estimate. The documentary excerpts can omit definitions elsewhere in a project.

Both assessments were executed by separate sessions of the same gpt-6.1-sol coding model at low reasoning effort with different role-conditioned instructions and a common coordinator context. The requested professional identities were prompt roles, not verified human qualifications. Access restrictions were specified in the original assignments and reported in the returned notes, without independent technical monitoring of isolation. Packet-construction personnel had inspected native outputs before distributing the documentary packets. Therefore the recovered protocol supports analysis of these configured model-session decisions; it does not establish independent-expert agreement, multi-model generalization or the correctness of every withholding decision. Full per-case analysis durations were not recorded.

The condition codes summarize selected, non-exclusive uncertainty topics in the original forms, not exhaustively adjudicated causes. The source-span audit found 212 line-range citations; 180 were within their files, while 32 case-context citations extended one to three lines beyond the file ends. Separately, 18 file-only citations across 16 cases pointed to existing files. A separate ledger retains valid endpoints without changing the archived forms. The five conditional calculations establish numerical effects of selected conventions rather than additional native-execution observations.

Supporting Four Keys SQL results use an adapted local replay, PVWatts observations inspect selected assignments, and dbt comparisons share one nullability mechanism in one demonstration dataset. The DevLake contrast comes from pinned website and panel-99 SQL and a constructed-input calculation, not an observed deployed Grafana execution.

Statements and Declarations

Author contributions

Meshaal Obaid Mouawad directed the study and manuscript preparation and takes responsibility for the reported analysis.

Funding

This research received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Competing interests

The author declares that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The primary published dataset is Mouawad (2026c), "Replication Data for Counting Units, Execution Boundaries, and Documentary Sufficiency in KPI Conformance (Core Replication Materials)", Harvard Dataverse, version 1.0, DOI 10.7910/DVN/W2RJ51. It contains 13 deposited files including the original completed 32-case assessment forms, fixed source packets, archived execution observations and a retrospective reanalysis. The separately deposited supplementary dataset is Mouawad (2026b), "Replication Data for Counting Units, Execution Boundaries, and Documentary Sufficiency in KPI Conformance", Harvard Dataverse, DOI 10.7910/DVN/4Q8BFP. It contains four files: the 32-case documentary-condition register, codebook, derived summary and reconstruction program. The second dataset was in draft/in-review status as of 20 November 2026. The C29 verification package includes an original-ID 32-case decision index, both original agents' outputs and uncertainty descriptions, exact source paths, the fifteen original column names, original assignment instructions, and separate line-range and file-level citation ledgers.

DOI1, core replication: https://doi.org/10.7910/DVN/W2RJ51

DOI2, documentary conditions: https://doi.org/10.7910/DVN/4Q8BFP

Research repository: https://github.com/Meshaal-Mouawad/LEAP-Paper5-KPI-Conformance

Preparation statement

During preparation of this work, the author used Elsevier LeapSpace and other tools to support literature discovery, manuscript organization, language refinement, and consistency checking. The author takes responsibility for the content of the article.

References

Apache DevLake application contributors (2026). Engineering throughput and cycle time: shipped Grafana dashboard, panel 99 (PR Review Depth). Pinned archived revision. https://github.com/apache/devlake/blob/652d7da9323fd4447d3c2d3705f482ff5a9f54d8/grafana/dashboards/mysql/engineering-throughput-and-cycle-time.json.

Apache DevLake website contributors (2026). PR Review Depth: pinned documentation and SQL. Pinned archived revision. https://github.com/apache/devlake-website/blob/10f9538f82d61a85be02760ca2864320b97ed0d8/docs/Metrics/PRReviewDepth.md.

Barr, E.T., Harman, M., McMinn, P., et al. (2015). The oracle problem in software testing: A survey. IEEE Transactions on Software Engineering 41(5):507–525. https://doi.org/10.1109/TSE.2014.2372785.

Cube contributors (2026). Cube: pinned source snapshot. https://github.com/cube-js/cube/tree/e2b4c6726d1a0d712cf0ad7cc8b164d469d19e28.

dbt Jaffle Shop DuckDB contributors (2026). dbt Jaffle Shop DuckDB: pinned source snapshot. https://github.com/dbt-labs/jaffle_shop_duckdb/tree/20cc904370818ceab08cf19f8fd196e9e81def88.

DORA Team (2024). Four Keys: pinned source and change failure rate definition; commit 6cc642e037a95a280ed8d3b1624dd003b647d797. https://github.com/dora-team/fourkeys/tree/6cc642e037a95a280ed8d3b1624dd003b647d797.

dora-team/fourkeys (2022a). Change failure query should have deploy_id as denominator, issue 351. https://github.com/dora-team/fourkeys/issues/351.

dora-team/fourkeys (2022b). Change to deploy as failure-rate denominator, pull request 352. https://github.com/dora-team/fourkeys/pull/352.

Fantino, G., Vetro, A., Torchiano, M., et al. (2026). Beyond syntax: enhancing automated documentation with data differences. Automated Software Engineering 33(77). https://doi.org/10.1007/s10515-026-00623-y.

Gathani, S., Lim, P., Battle, L. (2020). Debugging database queries: A survey of

tools, techniques and users. CHI 2020. https://doi.org/10.1145/3313831.3376485.

Haroon, S., Brown, C., Gulzar, M.A. (2024). DeSQL: Interactive debugging of SQL in data-intensive scalable computing. Proceedings of the ACM on Software Engineering 1(FSE). https://doi.org/10.1145/3643761.

Hossain, S.B., Taylor, R., Dwyer, M.B. (2025). Doc2OracLL: Investigating the impact of documentation on LLM-based test oracle generation. Proceedings of the ACM on Software Engineering 2(FSE):1870–1891. https://doi.org/10.1145/3729354.

Jin, H., Chen, H. (2026). Are LLMs reliable code reviewers? Systematic overcorrection in requirement conformance judgement. Automated Software Engineering 33(90). https://doi.org/10.1007/s10515-026-00638-5.

Kiecker, T., Sparka, J.A., Reuter, M., et al. (2026). CASCADE: Detecting inconsistencies between code and documentation with automatic test generation. Proceedings of the ACM on Software Engineering 3(FSE):3816–3838. https://doi.org/10.1145/3808175.

Lee, H., An, G., Yoo, S. (2025). MetaMon: Finding inconsistencies between program documentation and behavior using metamorphic LLM queries. LLM4Code 2025:120–127. https://doi.org/10.1109/LLM4Code66737.2025.00020.

Lenz, H.J., Shoshani, A. (1997). Summarizability in OLAP and statistical databases. SSDBM 1997:132–143. https://doi.org/10.1109/SSDM.1997.621175.

Mouawad, M. (2026a). LEAP: Automating KPI documentation in enterprise codebases through literate programming and AI. Zenodo v1. https://doi.org/10.5281/zenodo.20764919.

Mouawad, M. (2026b). Replication Data for Counting Units, Execution Boundaries, and Documentary Sufficiency in KPI Conformance. Companion documentary-condition register; Harvard Dataverse. https://doi.org/10.7910/DVN/4Q8BFP.

Mouawad, M. (2026c). Replication Data for Counting Units, Execution Boundaries, and Documentary Sufficiency in KPI Conformance (Core Replication Materials). Harvard Dataverse v1. https://doi.org/10.7910/DVN/W2RJ51.

Mouawad, M. (2026d). A source-linked review method for KPI definition–implementation disagreements. Research Square preprint v1. https://doi.org/10.21203/rs.3.rs-11525474/v1.

Muller, T., Engel, P. (2022). How, where, and why data provenance improves query debugging: fine-grained provenance analysis for SQL. ICDE. https://db.cs.uni-tuebingen.de/publications/2022/how-where-and-why-data-provenance-improves-query-debugging--a-visual-demonstration-of-fine-grained-provenance-analysis-for-sql/.

Rill examples contributors (2026). Rill examples: pinned source snapshot. https://github.com/rilldata/rill-examples/tree/5f0bf28b9f03f191aabcabe6836f1dab93d9754a.

System Advisor Model and SAM Simulation Core (2024). Capacity factor documentation and PVWatts V8 implementation, release 2024.12.12.r0.ssc.298. https://github.com/NatLabRockies/ssc/blob/5d7c3e2e8f113f1aaed0936ba2fc924295a496ba/ssc/cmod_pvwattsv8.cpp.

Tafese, J., Gurfinkel, A., Bayless, S., et al. (2026). Verifiable checks for business rule consistency. FMCAD 2026:338–348. https://doi.org/10.34727/2026/isbn.978-3-85448-093-8_39.

Tasnim, T., Dwyer, M.B., Hossain, S.B. (2026). TOGBench: A developer-written multi-variant dataset and benchmark suite for test oracle generation. AIware 2026:358–366. https://doi.org/10.1145/3805760.3814927.

Thanisch, P., Niemi, T., Nummenmaa, J., et al. (2019). Detecting measurement issues in SQL arithmetic expressions and aggregations. Data & Knowledge Engineering 122:116–129. https://doi.org/10.1016/j.datak.2019.06.001.

Wang, Z., Li, J., Guo, X., et al. (2026). From noisy feedback to evidence-aware issue specifications: an agent-governed retrieval-augmented generation approach. Automated Software Engineering 33(103). https://doi.org/10.1007/s10515-026-00640-x.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- LEAPPaper5C28VerificationProtocol.pdf
- LEAPPaper5C28SourceExecutionSupplement.pdf