

FinBalance: A Multi-Document Accounting Reconciliation Benchmark

Sasank Tumpati^{1,*}, Devansh Agarwal^{1,*}, Ayush Kedia¹, Arjun Neekhara¹
Murari Mandal², Krishna Garg³, Yash Sinha¹, Suman Gupta⁴, Dhruv Kumar¹

¹Birla Institute of Technology and Science, Pilani

²Kalinga Institute of Industrial Technology, Bhubaneswar

³University of Oxford

⁴Birla Institute of Technology and Science, Pilani, K K Birla Goa Campus

*Equal contribution.

Correspondence: f20230570@pilani.bits-pilani.ac.in

Abstract

Existing financial-NLP benchmarks mostly evaluate prepared artifacts such as filings, tables, or extracted values. Real accounting begins earlier: source documents must be reconciled into cited journal entries, aggregated into a balance sheet, and checked for contradictions. We introduce FINBALANCE, a multi-document accounting reconciliation benchmark built from source-document bundles across eight industries, three period types, and five difficulty levels. Human-authored business scenarios, accounting policies, tax/FX treatments, document schemas, distractors, and inconsistency templates are composed by a deterministic generator whose ledger produces journal entries, balance sheets, and 23 inconsistency-code labels. On a 710-record evaluation split, six contemporary large language models (LLMs) reach at most 46% exact final-balance-sheet accuracy. For four models, balance sheets reconstructed from their journal entries score 26–41 percentage points higher than their own reported balance sheets. Models often recover numerically plausible entries but fail to bind them to supporting documents and aggregate them consistently. Citation-pressure prompting barely changes document-linking errors, while ledger-feedback ablations substantially improve reported balance sheets and expose inconsistency-detection trade-offs. Expert finance reviewers validate the benchmark design and labels.

1 Introduction

Large language models (LLMs) are increasingly applied to financial tasks, but the dominant evaluation paradigm targets *comprehension* of already-prepared artifacts. FinQA (Chen et al., 2021), ConFinQA (Chen et al., 2022), and TAT-QA (Zhu et al., 2021) pose numerical questions over excerpted tables and text, while DocFinQA (Reddy et al., 2024) pairs FinQA-style questions with full SEC reports. FinanceBench (Islam et al., 2023) and

FinAuditing (Wang et al., 2026) test analysis or auditing over prepared filings, while FinanceMATH (Zhao et al., 2024) and FinBen (Xie et al., 2024) evaluate broader prepared financial problems and tasks. Across these settings, the model reasons over supplied inputs rather than constructing a ledger and balance sheet from source-document bundles.

Real accounting workflows begin one step earlier. A bundle of source documents, including invoices, bank statements, payment notices, contracts, schedules, and exemption certificates, must be reconciled into a transaction-level ledger of debit and credit journal entries and a balance sheet. This process requires extracting each transaction as a balanced entry with citations to its supporting documents, aggregating the entries into a statement that respects the accounting equation, and detecting and classifying contradictions instead of silently resolving them. We introduce FINBALANCE, a multi-document accounting reconciliation benchmark built from source-document bundles with templated OCR text across eight industries, three period types, and five difficulty levels. Its concept-coverage flags include revenue recognition under ASC 606, leases, deferred tax, asset disposals, multi-jurisdictional sales tax, foreign-exchange scenarios, and tax exemptions. Appendix B gives the complete list.

FINBALANCE supports both evaluation and diagnostics. Its construction requires explicit models of business processes, evidence documents, accounting policies, document dependencies, distractors, and contradiction types. Performance on fully specified templated bundles is a controlled diagnostic for real accounting records, which add layout noise, OCR errors, missing evidence, policy choices, and institutional conventions. We evaluate six contemporary-model baselines, citation ablations, context-vs.-logic stress tests, prompt-described and provider-native tool protocols, best-of-3 selection, and forced ledger feedback. In this

Benchmark	Evidence given to model	Target output	QA	Long/ multi ctx.	JE	BS	Doc/span grounding	Incons.	Det. replay
FinQA (Chen et al., 2021)	Extracted financial-report table and nearby text	Numeric answer plus executable reasoning program	✓	△	–	–	–	–	–
ConvFinQA (Chen et al., 2022)	Financial report context with conversational turns	Turn-level numeric answers and programs	✓	△	–	–	–	–	–
TAT-QA (Zhu et al., 2021)	One hybrid table-text context from financial reports	Answer with arithmetic/operator supervision	✓	△	–	–	△	–	–
DocFinQA (Reddy et al., 2024)	Full SEC filing attached to FinQA-style questions	Long-context numeric answer/program	✓	✓	–	–	△	–	–
MultiHiertt (Zhao et al., 2022)	Multiple hierarchical tables plus report text	Numeric answer with supporting facts	✓	✓	–	–	✓	–	–
FinanceBench (Islam et al., 2023)	Public filings and earnings-report evidence	Open-book financial QA answer	✓	✓	–	–	✓	–	–
FinanceMATH (Zhao et al., 2024)	Finance-domain word problems with text/table inputs	Knowledge-intensive math solution	△	△	–	–	–	–	–
FinBen (Xie et al., 2024)	Suite of finance datasets across IE, QA, RAG, forecasting	Task-specific outputs across 24 tasks	△	△	–	–	△	–	–
FinDABench (Liu et al., 2025)	Financial tables, charts, and documents	Data-analysis answers over prepared artifacts	△	△	–	–	△	–	–
FinTagging (Wang et al., 2025)	Financial-report text and tables with numerical facts	Extracted facts and aligned US-GAAP taxonomy concepts	–	△	–	–	✓	–	–
FinRule-Bench (Malarkkan et al., 2026)	Real financial-statement tables plus explicit principles	Rule verification, identification, and joint diagnosis	△	△	–	–	△	△	–
FinMaster (Jiang et al., 2026)	Synthetic transactions, journal entries, and statements	Task-specific metrics, diagnoses, or financial statements	△	△	△	△	△	△	–
FinAuditing (Wang et al., 2026)	Real XBRL filings with taxonomy-defined structure	Semantic, relational, and numerical auditing judgments	–	✓	–	–	✓	✓	–
FINBALANCE (this work)	Unlabeled source-document bundle with OCR text and hidden document roles	Cited journal entries, final balance sheet, inconsistency code	–	✓	✓	✓	✓	✓	✓

Table 1: Related-benchmark comparison. **QA** marks question-answering or problem-solving benchmarks. **JE** and **BS** mark construction of journal entries and a final balance sheet. **Incons.** marks explicit contradiction or consistency detection. **Det. replay** marks labels and evaluation produced by replaying a deterministic ledger. ✓ = central objective, △ = adjacent or subset capability, – = outside the benchmark objective.

panel, the highest exact final-balance-sheet score is 46%, and strict document-grounded end-to-end accuracy remains low.

FINBALANCE is synthetic, with its accounting content specified by the authors. The specification defines business scenarios across industries, document schemas, a chart of accounts, accounting policies, tax and FX treatments, support-document dependencies, distractor templates, contradiction templates, and valid ranges for amounts, dates,

counterparties, document roles, and scenario combinations. The generator samples valid combinations, fills document fields, selects compatible evidence and distractors, renders source documents with OCR text, and uses the fixed double-entry ledger to compute the expected journal entries and balance sheet. This process produces controlled records that would be expensive to write individually. Every expected entry, balance sheet, opening trial balance, and inconsistency code remains

computable, auditable, and reproducible from the specification.

We report two complementary balance-sheet metrics. BS_{exact} measures whether the model’s reported balance sheet matches ground truth exactly. BS_{recon} instead parses the journal entries produced by the model, applies those entries to the opening balance using our deterministic ledger, and checks whether the resulting balance sheet matches ground truth. The two diverge when the model writes entries with the right accounts and amounts, but its aggregated balance sheet is wrong. Evaluating six contemporary LLMs (Gemini 3 Flash, GPT-5, Claude Haiku 4.5, Grok-4.3, DeepSeek Chat, Qwen 3 235B) yields three findings. (i) An aggregation gap: four of six show 26–41 pp between BS_{recon} and BS_{exact} . The remaining two lack the gap for different reasons: GPT-5 is comparatively internally consistent, while Grok-4.3 has a smaller gap but lower entry recall (Section 6). (ii) Document-linking is the dominant entry-level failure. On the compact 143-record ablation split, document-reference mismatches affect 52% of expected postings, a rate that barely moves under explicit citation-pressure prompting (−1.4 pp). The failure concerns selecting the correct supporting documents instead of prompt compliance. (iii) When we use the deterministic ledger as a diagnostic tool and return its replayed balance sheet and account-level deltas to the model, BS_{exact} rises by +30 to +33 pp across three model families. The same feedback protocol reveals a trade-off by degrading inconsistency-code detection by 13 to 52 pp on smaller models.

Contributions. (1) A multi-document accounting reconciliation benchmark with deterministic ground truth, 23 inconsistency codes, and a released generator¹ that can create new datasets. (2) A six-model evaluation on 710 records revealing an aggregation gap, with the highest evaluated model reaching 46% BS_{exact} , plus failure breakdowns showing which industries, ledger families, concept flags, and error types drive the aggregate scores. (3) A diagnostic ablation suite covering citation pressure, context stress, prompt-described and provider-native tool access, best-of-3 selection, and forced ledger feedback. (4) Expert validation by practising finance professionals, including an independently double-reviewed subset and a certified-

accountant review spanning every industry–period–difficulty cell and every inconsistency code.

2 Related Work

Financial-NLP question answering. FinQA (Chen et al., 2021) and ConvFinQA (Chen et al., 2022) pose numerical questions over single financial reports and their tabular content, with the latter requiring conversational multi-turn reasoning. TAT-QA (Zhu et al., 2021) similarly evaluates models on hybrid tabular-and-text questions over filings. DocFinQA (Reddy et al., 2024) extends this paradigm to longer financial documents, and MultiHierrt (Zhao et al., 2022) targets reasoning over multiple hierarchical tables and accompanying text. These benchmarks assume that a prepared structured artifact, such as a table extracted from a filing or a single report, already exists and ask the model to answer questions *over* it. FINBALANCE instead requires the model to construct a statement from a bundle of raw source documents with OCR text.

Financial benchmark suites. FinanceBench (Islam et al., 2023), FinanceMATH (Zhao et al., 2024), FinBen (Xie et al., 2024), FinDABench (Liu et al., 2025), FinTagging (Wang et al., 2025), FinRule-Bench (Malarkkan et al., 2026), FinMaster (Jiang et al., 2026), and FinAuditing (Wang et al., 2026) broaden financial evaluation to filings, math, tagging, taxonomy-structured auditing, accounting principles, and workflow-style tasks (Table 1). Most evaluate answers, tags, judgments, or reasoning over prepared artifacts. FinMaster comes closest by simulating transaction-to-statement workflows and including statement-generation and auditing tasks, but its simulator omits a general ledger and trial balance, while the benchmark stops short of a citation-grounded ledger and balance sheet reconstructed from raw source-document bundles. FINBALANCE fills this gap through computed ledger ground truth that makes consistency between journal entries and their aggregated balance sheet directly measurable.

Inference-time diagnostics. Self-consistency decoding (Wang et al., 2023), refinement methods (Madaan et al., 2023; Shinn et al., 2023), trained verifiers (Cobbe et al., 2021; Lightman et al., 2024), retrieval augmentation (Lewis et al., 2020), long-context analyses (Liu et al., 2024), and tool-use agents (Yao et al., 2023) motivate our diagnostics. Our diagnostics use the benchmark ledger,

¹Code and data: <https://github.com/Devansh1105/finbalance>.

evidence-only visibility, context stress, and tool conditions to localize failures in this accounting task.

3 Benchmark Design

Task. A FINBALANCE record is a source-document bundle containing templated OCR text for one fiscal period of a synthetic business. [Figure 1](#) summarizes the model-visible inputs and hidden scoring labels. Each bundle contains *posting* documents (invoices, bills, statements, payment notices, and vouchers), *support* documents (contracts, schedules, rate sheets, and certificates), *adjustment* documents (period-end accruals, rollforwards, and reclassifications), an *opening trial balance*, and neutralised but visually plausible *distractor* documents. Given the bundle, opening trial balance, and allowed account list, the model returns JSON containing journal entries, a final balance sheet, and an inconsistency assessment. The journal entries comprise balanced debit and credit postings with an account, amount, date, and list of supporting-document references (`doc_refs`). The balance sheet is structured by section, while the inconsistency assessment contains a flag, code, and justification.

Deterministic ledger ground truth. FINBALANCE fixes the chart of accounts, accounting policies, tax assumptions, and transaction generator, making its labels deterministic. The benchmark scores the canonical treatment defined by these visible policies. The human-authored specification defines the business scenarios, document schemas, tax and FX rules, lease, revenue, and deferred-tax treatments, supporting-document dependencies, distractors, and inconsistency perturbations. The generator selects scenarios, builds subledgers, renders PDF-style documents and OCR text, validates standard records, and replays the ledger to produce the journal entries and final balance sheet. [Appendix Figure 9](#) diagrams this generation and scoring workflow. [Appendix Figures 7–8](#) shows complete sample records. Every label is checked against double-entry invariants. No record is hand-labeled after generation.

Coverage. Records are stratified across eight industries (professional services, field services, retail, wholesale distribution, healthcare clinics, property management, manufacturing, and subscription SaaS), three period types (month, quar-

ter, and year), and five difficulty levels (L1–L5). Difficulty is cumulative by construction. Each level retains the lower-level requirements while adding document volume, cross-document dependencies, jurisdictional complexity, temporal look-back, or advanced accounting concepts. *Concept flags* include ASC 606 revenue recognition, lease accounting, deferred tax, asset disposals, FX scenarios, multi-jurisdictional tax with exemptions, and rollforwards. [Appendix B](#) gives the complete list. The main evaluation split contains 710 records: four standard records for every industry–period–difficulty cell (480 total) and ten forced-inconsistency records for each of the 23 forced-inconsistency codes (230 total). We retain a compact 143-record split for targeted ablations. The deterministic seeded generator allows users to generate new datasets, create held-out evaluation sets, and extend the scenario space without hand-labeling each record. [Appendix B](#) gives the construction rubric and practitioner judgments.

Forced-inconsistency records. A 23-code taxonomy of deliberate contradictions covers banking, billing, contract, jurisdictional, and reconciliation errors, including closing-balance conflicts, invoice-total mismatches, tax-total mismatches, and invalid exemption certificates. These records test whether a model recognizes that the source documents are internally inconsistent and withholds a fabricated balance sheet. In a forced-inconsistency record, the generator first builds a standard record and then applies a code-specific perturbation that makes the document bundle unreconcilable. The model must report that the bundle is inconsistent, emit the correct code, and leave the journal entries and balance sheet empty. Forced-inconsistency records are 32% of the main evaluation split. [Appendix Table 2](#) explains every code and reports per-code detection rates.

Metrics. For standard records, BS_{exact} measures whether the model’s reported balance sheet matches ground truth to a tolerance of 0.01. BS_{recon} parses the model’s journal entries, applies them to the opening balance with our deterministic ledger, and checks whether the resulting balance sheet matches ground truth. The difference $BS_{\text{recon}} - BS_{\text{exact}}$ is the aggregation gap. Because BS_{recon} is an outcome-level replay metric, it can succeed even when the predicted entry set is not an exact posting-level match. Entry matching operates at the posting level. A strict match requires

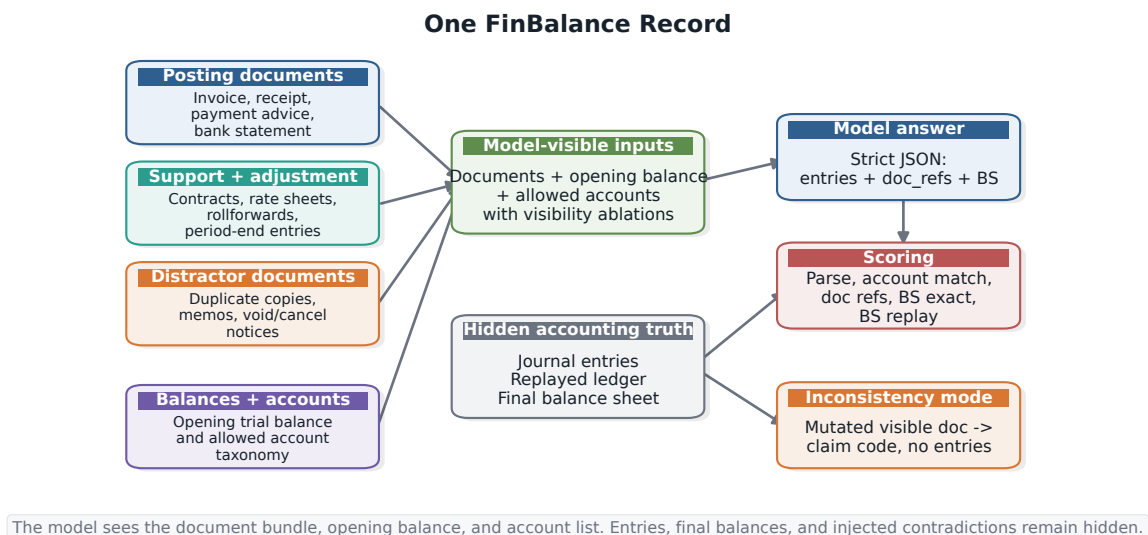


Figure 1: One FINBALANCE record. The model sees a visible document bundle of posting, support, adjustment, and distractor documents plus the allowed account taxonomy. Hidden journal entries, balance sheets, and inconsistency labels remain available only to the deterministic scorer. For each visible document, the baseline prompt contains only its ID, type, title, and OCR text. The OCR-text-only condition retains only document IDs and OCR text, dropping type and title. Posting, support, and distractor annotations are used only by the generator and scorer and are never shown to the model.

the debit account, credit account, rounded amount, and sorted supporting-document references to be correct. The lenient variant ignores supporting-document references. Balance-sheet and entry metrics do not apply to forced-inconsistency records because the correct output is an inconsistency flag and code with empty entries and an empty balance sheet. We therefore report inconsistency-code match rate separately as the rate of correct flag-plus-code predictions over the 230 forced-inconsistency records. Strict end-to-end accuracy requires the reported balance sheet, journal entries, and supporting-document citations to match. Parse rate is reported over all records. [Appendix D](#) gives the full matching procedure.

Ledger-feedback diagnostic. One ablation uses the ledger as inference-time feedback to test whether a model can repair its reported balance sheet after receiving a deterministic consistency check. The evaluation procedure reads the model’s structured answer, replays its entries, computes the implied balance sheet, and compares each account and section total with the model’s reported balance sheet. It returns the full replayed balance sheet together with account-level and total deltas. For example:

account: assets/Cash

submitted: \$48,500
replayed: \$51,300
delta: \$-2,800

The feedback contains no ground-truth values or entry-correctness hints. The model receives one revision turn. This replay-and-revision protocol provides structured account-level feedback.

4 Experimental Setup

Models and decoding. We evaluate six contemporary LLMs: Gemini 3 Flash (primary ablation model), GPT-5 (low reasoning effort), Claude Haiku 4.5, Grok-4.3, Qwen 3 235B, and DeepSeek Chat. This panel covers a range of closed, open, and general-purpose model families. All six receive the full main evaluation split ($n=710$) at baseline. The ledger-feedback diagnostic and other targeted ablations are run on the compact 143-record ablation split, where every industry–period–difficulty cell and every inconsistency code is represented at least once. Decoding uses temperature 0.0, except best-of-3 selection (temperature 0.7, three samples). Additional controls test reasoning level ([Appendix G.1](#)), provider-native function calling ([Appendix G.2](#)), character corruption ([Appendix G.3](#)), and run/split stability through repeated decoding and a new seed ([Appendix G.4](#)). These diagnostics supplement the six-model headline panel.

Ablations. We run four ablation families and report all effects against the same-model baseline. (1) **Prompt** tests guided solving, self-checking, and stronger citation instructions. Model behavior is comparatively insensitive to these prompt changes. (2) **Visibility** removes the account list, support documents, metadata, or non-evidence documents. These runs show that accounts and support evidence are necessary and that the evidence-only condition improves document grounding. (3) **Context stress** adds distractors or moves relevant evidence later to separate longer-context errors from accounting-logic errors. Context hurts performance, while controlled-context records remain difficult. (4) **Tool/repair** tests calculators, search, the ledger-feedback tool, and best-of-3 sampling. Prompt-described tool access produced no calls, so an additional control exposes the same tools through provider-native function calling. Ledger feedback repairs many reported balance sheets but weakens inconsistency detection. [Appendix F](#) gives the evaluated grid, and [Appendix G.2](#) separates the two tool protocols. For uncertainty, we resample record IDs 5000 times and recompute each baseline-ablation difference on the same resampled records, yielding intervals for BS_{exact} , BS_{recon} , strict/lenient joint match, inconsistency-code match rate, and document-reference mismatch.

Human validation. We released 75 expert-validation records spanning all eight industries, all five difficulty levels, and 15 inconsistency codes. Reviewers were practising finance professionals at two major U.S. investment banks and worked from self-contained Markdown records. They assessed accounting plausibility, entry completeness, document support, and calibration, while deterministic generator checks covered arithmetic invariants. Records 1–25 were independently double-reviewed. Collapsing acceptable judgments (“Yes”/“Mostly” or “Accept”/“Accept with minor fixes”) gives 100% agreement on document analogy, entries, supporting-document references, and the overall verdict, and 96% on difficulty calibration. Strict exact-option agreement is lower where reviewers differ on severity rather than acceptability: 64% on analogy, 96% on entry correctness, 100% on entry completeness, 72% on citation support, 96% on difficulty, and 80% overall. Separately, a certified accountant independently reviewed the compact coverage split (143 records: 120 standard + 23 forced-inconsistency). That pass

covers every industry–period–difficulty cell and every inconsistency code. The review accepted every record. A small number received minor tightening notes on document references or wording, while all 23 forced-inconsistency records were confirmed to contain a real contradiction. [Appendix H](#) gives the protocol, assignment, and marginals.

5 Results

FINBALANCE is difficult for the models in our evaluation panel. [Figure 2](#) gives the baseline profile: the highest BS_{exact} score is 46%, strict document-grounded end-to-end accuracy remains low, and the best entry-level accounting scores are much higher than the final balance-sheet scores. JSON formatting is therefore not the main bottleneck. Models often identify numerically plausible postings, but they lose correctness when those local decisions must be tied back to documents and aggregated into the final statement.

The central pattern in [Figure 2](#) is the gap between BS_{exact} and BS_{recon} , where BS_{recon} parses the model’s journal entries, applies them to the opening balance with our ledger, and checks the resulting balance sheet against ground truth. In four of six models the two metrics disagree dramatically: DeepSeek Chat reaches 0.473 BS_{recon} but only 0.067 BS_{exact} , a 40.6 pp aggregation gap. Gemini 3 Flash (+31.3 pp), Qwen 3 235B (+29.0 pp), and Claude Haiku 4.5 (+26.3 pp) show the same pattern. Because BS_{recon} ignores document support, the metric alone cannot establish audit-grounded entries. The observed gap shows that models often recover plausible entries, fail to bind them to evidence, and then fail to aggregate them into an internally consistent reported statement. GPT-5 is comparatively internally consistent (46.5% BS_{exact} , 46.3% BS_{recon}), while Grok-4.3 has a smaller gap but lower entry recall ([Section 6](#)).

GPT-5 and Grok-4.3 used reasoning in the headline panel. Compact-split controls associate reasoning with a smaller gap. Switching from DeepSeek Chat to DeepSeek Reasoner raises BS_{exact} from 8.3% to 54.2% and shrinks the gap from 42.0 to 1.7 pp. Higher-reasoning Haiku and Qwen configurations move in the same direction, while disabling Grok reasoning creates a 27.5 pp gap. Strict accuracy remains 0–4.2% across these controls, and inconsistency-code effects are mixed. Improved aggregation consistency therefore does not yield evidence-grounded reconciliation ([Appendix G.1](#)).

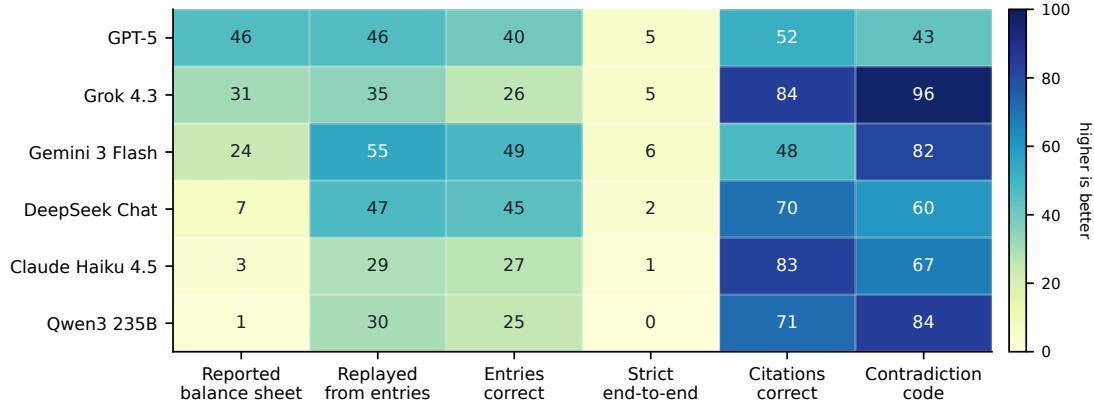


Figure 2: Baseline model profile on the 710-record core evaluation split. Balance-sheet and entry metrics use the 480 standard records. *Contradiction code* uses the 230 forced-inconsistency records. *Reported balance sheet* is BS_{exact} . *Replayed from entries* is BS_{recon} , the balance sheet implied by the model’s own journal entries. *Entries correct* ignores document citations. *Strict end-to-end* requires the balance sheet, entries, and citations to match. *Citations correct* is one minus document-reference mismatch incidence per expected posting.

To test whether the gap reflects aggregation errors beyond entry errors, we apply the model’s own entries through the ledger. In this feedback condition (Section 3), the ledger returns the full balance sheet implied by the model’s entries, account-level and total deltas against its reported balance sheet, and one revision turn. Figure 3 shows that BS_{exact} gains +30 to +33 pp across three model families with a large gap, all significant at 95% paired bootstrap. The estimate compares an independent same-model baseline with the feedback condition. Feedback contains only replayed balances and deltas, without ground-truth values or entry-correctness hints.

Ledger feedback also reduces inconsistency-code accuracy. On the compact ablation split (Figure 3, right), Gemini 3 Flash loses 13.0 pp (95% CI $[-30.4, +4.3]$, not significant), whereas Qwen 3 235B loses 17.4 pp (95% CI $[-34.8, -4.3]$) and Claude Haiku 4.5 loses 52.2 pp (95% CI $[-73.9, -30.4]$). Ledger feedback may focus revisions on reconciling the bundle instead of reconsidering its consistency. A targeted causal test is needed to distinguish this mechanism.

Tool use depends strongly on the interface. Under the prompt-described protocol, models were instructed to request tools by emitting JSON, but none produced an executable call across 572 compact-split model-record evaluations. When the same tools were exposed through provider-native function calling, models invoked them on 141/143 Gemini 3 Flash, 134/143 DeepSeek Chat, 56/143 GPT-5, and 140/143 Claude Haiku 4.5

records. Native BS_{exact} is 52.5%, 45.8%, 52.5%, and 23.3%, respectively, but strict end-to-end accuracy remains only 3.3–5.8%. Native tools recover much of the aggregation component when invoked, but strict grounding remains low (Appendix Table 8).

Document linking is the second major failure mode. On the compact split, document-reference mismatches affect 52.4% of expected postings under the greedy matcher, collapsing strict joint accuracy to 5% despite 57% entry-match when citations are ignored. Citation-pressure prompting changes this incidence by only -1.4 pp (CI $[-2.9, 0.0]$), while the *evidence-only* condition lowers it by -26.4 pp (CI $[-29.8, -23.0]$). This contrast locates the main failure in selecting supporting documents instead of following citation instructions (Figure 4).

The primary citation score requires the exact reference set, so we also report a superset-tolerant sensitivity score. Of mismatch postings, 42.0% for Gemini 3 Flash and 58.8% for GPT-5 contain every expected reference plus extras, and 94–96% of those extra citations are support documents. Under the sensitivity score, strict end-to-end accuracy rises to 8.8% and 15.8%, respectively, but remains low. DeepSeek Chat changes from 2.5% to 3.3% (Appendix G.5).

The uncorrupted templated-text setting isolates accounting reasoning from OCR and layout errors. Seeded substitutions, deletions, token splits, and token merges at achieved character-edit rates of 0.94%, 2.20%, and 4.45% reduce BS_{recon} mono-

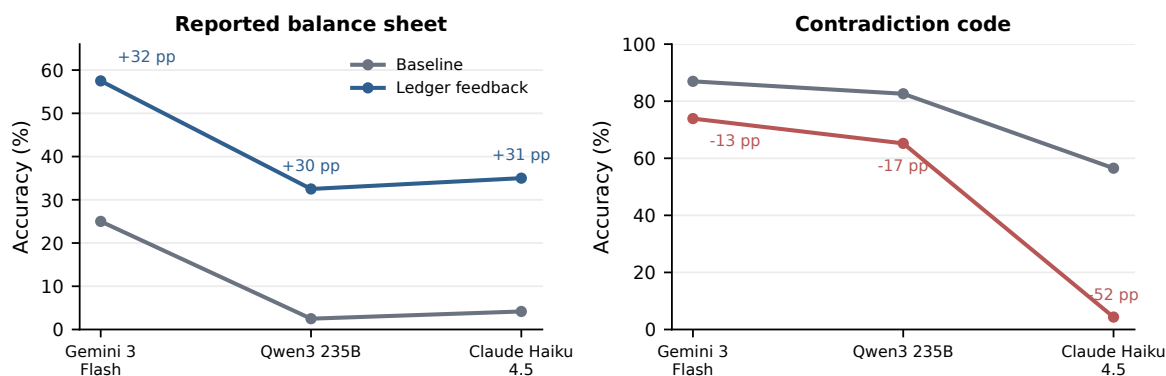


Figure 3: One-pass ledger-feedback effect on the compact ablation split ($n=143$ per model, with BS_{exact} over $n=120$ standard records and inconsistency-code match rate over $n=23$ forced-inconsistency records). Lines compare each model’s independent baseline with the same model after feedback. Labels are percentage-point changes. The tool strongly improves reported balance-sheet accuracy (left) but can reduce contradiction-code accuracy (right). GPT-5 and Grok-4.3 are omitted because they do not exhibit the same aggregation-gap mechanism (Table 4).

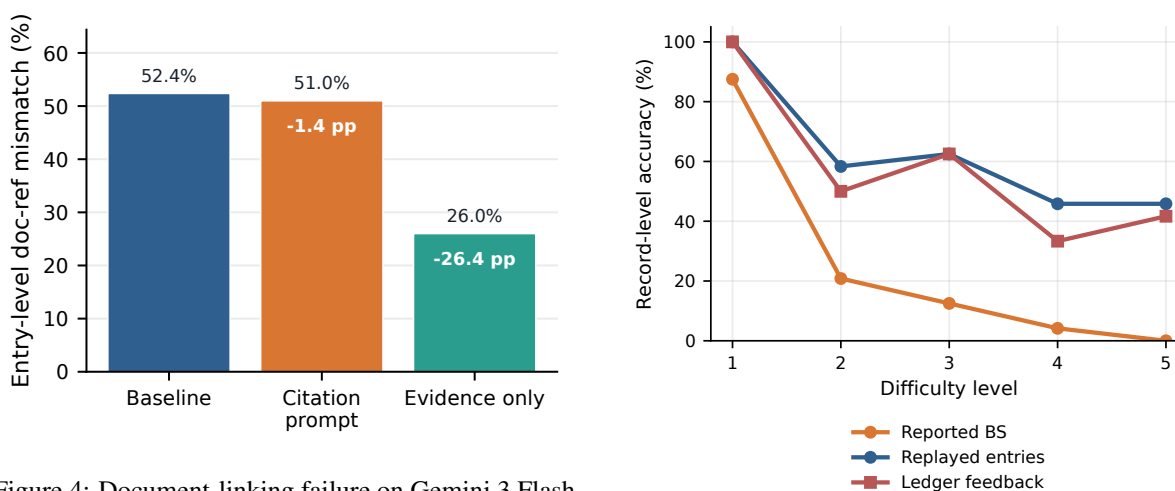


Figure 4: Document-linking failure on Gemini 3 Flash over the compact split ($n=143$, with the posting metric over $n=120$ standard records) persists under explicit citation pressure but drops under the evidence-only condition.

tonically: Gemini 3 Flash falls from 61.7% uncorrupted to 49.2%, and DeepSeek Chat from 50.0% to 36.7%. Parse success remains 99–100%. This control covers character corruption of templated text and excludes raw OCR output and document layout (Appendix G.3).

We also test selection among multiple candidates. On the 45 standard records where Gemini’s entries replay to the correct balance sheet but its reported balance sheet is wrong, a ledger-aware best-of-3 selector raises BS_{exact} from 0% to 28.9% (CI [15.6, 42.2]), while one ledger-feedback revision reaches 84.4% (CI [73.3, 93.3]). Candidate selection recovers part of the gap. Revision after structured ledger feedback recovers much more

Figure 5: Accuracy by difficulty level for Gemini 3 Flash on the compact split ($n=143$, with 24 standard records per level). As records move from L1 to L5, BS_{exact} collapses much more sharply than the context-stress penalty in Figure 6. BS_{recon} remains higher, showing that entry replay remains easier than reporting the final balance sheet.

(Appendix Figure 14).

Appendix Table 6 distills the ablation results by diagnostic question. Forced ledger-feedback variants produce the only substantial upward movement in BS_{exact} . Removing scaffolding (allowed accounts and support documents) is the most damaging ablation. Prompt-only variants, including a stronger citation instruction and a guided solve protocol, leave the large effects above unchanged. The full primary-model delta plot is in Appendix Figure 13.

Figures 5 and 6 compare context length with

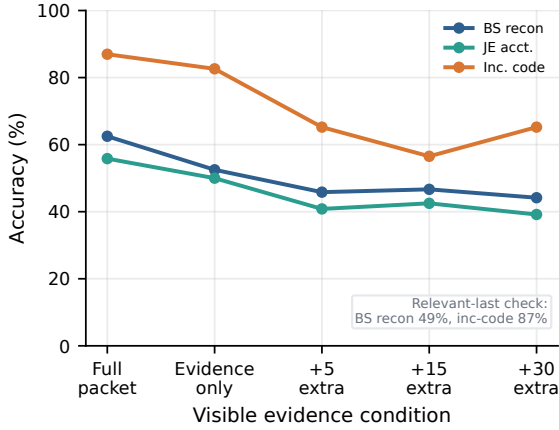


Figure 6: Context stress for Gemini 3 Flash on the compact standard-record split ($n=120$), designed to separate longer-context effects from the underlying accounting-logic difficulty. The task remains difficult under controlled evidence. Distractors hurt most when first introduced and then saturate.

accounting-logic difficulty. Difficulty produces the larger effect: baseline BS_{exact} falls from 88% at L1 to 0% at L5, while BS_{recon} declines from 100% to 46%. Adding 5, 15, or 30 distractors to an evidence-only document bundle decreases BS_{recon} by 6.7, 5.8, and 8.3 pp, respectively. Context accounts for part of the degradation. Accounting dependencies and document choice remain binding even when context is controlled.

6 Analysis

Why two models lack the gap. The no-gap models differ. GPT-5 submits 91% of expected entries and has BS_{exact} essentially equal to BS_{recon} (0.465 vs. 0.463). Grok-4.3 submits only 40% of expected entries, so its balance sheet is internally aligned but incomplete (Appendix Table 4).

Where the gap concentrates. On the compact split, BS_{exact} drops from 0.88 at L1 to 0.00 at L5 while BS_{recon} declines more gently. The aggregation gap peaks near L3 at roughly 46 pp (Figure 5). Ledger feedback narrows the gap at every level. Industry, ledger-family, and concept slices in Appendix C and Figure 11 show where the benchmark stresses models beyond the aggregate score.

Error taxonomy. On the 480 standard records, document-linking failures dominate the primary ablation model: 93.1% of records and 3890 entries, compared with 1033 omissions, 584 extras, 202 amount mismatches, and 85 account mismatches.

The hard part is grounding plausible entries to the right evidence. Appendix Figure 11 breaks failures down by industry, ledger family, and error type.

7 Discussion

Long-context retrieval alone leaves source-document accounting errors unresolved. Distractors hurt, and controlled-evidence records still degrade as cross-document accounting dependencies grow because models must select evidence, form balanced entries, and aggregate them. Repair ablations target the entry-to-balance-sheet aggregation stage. The larger gain from ledger feedback implicates aggregation errors beyond insufficient sampling. Strict document-grounded accuracy remains low. A targeted causal test is needed to explain the reduction in inconsistency detection.

Intended utility. Beyond fixed evaluation, the released generator can create stratified practice data and organization-specific stress tests, while the deterministic ledger can provide machine-checkable feedback for training or candidate selection. Possible uses include supervised examples, verifier/reward signals, and distillation of tool-assisted workflows.

8 Conclusion

FINBALANCE makes document-grounded accounting reconciliation a concrete, computable benchmark: models must select evidence, form cited ledger entries, and aggregate a balance sheet. Across six LLMs, the end-to-end task remains difficult. Replayed entries often outscore reported balance sheets, while strict citation-grounded accuracy stays low. The released generator supports new datasets and larger splits from the same human-authored accounting specification.

Limitations

Synthetic construction. FINBALANCE is synthetic: it combines human-authored business scenarios, document schemas, accounting policies, tax and FX assumptions, support-document dependencies, distractors, and contradiction templates with programmatic sampling, rendering, and ledger replay. This design provides inspectable labels and controlled counterfactuals but limits document fidelity. Real accounting document bundles contain messier layouts, OCR errors, missing pages, handwriting, ambiguous vendor formats, inconsistent

naming, and institution-specific artifacts that our rendered documents only approximate. The benchmark measures reasoning over templated OCR text under these controlled conditions. The character-corruption control covers seeded text noise and excludes raw OCR-engine output and rendered layout. Consequently, the experiments leave the contribution of flattened layout to document-linking failure unresolved.

Accounting coverage. Accounting is broad and policy-dependent. Our labels are deterministic after fixing the chart of accounts, accounting policy, tax assumptions, and transaction generator. We score one canonical treatment under the visible policies. Equivalent decompositions, presentation variants, and other professionally acceptable treatments remain outside the current exact-match scope. We cover multi-document postings, period-end adjustments, tax, foreign exchange, leases, deferred tax, revenue schedules, subledgers, distractors, and forced contradictions. The current scope excludes the full range of GAAP, IFRS, local tax regimes, materiality judgments, industry-specific policy choices, and audit procedures. Future policy variants could enumerate multiple acceptable answer sets. The benchmark defines a precise document-grounded accounting setting within this fixed policy space.

Analysis depth. The ablations show several patterns, including the aggregation gap, weak gains from prompt-only changes, document-linking failures, context/evidence effects, and the ledger-feedback trade-off with inconsistency detection. The analysis is behavioral and leaves causal mechanisms and cross-domain transfer open. The released evaluation framework supports follow-up studies with broader ablations, controlled perturbations, model-family comparisons, and mechanistic or representation-level analyses of why models fail after the relevant evidence is present.

Evaluation scope. Headline numbers use a single temperature-0 decoding run except for the best-of-3 condition, so the paper emphasizes large paired effects over single-point differences. Six models are evaluated on the full 710-record core split. Targeted ablations use the compact 143-record split, which preserves coverage over every industry–period–difficulty cell and inconsistency code. Repeated-run controls measure stability, while commercial APIs may drift despite

pinned request/response identifiers. Released outputs support exact offline rescoring, whereas future API calls may differ. The evidence supports an aggregation gap for this evaluated model panel. Broader generalization requires additional model families.

Benchmark longevity. The public core split may become contaminated as released benchmarks enter model training data. FINBALANCE therefore includes a generator alongside the fixed test file. Users can sample new seeded document bundles from the same human-authored scenario space, create larger train/test splits, and add new industries, document schemas, policies, contradiction templates, and ablation conditions without hand-labeling each record. A fixed release enables comparison across studies, while periodically generated held-out bundles and community extensions can reduce contamination over time.

Ethical Considerations

FINBALANCE evaluates research systems under synthetic conditions and does not certify accounting competence. Professional review remains necessary when selecting or deploying systems for bookkeeping, audit, tax, regulatory reporting, or financial decision-making. Applying model outputs to real records without qualified human oversight could propagate incorrect postings, overlook contradictory evidence, or produce materially misleading statements. The synthetic scope and exact-match metrics leave those deployment risks unmeasured.

The benchmark uses synthetic data generated without real records, personally identifiable information, or production accounting data. [Appendix I](#) reports reproducibility and release details.

Acknowledgments

Generative AI tools were used for coding assistance and for language, wording, and formatting edits throughout the manuscript. The authors developed and verified the benchmark design, accounting logic, experiments, analyses, interpretations, and final claims, and they remain responsible for all AI-assisted code and text.

References

Zhiyu Chen, Wenhui Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa,

- Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. [FinQA: A dataset of numerical reasoning over financial data](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. [ConvFinQA: Exploring the chain of numerical reasoning in conversational finance question answering](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6292, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. 2023. [FinanceBench: A new benchmark for financial question answering](#). *Preprint*, arXiv:2311.11944.
- Junzhe Jiang, Chang Yang, Aixin Cui, Sihan Jin, Yujing Zhang, Yilin Xiao, Ruiyu Wang, Bo Li, Xiao Huang, Danny Dongning Sun, and Xinrun Wang. 2026. [FinMaster: A holistic benchmark for full-pipeline financial management with large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 7787–7844, San Diego, California, United States. Association for Computational Linguistics.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive NLP tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. [Let’s verify step by step](#). In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. [Lost in the middle: How language models use long contexts](#). *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Shu Liu, Shangqing Zhao, Chenghao Jia, Xinlin Zhuang, Zhaoguang Long, Jie Zhou, Aimin Zhou, Man Lan, and Yang Chong. 2025. [FinDABench: Benchmarking financial data analysis ability of large language models](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 710–725, Abu Dhabi, UAE. Association for Computational Linguistics.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. [Self-Refine: Iterative refinement with self-feedback](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 46534–46594. Curran Associates, Inc.
- Arun Vignesh Malarkkan, Manan Roy Choudhury, Guangwei Zhang, Vivek Gupta, Qingyun Wang, Yanjie Fu, and Denghui Zhang. 2026. [FinRule-Bench: A benchmark for joint reasoning over financial tables and principles](#). *Preprint*, arXiv:2603.11339.
- Varshini Reddy, Rik Koncel-Kedziorski, Viet Dac Lai, Michael Krumbick, Charles Lovering, and Chris Tanner. 2024. [DocFinQA: A long-context financial reasoning dataset](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 445–458, Bangkok, Thailand. Association for Computational Linguistics.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. [Reflexion: Language agents with verbal reinforcement learning](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 8634–8652. Curran Associates, Inc.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-Consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations*.
- Yan Wang, Lingfei Qian, Xueqing Peng, Yang Ren, Keyi Wang, Yi Han, Dongji Feng, Fengran Mo, Shengyuan Lin, Qinchuan Zhang, Kaiwen He, Chenri Luo, Jianxing Chen, Junwei Wu, Chen Xu, Ziyang Xu, Jimin Huang, Guojun Xiong, Xiao-Yang Liu, and 2 others. 2025. [FinTagging: Benchmarking LLMs for extracting and structuring financial information](#). *Preprint*, arXiv:2505.20650.
- Yan Wang, Keyi Wang, Shanshan Yang, Jaisal Patel, Jeff Zhao, Fengran Mo, Xueqing Peng, Lingfei Qian, Yankai Chen, Víctor Gutiérrez-Basulto, Jimin Huang, Guojun Xiong, Xiao-Yang Liu, and Jian-Yun Nie. 2026. [FinAuditing: A financial taxonomy-structured multi-document benchmark for evaluating LLMs](#). In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’26*, pages 3456–3463, New York, NY, USA. Association for Computing Machinery.

Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, Yijing Xu, Haoqiang Kang, Ziyang Kuang, Chenhan Yuan, Kailai Yang, Zheheng Luo, Tianlin Zhang, Zhiwei Liu, Guojun Xiong, and 15 others. 2024. [FinBen: A holistic financial benchmark for large language models](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 95716–95743. Curran Associates, Inc.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. [ReAct: Synergizing reasoning and acting in language models](#). In *The Eleventh International Conference on Learning Representations*.

Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. 2022. [MultiHiertt: Numerical reasoning over multi hierarchical tabular and textual data](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6588–6600, Dublin, Ireland. Association for Computational Linguistics.

Yilun Zhao, Hongjun Liu, Yitao Long, Rui Zhang, Chen Zhao, and Arman Cohan. 2024. [FinanceMATH: Knowledge-intensive math reasoning in finance domains](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12841–12858, Bangkok, Thailand. Association for Computational Linguistics.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. [TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.

A Dataset Examples

Figures 7 and 8 expand two core-evaluation records into the pieces a model receives and the hidden labels the scorer expects. The first record is a forced-inconsistency example: the opening trial balance and documents are visible, and the correct output is an inconsistency flag and code in place of a reconciled balance sheet. The second is a consistent foreign-exchange document bundle where the model must use an invoice, an exchange-rate notice, a payment advice, and a bank row to book both the payable settlement and FX loss. The full release includes every document’s OCR text, associated document artifact, expected entries, expected balance sheet, and inconsistency labels.

B Dataset Details

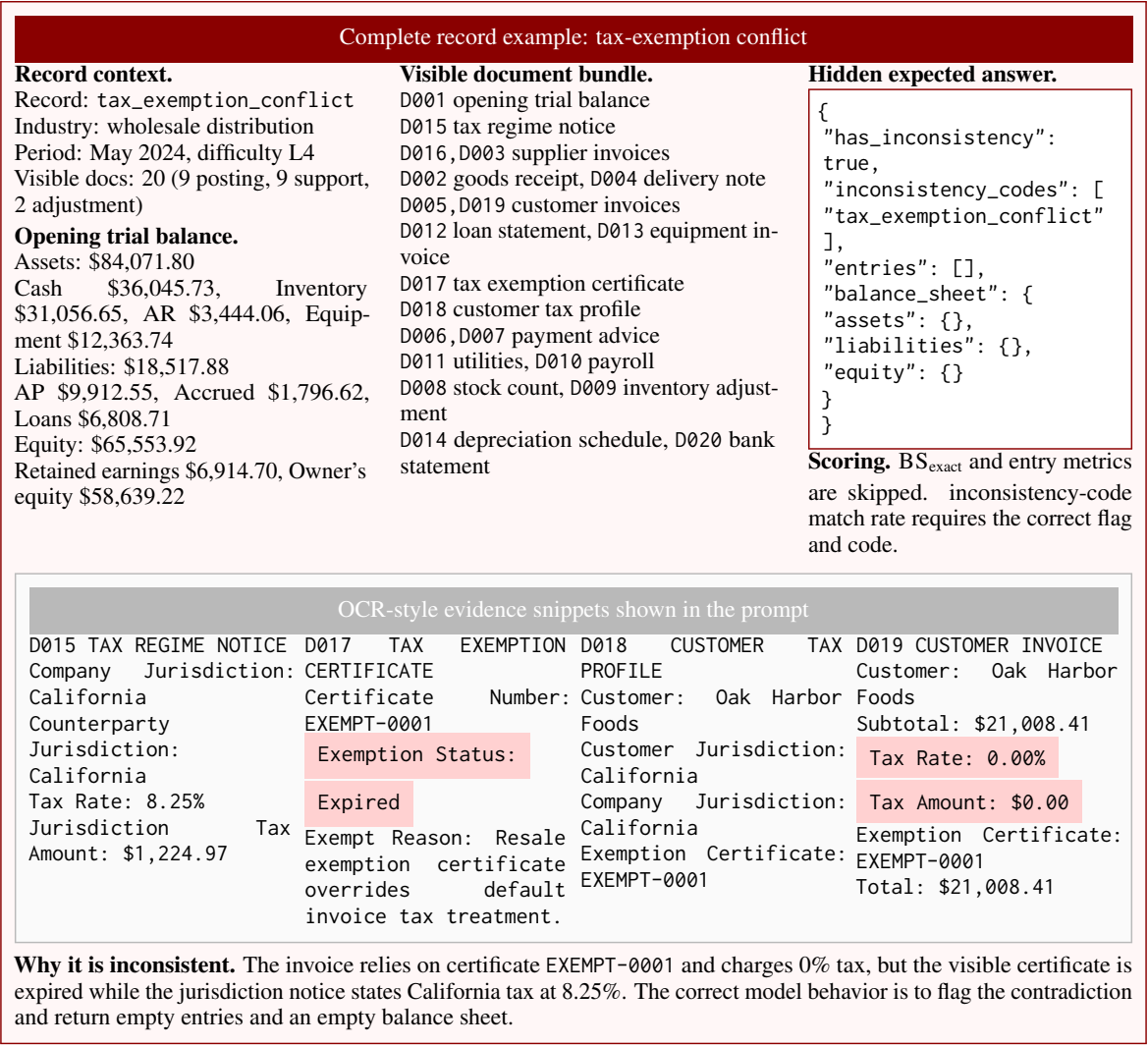
Document roles. Every document in a record carries a role tag that determines how the ledger ground-truth generator handled it.

- `posting_doc`: directly generates one or more journal entries (e.g., invoices, bills, payment notices, journal vouchers).
- `support_doc`: provides inputs for postings derived from other documents (rate cards, contracts, schedules, tax certificates) and may appear in an entry’s supporting-document references (`doc_refs`).
- `adjustment_doc`: supplies period-end corrections such as depreciation schedules, rollforwards, and reclassification memos.
- `distractor_doc`: is intentionally neutralised and produces no entry.
- `opening_trial_balance`: supplies starting balances and produces no expected entries.

Industries. Professional services, field services, retail, wholesale distribution, healthcare clinic, property management, manufacturing, and subscription SaaS.

Concept flags. Each record is annotated for the presence of several accounting concepts. These include ASC 606 revenue recognition for customer contracts with standalone selling price (SSP) allocation and fixed-asset disposals with net book value (NBV) and gain-or-loss computation. Other flags cover deferred tax from book/tax depreciation differences, lease accounting with right-of-use (ROU) asset and liability schedules, foreign-exchange invoice and settlement scenarios, and multi-jurisdictional sales tax with exemption certificates. Additional flags identify tax exemptions, multi-invoice payments, interbank transfers, reclassifications, and period-end rollforwards.

Inconsistency codes. The released forced-inconsistency taxonomy contains 23 machine-readable codes grouped into five categories. Banking and transfer codes cover closing balances, statement balances, and transfers. Invoicing and payment codes cover invoice totals, payment allocation, and duplicate references. Tax and jurisdiction codes cover rates, totals, input tax, exemption conflicts, and jurisdictional treatment. Asset, inventory, lease, and deferred-tax codes cover disposals, rollforwards, remeasurement,



Complete standard-record example: foreign-currency payable settlement

Record context.

Record: COV_MAN_Q5_0099

Industry: manufacturing

Period: Q2 FY 2025, difficulty L5

Visible docs: 24, expected entries: 23

Opening trial balance.

Assets: \$568,831.08

Cash \$242,574.01, raw materials \$110,742.08, WIP \$37,368.35, finished goods \$71,595.44, AR \$25,479.08, equipment \$81,072.12

Liabilities: \$129,500.85

AP \$41,903.51, accrued \$9,355.44, notes \$31,326.16, loans \$46,915.74

Equity: \$439,330.23

Key visible documents.

D017 supplier invoice: GBP 37,480.16, invoice FXBILL-0001.

D018 exchange-rate notice: spot rate 1.3234, functional amount \$49,601.24.

D019 payment advice: paid \$51,422.78 at rate 1.3720, with an FX difference of \$1,821.54.

D023 bank statement: row "Foreign payment FXBILL-0001" for \$-51,422.78.

Why this is consistent.

The four documents agree: the payable was booked at the bill-date rate, settled at a higher USD amount, and the difference is a realized FX loss.

Hidden expected FX entries.

1. Dr Raw Materials Inventory / Cr Accounts Payable, \$49,601.24, refs D017, D018.

2. Dr Accounts Payable / Cr Cash, \$49,601.24, refs D019, D017.

3. Dr Foreign Exchange Loss / Cr Cash, \$1,821.54, refs D019, D017.

Final balance-sheet check.

Assets \$719,627.12, liabilities \$330,057.95, equity \$389,569.17, balanced: true.

OCR-style evidence snippets shown in the prompt

D017 SUPPLIER INVOICE

Invoice Number: FXBILL-0001

Currency: GBP

Total: GBP 37,480.16

Lines: consulting sprint GBP 9,173.27, foreign-currency support GBP 28,306.89

D018 EXCHANGE RATE

NOTICE

Reference: FXBILL-0001

Rate Date: 2025-04-25

Source Amount: GBP 37,480.16

Functional Currency: USD

Exchange Rate: 1.3234

Functional Amount: \$49,601.24

D019 PAYMENT ADVICE

Reference: FXBILL-0001

Currency: USD

Amount: \$51,422.78

Exchange Rate: 1.3720

FX Difference: 1,821.54

Payment For: payable settlement

D023 BANK STATEMENT

Statement Currency: USD

Row: 2025-06-06

Description: Foreign payment FXBILL-0001

Amount: \$-51,422.78

Rows generated from internal cash ledger.

Figure 8: One complete standard foreign-exchange record. The model must connect a GBP supplier invoice to a bill-date exchange-rate notice, settlement advice, and bank statement row. The expected answer books the payable at the bill-date functional amount, settles cash at the payment amount, and records the realized FX loss. Unlike Figure 7, this record should produce entries and a final balance sheet.

The levels are cumulative by construction: each level retains the sources of complexity below it and adds document volume, dependencies, jurisdictions, temporal lookback, or advanced concepts. The certified-accountant review judged 139/143 compact-split assignments calibrated as written (three “too easy,” one “too hard”). The independently double-reviewed overlap showed 96% agreement on difficulty calibration. These judgments support practitioner-perceived ordering for the sampled benchmark records and cover the benchmark’s authored complexity factors.

Figure 10 summarizes the released core split across industries, difficulty levels, period types, and document roles.

C Per-Concept Results

Table 3 reports baseline BS_{recon} by concept flag across all six models. Appendix B defines the terms. Each flag identifies an accounting concept

within a record. The core evaluation split covers the authored scenario families and concept flags at least once, while the released generator can create larger concept-targeted samples for training, stress tests, or future evaluation splits. The period-rollforward flag appears most often in the core split. These smaller concept slices are coverage diagnostics over the authored scenario families. The released generator can produce larger concept-targeted samples beyond these coverage slices.

D Detailed Metrics

Metric definitions. BS_{exact} requires the model’s reported assets, liabilities, and equity account maps and their totals to match ground truth at 0.01 tolerance. BS_{recon} runs the model’s entries through our ledger, computes the implied balance sheet, and applies the same equality check. It is an outcome-level metric and can succeed without an exact match to the expected post-

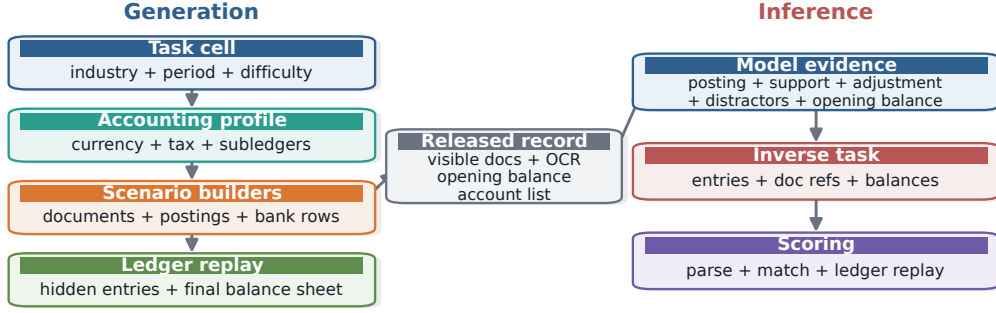


Figure 9: Benchmark construction and evaluation. The generator executes a deterministic accounting program to create source-document bundles, hidden journal entries, balance sheets, and inconsistency labels. The model sees only the visible documents and account taxonomy. Scoring replays its entries through the same ledger, enabling BS_{recon} and the ledger-feedback diagnostic.

ing set. These balance-sheet and entry metrics are reported over standard records only. On forced-inconsistency records, the expected behavior is `has_inconsistency=True`, the correct code, empty entries, and an empty balance sheet. We therefore score them with inconsistency-code match rate and the empty-answer requirement instead of BS_{exact} or entry match.

Entry matching. Journal entries are scored at the posting level. A strict posting match requires the same debit account, credit account, amount rounded to cents, and sorted supporting-document references (`doc_refs`). The lenient entry match uses the same account-and-amount criteria but ignores the reference set. We greedily match predictions in four passes: exact match, correct debit/credit/amount with an incorrect reference set, correct debit/credit/reference set with an incorrect amount, and correct reference set/amount with an incorrect account. Unmatched expected postings are missing. Unmatched predicted postings are extra. Duplicate identical postings are matched first-in-first-out. The allowed account list is closed, so the evaluator uses no account synonyms. Posting dates are parsed and retained for diagnostics but are outside the headline entry-match criterion. Split transactions receive credit only when the predicted posting decomposition matches the generator’s expected postings.

Parse rate. Across the six completed headline baselines at $n=710$, parse-success rates are at least 97% and five models parse at $\geq 99.8\%$. A DeepSeek Reasoner run on the same split produced outputs with a low parse-success rate under the

available provider conditions, so DeepSeek Chat serves as the DeepSeek-family baseline.

Two-pass ledger-feedback diagnostic. For Gemini 3 Flash, the 2-pass ledger-feedback variant changes BS_{exact} by +35.8 pp (CI [+26.9, +44.9]) compared with +32.5 pp at 1-pass, and degrades inconsistency-code match rate by −21.7 pp (CI [−40.0, −5.3]) compared with −13.0 pp at 1-pass. The main text reports the simpler 1-pass diagnostic.

E Cost Analysis

Figure 12 reports API cost per evaluated core-evaluation record against BS_{exact} accuracy. Costs are provider-billed totals for the completed baseline runs. The plot supports cost reproducibility. Changing prices and routing prevent a stable cross-provider ranking. DeepSeek Chat was run through the DeepSeek Platform, while the other completed baselines were run through OpenRouter.

F Ablation Matrix

Table 5 summarizes the prompt, visibility, and tool configurations evaluated on the primary ablation model, Gemini 3 Flash. Conditions use the compact ablation split or a labeled targeted subset, so n varies by condition. The matrix covers all single-axis variants and selected combined-condition diagnostics, preserving paired comparisons without enumerating the full cross-product.

Appendix Table 6 summarizes the measured effects, Figure 6 isolates distractor count and evidence position, Figure 4 tests whether citation pressure fixes document-linking, and Appendix Figure 14 compares best-of-3 selection against ledger-

Expected code	Meaning	Detected
invoice_total_mismatch	Invoice subtotal, tax, and stated total do not tie.	60%
bank_closing_mismatch	Bank statement closing balance conflicts with its listed activity.	98%
statement_balance_mismatch	Statement opening/closing balance conflicts with visible transaction rows.	73%
payment_allocation_mismatch	Payment advice allocates cash to invoices in a way that cannot reconcile.	85%
duplicate_reference_conflict	Two visible documents reuse the same reference for incompatible transactions.	68%
schedule_rollforward_mismatch	Rollforward schedule opening balance, movements, and closing balance do not tie.	80%
inventory_rollforward_mismatch	Inventory movement schedule cannot reconcile to purchases, COGS, or ending stock.	77%
transfer_mismatch	Interbank transfer outflow and receiving-bank inflow disagree.	97%
reclassification_support_mismatch	Reclassification memo is unsupported or contradicts the source schedule.	60%
tax_total_mismatch	Stated tax total disagrees with taxable base and tax lines.	72%
tax_rate_mismatch	Applied tax rate conflicts with the visible jurisdiction or tax notice.	65%
input_tax_mismatch	Claimed recoverable input tax is not supported by the supplier/tax documents.	23%
jurisdiction_tax_mismatch	Tax treatment uses the wrong jurisdiction or jurisdiction-specific rule.	37%
tax_exemption_conflict	Exemption certificate is invalid, expired, or conflicts with the tax charged.	82%
ssp_allocation_mismatch	Standalone selling price allocation in a bundled contract is inconsistent.	62%
performance_obligation_release_mismatch	Revenue release conflicts with the performance-obligation schedule.	28%
asset_disposal_mismatch	Disposal proceeds, net book value, and gain/loss do not reconcile.	83%
deferred_tax_rollforward_mismatch	Deferred-tax rollforward conflicts with book/tax temporary differences.	95%
lease_schedule_mismatch	Lease amortization schedule does not tie to liability, interest, or payments.	92%
lease_remeasurement_mismatch	Lease modification or remeasurement calculation conflicts with support.	98%
exchange_rate_mismatch	Exchange rate used in a document conflicts with the visible rate notice.	72%
fx_settlement_mismatch	Foreign-currency invoice settlement does not tie to cash paid or FX gain/loss.	70%
remeasurement_mismatch	Period-end FX remeasurement uses an inconsistent rate or carrying amount.	80%

Table 2: Per-code inconsistency diagnostics over the six-model baseline panel on the core split. *Detected* is the percentage of 60 attempts per code (10 records $\times$ 6 models) that set `has_inconsistency=True` and selected the expected code. These generated-instance diagnostics estimate performance on the released code samples. Population-level code difficulty requires broader sampling.

feedback revision. Across these analyses, prompt-described tool access and prompt-only variants have weak effects. Removing support evidence or the account taxonomy is strongly damaging, and forced ledger feedback is the only diagnostic that substantially improves the reported balance sheet. Figure 13 gives the full primary-model delta plot.

G Additional Diagnostics

These controls address specific confounds. They use the fixed compact split (143 records: 120 standard + 23 forced-inconsistency) unless explicitly labeled otherwise. These compact-split controls sup-

plement the six-model, 710-record headline panel.

G.1 Reasoning controls

GPT-5 and Grok-4.3 both used reasoning in the headline runs. Table 7 varies reasoning within a model family. Because reasoning effort and output budget change together, the results measure configuration effects without isolating reasoning alone.

Relative to matched non-reasoning configurations, switching from DeepSeek Chat to DeepSeek Reasoner moves BS_{exact} from 8.3% to 54.2% and reduces the gap from 42.0 to 1.7 pp. Haiku moves



Figure 10: Core evaluation split composition. The released evaluation split spans eight industries, five difficulty levels, three period types, and posting, support, adjustment, distractor, and opening-balance documents.

from 4.2% to 38.3% and from 29.1 to 8.4 pp. Qwen moves from 1.7% to 31.7% and to an 8.3 pp gap. The reverse Grok and GPT-5 controls create large gaps. On the full 710-record split, DeepSeek Reasoner reaches 44.0% BS_{exact} with a 4.8 pp gap but raises false inconsistency alarms on standard records to 29.0% from near zero. Higher-reasoning configurations therefore improve aggregation consistency unevenly across document grounding and contradiction detection.

G.2 Text versus native tool protocols

In the prompt-described protocol, the prompt described each tool and required a JSON tool request. Gemini and the two additional prompt-described runs made zero calls over 572 model-record evaluations. We then exposed the same calculator, document search, and ledger functions through each provider’s native function-calling interface with an eight-call budget. Table 8 reports the resulting invocation rates, parsing rates, and accuracies.

Claude Haiku 4.5 rises from its 4.2% baseline to 20.0% under the prompt-described protocol without producing a call, indicating a prompt-format effect. DeepSeek Chat provides the clearest execution contrast: its prompt-described score remains near baseline, whereas provider-native function

calling improves BS_{exact} by 37.5 pp. Across models, the prompt-described protocol produced zero executable calls, while provider-native interfaces were used frequently and recovered part of the aggregation deficit.

G.3 Character corruption

The headline input is uncorrupted templated text. A seeded corruption control adds visually similar-character substitutions, deletions, token splits, and token merges without line deletion. Table 9 labels conditions by achieved character-edit rate measured against the uncorrupted input.

The monotone declines show that character corruption is a measurable second source of error. At the harshest tested rate, corruption accounts for roughly one third of the uncorrupted-text shortfall.

G.4 Run and split stability

Three temperature-zero compact-split runs give the following BS_{exact} ranges: Gemini 3 Flash 24.2–25.0%, DeepSeek Chat 7.5–9.2%, GPT-5 51.7–55.0%, Claude Haiku 4.5 4.2–4.2%, Grok-4.3 33.3–36.7%, and Qwen 3 235B 0.8–2.5%. No range exceeds 3.3 pp. BS_{recon} varies by at most 3.3 pp except for GPT-5 (7.5 pp). The repeated runs measure run stability. Paired-bootstrap intervals in

Concept flag	Gemini	GPT-5	Haiku	Grok	DeepSeek	
					Chat	Qwen
Revenue recognition ($n=24$)	17%	25%	4%	4%	8%	12%
Asset disposal ($n=24$)	21%	17%	0%	0%	8%	0%
Deferred tax ($n=12$)	25%	0%	0%	0%	8%	0%
Lease accounting ($n=24$)	21%	17%	0%	0%	8%	0%
Tax exemption ($n=24$)	29%	12%	0%	0%	33%	4%
Foreign exchange ($n=24$)	8%	8%	0%	0%	0%	0%
Period rollforward ($n=212$)	48%	41%	11%	19%	39%	19%

Table 3: Baseline BS_{recon} by accounting concept on the 480 standard records in the core evaluation split. All values are percentages, and each row reports its record count. Revenue recognition corresponds to ASC 606 customer-contract revenue. Foreign exchange covers non-home-currency invoices and settlements. Period rollforwards are schedules that carry prior balances forward. The concept slices are unbalanced coverage diagnostics. The generator supports larger balanced concept-specific samples.

Model	BS_{exact}	BS_{recon}	gap	entry ratio
Gemini 3 Flash	0.235	0.548	+0.313	0.94
GPT-5	0.465	0.463	-0.002	0.91
Claude Haiku 4.5	0.029	0.292	+0.263	0.53
Grok-4.3	0.310	0.350	+0.040	0.40
DeepSeek Chat	0.067	0.473	+0.406	0.98
Qwen 3 235B	0.013	0.302	+0.290	0.94

Table 4: Three regimes on the 480 standard records in the core split. *Entry ratio* is predicted-entry count divided by expected-entry count. GPT-5 is comparatively internally consistent while submitting most expected entries. Grok-4.3 has a smaller gap partly because it submits only 40% of expected entries.

the main paper quantify record-level sampling uncertainty for fixed saved runs.

We also generated a second 710-record split from unused seed 20260712 with the same coverage design and evaluated DeepSeek Chat under its headline configuration. BS_{exact} changes from 6.67% to 7.71%, BS_{recon} is 47.29% on both, strict accuracy changes from 2.50% to 2.92%, inconsistency-code match rate from 60.00% to 60.87%, and parse success from 100.00% to 99.86%. This single-model replication measures split stability for DeepSeek Chat.

G.5 Citation-score sensitivity

Exact document-reference scoring distinguishes the generator’s expected evidence set from supersets. Table 10 accepts either the exact set or a strict superset while keeping extra predicted entries and all other strict requirements unchanged.

G.6 Structural audit and validation scope

An independent JSON-level audit uses a replay procedure implemented separately from the generator’s ledger. It checks nine invariants: unique

visible IDs, dates within the record period, resolvable citations, account-taxonomy membership, positive non-self postings, section sums, the balance identity, replay of expected entries or the forced-inconsistency empty-answer convention, and label conventions. The released 710 records and new samples of 1,200 and 4,800 records all pass every invariant (6,710/6,710). Cell-level results and audit code are released.

This audit verifies structural consistency across sampled values and combinations. Human review evaluates semantic realism, contradiction-code fit, and accounting interpretation. The 143-record accountant review covers every industry–period–difficulty cell and every code, while the separate 75-record set spans all industries and levels.

H Human Verification Protocol

Reviewer recruitment. The release subset was reviewed by two practising finance professionals working at major U.S. investment banks. Both reviewers had prior accounting and reconciliation responsibilities in their day-to-day work and received a one-paragraph briefing that scoped the task to accounting plausibility and reconciliation evidence while excluding OCR fidelity. Deterministic generator checks covered arithmetic invariants. A third reviewer, a certified accountant, independently reviewed the full compact coverage split under the same instrument and briefing. This reviewer worked separately and did not see the other reviewers’ responses. All three reviewers participated as unpaid volunteers and consented to the anonymized use and public release of their review responses. No formal institutional ethics review was sought for this activity.

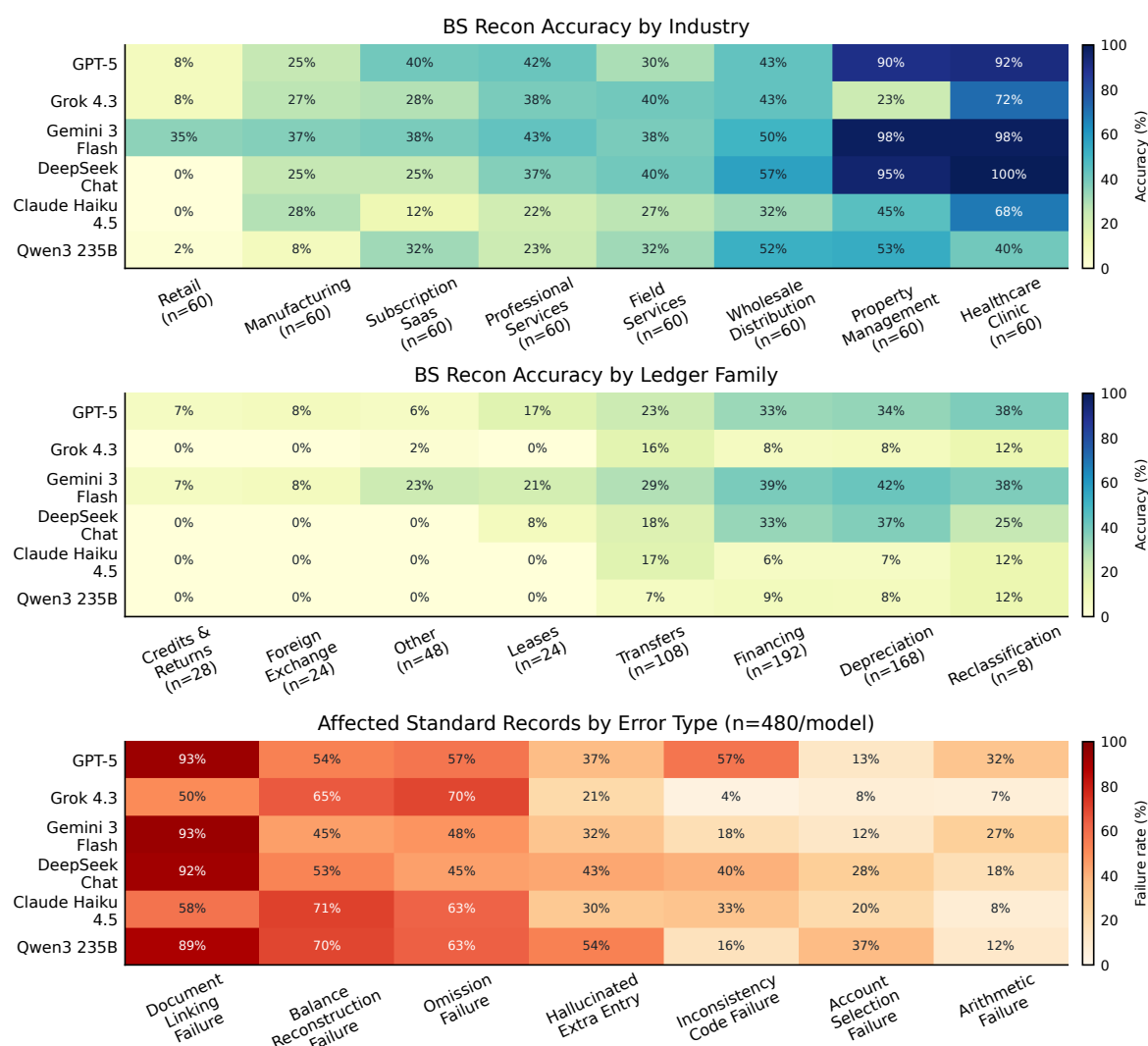


Figure 11: Failure slices across the six-model panel. Industry and ledger-family panels show BS_{recon} accuracy (higher is better), with the standard-record denominator in each column label. The error-taxonomy panel uses all 480 standard records per model (lower is better).

Instrument. Each of 75 release records was rendered as a self-contained Markdown file with seven sections: (1) record context (industry, period, currency, tax regime, difficulty, inconsistency flag), (2) the allowed-account-name list, (3) opening trial balance, (4) every document with its OCR text and role tag, (5) ground-truth journal entries, (6) ground-truth balance sheet, and (7) the verification form. The form contains six checks: document analogy, entry correctness, entry completeness, document-reference support, difficulty calibration, and overall verdict. Each check offers three options (positive, mixed, negative) plus a free-text notes field.

Protocol. The release subset contains 75 expert-reviewed records: 60 standard records and 15 forced-inconsistency records. The intended assign-

ment used in the paper is Reviewer 1 on records 1–50 and Reviewer 2 on records 1–25 plus 51–75. This gives every record one expert review and creates a 25-record independent overlap on records 1–25. Reviewers worked independently and did not see each other’s responses. The third reviewer (certified accountant) instead covered the entire compact coverage split of 143 records (120 standard + 23 forced-inconsistency). This split contains one record per industry–period–difficulty cell plus one per inconsistency code, so the review covers every such cell and code.

Results. Table 11 reports strict three-option agreement, collapsed acceptable agreement, and per-reviewer first-option rates on the 25-record double-reviewed overlap. The collapsed column

Axis	Variants evaluated	Diagnostic purpose
Prompt	baseline, guided private solve, self-check, strict document-citation pressure, explicit balance reconstruction	Separates baseline prompting, task-specific workflow, final verification, document-citation pressure, and aggregation instructions.
Visibility	full document bundle, OCR-only, no allowed accounts, no distractors, no support docs, evidence-only condition, evidence + 5/15/30 distractors, evidence relevant-last	Tests whether failures arise from missing evidence, account taxonomy, metadata, distractors, document position, context length, or the underlying accounting logic after evidence is controlled.
Tool / diagnostic	no tools, calculator, document search, ledger check, combined tool access, best-of-3 self-consistency, forced ledger-feedback tool, two-pass ledger-feedback tool	Tests prompt-described tool access, ledger-aware candidate selection, and forced exposure to the deterministic ledger as a consistency-checking tool.

Table 5: Ablation matrix. The matrix covers all single-axis variants and selected combined-condition diagnostics. This design preserves paired comparisons without enumerating the full prompt $\times$ visibility $\times$ tool cross-product.

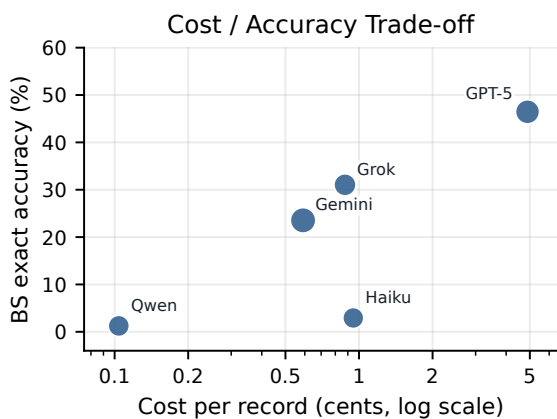


Figure 12: Cost/accuracy trade-off for completed baseline model runs. Marker area scales with BS_{recon} , so vertically low but large markers indicate models that write entries with plausible accounts and amounts but fail to aggregate them.

treats “Yes/Mostly” or “Accept/Accept with minor fixes” as the same acceptability judgment. Difficulty remains first option versus the other options. We report these descriptive rates because the sample is small and marginals are highly skewed toward positive judgments, making Cohen’s κ less informative. The overlap contains standard records only. Inconsistency validity was reviewed in the single-reviewed forced-inconsistency records.

Full-coverage review. The third reviewer accepted all 143 compact-coverage records. On the first response option, the reviewer chose “analogous” on 139/143 records, “entries correct” on 136/143, “complete and exact” on 143/143, “document references correct” on 135/143, and “calibration right” on 139/143. The overall verdict was “acceptable as ground truth” on 136/143 records

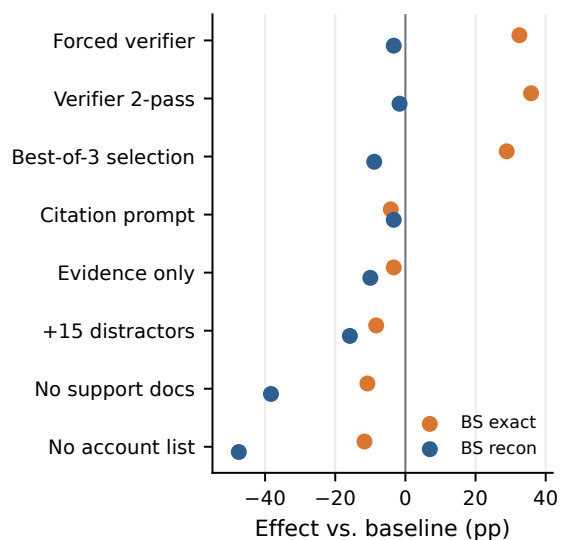


Figure 13: Gemini 3 Flash ablation deltas on the compact split ($n=143$ unless a condition uses the labeled targeted subset). Dots are paired-bootstrap mean deltas vs. the matched baseline. Visibility stresses dominate the negative tail.

and “acceptable with minor fixes” on the remaining 7. Collapsing acceptable judgments, document analogy, entry correctness, entry completeness, document-reference support, and the overall verdict are all at 100%. Difficulty calibration is 97% (3 “too easy,” 1 “too hard”). The minor-fix notes concern localized document-reference cleanups and document-label wording, while entry and balance-sheet values remained correct. This pass includes all 23 forced-inconsistency records. Every one was confirmed to contain a real contradiction that would block a consistent reconciliation, and on 3 of 23 the reviewer additionally noted that the assigned code was a loose fit for an otherwise genuine break.

Question	Main effect	Interpretation
Prompt	Guided/self-check prompts change BS_{exact} by at most 2.5 pp. Citation pressure changes ref mismatch by -1.4 pp.	Failures primarily reflect mechanisms beyond prompt compliance.
Evidence	Removing accounts and drops BS_{recon} by 47.5 pp. Removing support documents drops it by 38.3 pp.	Account taxonomy and support evidence are essential.
Grounding	Evidence-only visibility lowers ref mismatch by 26.4 pp.	Evidence selection drives citation failure.
Context	Adding 5/15/30 distractors to evidence-only visibility costs 6.7/5.8/8.3 pp BS_{recon} .	Longer context explains part of the task difficulty.
Repair	One ledger-feedback turn raises BS_{exact} by 32.5 pp on Gemini. Best-of-3 raises it by 28.9 pp on the targeted subset.	Global ledger checks re-pair many reported balance sheets.

Table 6: Compact-split ablation summary for Gemini 3 Flash ($n=143$, with balance-sheet metrics over $n=120$ standard records). Deltas are percentage points against the matched baseline unless a targeted subset is stated.

I Reproducibility

Prompt availability. All evaluations use ordinary text prompts without structured-output mode or a response-schema constraint. The exact baseline prompt, record-dependent suffix, output examples, forced-inconsistency example, and citation-pressure condition are released in `finbalance/benchmark/prompt.py` at <https://github.com/Devansh1105/finbalance>. The model-visible prompt excludes document role tags.

Seeds. Dataset generation uses seed 42. Headline LLM evaluations use temperature 0.0 with no explicit seed because closed APIs expose no deterministic seed control. The best-of-3 condition uses temperature 0.7 with three independent samples.

Sampling. The main evaluation split, stored under `data/main/`, uses four standard records per (industry, period type, difficulty level) cell and ten forced-inconsistency records per code, drawn from a stratified deterministic sampler. The compact ablation split, stored under `data/coverage/`, contains one record per cell plus one forced-

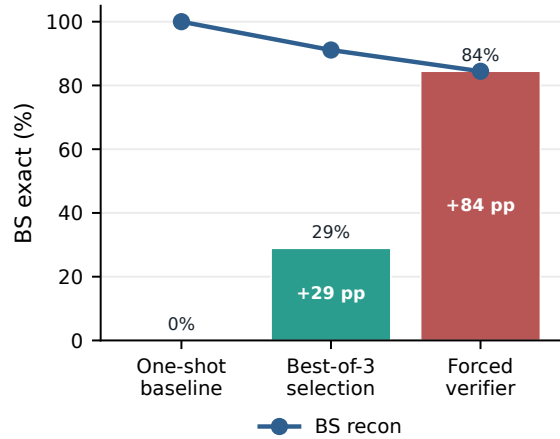


Figure 14: Gemini 3 Flash on the targeted aggregation-gap subset ($n=45$ standard records): best-of-3 selection recovers part of the gap, while one ledger-feedback revision recovers substantially more on the same records.

inconsistency record per code.

Table 12 records the requested and returned model identifiers, providers, and headline decoding settings.

API reproducibility. OpenRouter runs and DeepSeek Platform runs use chat-completions endpoints with ordinary text prompts and no structured JSON mode or response-schema constraint. The prompt and parser alone enforce strict JSON. Each run directory stores `evaluation.json`, `summary.json`, `per_record_results.jsonl`, and slice tables. The evaluation files include run start/end timestamps, requested model identifier, returned model identifier, selected provider where available, service tier where available, raw response text, raw response metadata where provider terms permit, token counts, cost where available, latency, parse status, tool-use and ledger-feedback metadata, and per-record metrics. Released prompt templates and record JSON reconstruct every prompt, and released raw model outputs permit offline recomputation of the reported scores. Transient gateway errors use at most three client retries with exponential backoff for 429/5xx responses. Parse failures remain failures and are logged. Commercial APIs may later change model snapshots or routing. Requested and returned model identifiers, provider metadata, and decoding settings identify what was measured. Repeated-run ranges are reported in Appendix G.

Bootstrap. All confidence intervals are paired bootstrap with $n=5000$ resamples drawn over a

Control configuration	n	BS _{exact}	BS _{recon}	gap	inconsistency-code match rate	parse
DeepSeek Reasoner, 32k	143	54.2%	55.8%	1.7 pp	56.5%	100.0%
Haiku 4.5 reasoning-low, 32k	143	38.3%	46.7%	8.4 pp	91.3%	100.0%
Qwen3 thinking, 32k	143	31.7%	40.0%	8.3 pp	52.2%	98.6%
Grok-4.3 reasoning disabled, 16k	143	3.3%	30.8%	27.5 pp	95.7%	98.6%
GPT-5 minimal effort, 16k	143	0.0%	24.2%	24.2 pp	73.9%	77.6%

Table 7: Reasoning controls on the compact split. inconsistency-code match rate is reported per configuration because reasoning changed contradiction detection in different directions. Strict end-to-end accuracy remains between 0 and 4.2% across these controls. The GPT-5 minimal run has 22.4% unparseable outputs, which are scored as failures.

Model	Text-tool BS _{exact}	Native BS _{exact}	Native BS _{recon}	used tools	strict	parse
Gemini 3 Flash	20.0%	52.5%	59.2%	141/143	5.8%	98.6%
DeepSeek Chat	8.3%	45.8%	54.2%	134/143	5.0%	81.1%
GPT-5	—	52.5%	54.2%	56/143	4.2%	99.3%
Haiku 4.5	20.0%	23.3%	37.5%	140/143	3.3%	98.6%

Table 8: Dual-protocol tool study on the compact split. “Used tools” counts records with at least one native call. Strict requires the reported balance sheet, entries, and citations to match. Native tools improve aggregation when invoked, while residual gaps and low strict accuracy remain. DeepSeek Chat’s lower parse rate reflects tool-calling interactions that exhausted the response budget before a final answer was completed. These cases remain scored as failures.

Char-edit rate	DeepSeek Chat	
	Gemini BS _{recon}	BS _{recon}
0% (uncorrupted)	61.7%	50.0%
0.94%	54.2%	46.7%
2.20%	53.3%	44.2%
4.45%	49.2%	36.7%

Table 9: Character-noise control on the compact split ($n=143$ per cell, with BS_{recon} over $n=120$ standard records). Parse success is 99–100% throughout. The control covers character corruption of templated text and excludes raw OCR and rendered layouts.

shared record-id set, so baseline and treatment are compared on matched records. CIs are reported as percentile 95%. These intervals cover record-level sampling uncertainty for each saved run and exclude API nondeterminism, routing changes, model updates, and future-run variance.

Release and licenses. The benchmark generator, evaluation and ablation code, bootstrap analysis tools, and tests are released under Apache-2.0. The release includes a Python interface and command-line tools for custom datasets. Users can request an exact record count or a balanced number of records per selected industry, period type, and difficulty cell. The repository README explains how to add industries, policies, document schemas, contradiction types, and versioned scoring. Generated benchmark records, OCR text, PDF-style

document artifacts, labels, sample manifests, and expert-validation records are released under CC BY 4.0, which permits commercial use with attribution. Synthetic company names, counterparties, and document templates are author-created and independent of real customer records or third-party proprietary templates. Raw model outputs and non-secret response metadata from paper runs are released as reproducibility artifacts where provider terms permit, separate from the dataset license. API keys, billing logs, and restricted provider fields are excluded or redacted. Users who rerun evaluations must comply with their model and API provider terms. The paper uses no hidden evaluation split.

Model	Expected+extras		
	among mismatches	exact strict	tolerant strict
Gemini 3 Flash	42.0%	5.8%	8.8%
GPT-5	58.8%	4.6%	15.8%
DeepSeek Chat	3.6%	2.5%	3.3%

Table 10: Citation sensitivity on the 480 standard core records. For Gemini and GPT-5, 96.2% and 94.0% of extra cited documents in superset/partial-overlap cases are support documents. Tolerant strict accuracy reaches 8.8%, 15.8%, and 3.3% for Gemini, GPT-5, and DeepSeek Chat, respectively.

Question	R1+	R2+	Strict	Collapsed
Q1 Doc. analogy	92%	72%	64%	100%
Q2 Entry correct	92%	96%	96%	100%
Q3 Entry complete	100%	100%	100%	100%
Q4 Document refs	92%	80%	72%	100%
Q5 Difficulty	100%	96%	96%	96%
Overall verdict	84%	96%	80%	100%

Table 11: Inter-annotator agreement on the 25-record overlap (records 1–25). “R1+” and “R2+” are first-option rates. “Strict” requires the exact same option. “Collapsed” merges the first two acceptable options for analogy, entries, document references, and overall verdict. Most disagreements concern severity calibration, with no reject-level split.

Paper name	Requested model	Returned model	Provider	Settings
Gemini 3 Flash	google/ gemini-3-flash-preview	google/ gemini-3-flash- preview-20251217	Google	temp 0, max 12k, no reasoning
GPT-5	openai/gpt-5	openai/ gpt-5-2025-08-07	OpenAI	temp 0, max 32k, reasoning effort low
Claude Haiku 4.5	anthropic/claude-haiku-4. 5	anthropic/ claude-4.5-haiku- 20251001	Bedrock	temp 0, max 16k, no reasoning
Grok-4.3	x-ai/grok-4.3	x-ai/ grok-4.3-20260430	xAI	temp 0, max 16k, reasoning
Qwen 3 235B	qwen/ qwen3-235b-a22b-2507	qwen/ qwen3-235b-a22b- 07-25	DeepInfra	temp 0, max 12k, no reasoning
DeepSeek Chat	deepseek-chat	deepseek-chat	DeepSeek	temp 0, max 8k, non-reasoning endpoint

Table 12: API-model settings for the six completed headline baseline runs. GPT-5 and Grok-4.3 used reasoning. Providers and returned identifiers come from stored API responses where available. The best-of-3 condition uses temperature 0.7 and three samples on the aggregation-gap subset.