FINANCIALAUDITBENCH: BENCHMARK CONSTRUCTION UNDER DIFFERENTIAL PRIVACY USING REAL-WORLD PRIORS

Anonymous authors
Paper under double-blind review

ABSTRACT

As AI agents are becoming widely adopted in the financial services industry, careful measurement is essential to understand where they can be reliably deployed and where oversight and professional review remain necessary. Such measurement, however, is constrained by limited access to proprietary or privacy-sensitive data. Existing benchmarks therefore often rely on publicly available data, human- and/or LLM-authored tasks, or simplified settings. We introduce FinancialAuditBench, a benchmark for evaluating agents on financial statement audit tasks, along with a framework for systematically generating synthetic engagements. Our task generation framework leverages differentially private aggregate statistics from historical audits along with audit expertise contributed through over 1,100 hours of benchmark development and review. FinancialAuditBench consists of 90 tasks spanning workpaper completion and review across six synthetic audit engagements, each containing an average of 179 files. Evaluation on eleven frontier models shows that while agents complete substantial portions of staff-level audit tasks well, they sometimes perform inappropriate procedures or produce incorrect documentation. Beyond financial auditing, our framework offers an approach for systematically generating synthetic tasks for model evaluation and training in privacy-sensitive domains.[1]

1 INTRODUCTION

As artificial intelligence capabilities have continued to improve, financial institutions are increasingly adopting generative AI in their workflows, with 71% of financial services firms having reported adoption in a recent survey (Cambridge Centre for Alternative Finance, 2026). Yet, while this domain is highly suitable for the strong information retrieval and reasoning capabilities that agents have shown to possess, publicly available benchmarks remain limited in number and complexity. As a consequence, researchers and financial experts lack the datasets required to accurately measure where autonomous agents can be deployed reliably and where professional oversight remains necessary.

Due to the privacy-sensitive and confidential nature of financial services, further progress in the evaluation and training of frontier models is increasingly constrained by the amount of accessible data (Villalobos et al., 2024; Muennighoff et al., 2023). Existing approaches to benchmark construction rely heavily on publicly available data (Jimenez et al., 2024; Islam et al., 2023), use large language models (LLMs) to propose and validate tasks (Gandhi & Neubig, 2026; Xie et al., 2026), and/or require many hours of work by domain experts (Wang et al., 2026; Grupen et al., 2026; Xu et al., 2026). To construct *realistic* financial tasks and environments, key challenges remain in leveraging privacy-sensitive data with appropriate privacy guarantees and developing scalable methods for task construction.

In this work, we focus on financial statement auditing[2], a domain that examines whether a company's reported financials are supported by sufficient evidence and are accurately reported within a set of

[1] Our dataset and repository are provided in the supplementary material.

[2] Key audit terminology is defined at the start of Section 3.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

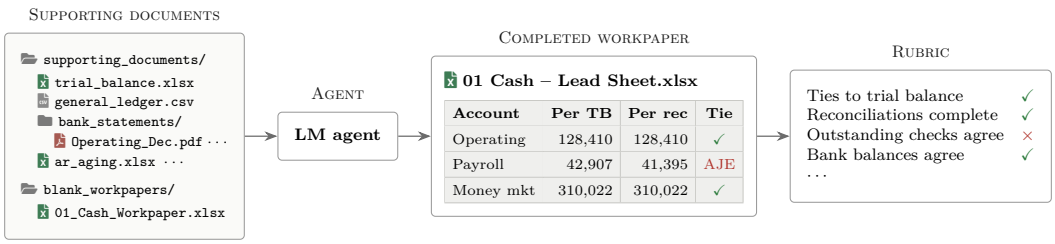


Figure 1: Task structure in FinancialAuditBench. The agent uses the provided supporting documents to perform the appropriate audit procedures in a workpaper template. The completed workpaper is then evaluated against a rubric.

tolerance bounds. Auditors identify areas where significant reporting errors could occur, obtain supporting evidence, and test the company's reported amounts against that evidence. For example, checking reported revenue may require tracing a recorded sale to a customer order, an invoice, and shipping confirmations. Thus, a data room used for evaluation or training requires a set of financial records whose amounts, dates, and relationships are internally consistent across documents.

To solve these aforementioned challenges, we develop a computational graph that generates synthetic companies and their corresponding financial documents. Auditors help define this graph's variables, dependencies, and accounting rules, while differentially private aggregate statistics from historical audits provide priors for a subset of selected variables. These aggregate statistics ground task construction in observed historical patterns with the goal of reducing the amount of domain expert feedback necessary to correct implausible initial assumptions otherwise generated from public sources or from LLMs. Differential privacy furthermore provides formal guarantees for releasing aggregate statistics from confidential data (Dwork & Roth, 2014), while simpler anonymization methods such as removing personally identifiable information (PII) have been shown to not necessarily prevent sensitive information from being inferred from the remaining context (Oh et al., 2026; Narayanan & Shmatikov, 2008). Section 4 describes the task generation pipeline, its construction process, and the privacy boundaries established during construction, while Section 5 presents the formal differential privacy guarantee.

Using this framework, we construct FinancialAuditBench (Figure 1), a benchmark of 90 workpaper-completion and review tasks that measures agents' accuracy and reliability on a set of commonly performed real-world audit tasks, required in a financial statement audit as performed by licensed certified public accountants (CPAs). Each task provides an agent with supporting documents from a synthetic engagement binder and a workpaper to complete or review. The submitted workpapers are then evaluated against rubrics designed by licensed auditors.

We evaluate eleven open- and closed-weight frontier models in a range of realistic settings commonly encountered in practice. While prior work has shown that LLMs perform well on CPA exam questions (Eulerich et al., 2024; Zacher & Kuppannagari, 2024), we find that agents still struggle to complete real-world audit tasks reliably without human supervision or assistance. More specifically, the best-performing frontier model achieves a single-attempt task pass rate of 69.17% (Pass@1), falling to 44.44% as measured by pass^8 (Yao et al., 2024). Thus, audit work completed by frontier models still requires review by domain experts and, where necessary, correction before it can be used in practice. Overall, we believe the quality, scale, and complexity of FinancialAuditBench make it a realistic and representative benchmark for evaluating agents' abilities to complete common financial auditing procedures.

Our work makes three primary contributions: (1) FinancialAuditBench, a realistic benchmark containing 90 tasks across six synthetic audit engagements, (2) a systematic framework for generating internally consistent synthetic audit engagements, and (3) an empirical evaluation of eleven frontier models across several experimental settings.

We release a development version of the task generation framework under the MIT license for use by the broader research community.

2 Related Work

Benchmarks for AI systems. Benchmarks for agent evaluation have progressed from information retrieval tasks to longer-horizon, real-world tasks. While initial benchmarks focused on simpler question-answering and general tool-use settings (Islam et al., 2023; Patil et al., 2025), more recent benchmarks have focused on knowledge work settings that more closely mirror work that professionals do (Lee et al., 2026). Within the domain of finance, APEX–Accounting (Benchek et al., 2026) evaluates accounting workflows such as recording transactions, reconciling accounts, and preparing reports. AuditFraudBench (Liu et al., 2026), on the other hand, evaluates audit judgment using company filings and regulatory evidence. FinancialAuditBench focuses on financial statement audit fieldwork, requiring agents to examine engagement evidence, perform specified audit procedures, and document their findings in workpapers.

Scalable task construction. Several prior works have proposed methods for systematic task generation rather than relying solely on manual task creation. In software engineering, SWE-smith (Yang et al., 2026) uses an automated pipeline to construct tasks and write tests to validate candidate tasks. In the domain of computer-use, AgentSynth (Xie et al., 2026) uses LLM-based task proposal, execution, and verification to construct long-horizon tasks, and Go-Browse (Gandhi & Neubig, 2026) uses LLMs to propose webpage-grounded tasks through structured exploration. In finance, FinMaster (Jiang et al., 2025) uses a financial simulator to generate transactions and financial statements, with auditing tasks focusing on identifying injected errors in transaction records.

Differential privacy. Prior work has developed differentially private generative models, including GANs (Yoon et al., 2019) and diffusion models (Dockhorn et al., 2023). For generating data, AIM (McKenna et al., 2024) generates synthetic tabular data from privately measured marginals, and Private Evolution (Lin et al., 2025b) and Sim-PE (Lin et al., 2025a) use differentially private nearest-neighbor counts to guide iterative generation with foundation models and simulators, respectively. Other approaches use LLMs to generate differentially private synthetic text (Amin et al., 2024) or generate differentially private examples for in-context learning (Gao et al., 2025).

3 FinancialAuditBench

Terminology. An *engagement* is the audit of a company for a specific reporting period, usually a fiscal year. *Prepared-by-client* (PBC) files or PBCs are records, financial documents, and ledgers supplied by the audited company or client. In FinancialAuditBench, a *workpaper* is a spreadsheet organized using a predefined template in which auditors document procedures performed, evidence examined, and conclusions reached. An engagement *binder* contains the materials for one synthetic audit engagement, including planning documents, client-provided records, third-party evidence, and blank workpaper templates. Agents use these materials to produce completed workpapers.

Overview. FinancialAuditBench is a benchmark dataset that evaluates how well modern AI systems complete and review real-world fieldwork tasks typically performed by CPAs during a financial statement audit. The benchmark includes financial documents for six synthetic companies spanning two industries, manufacturing and staffing services. Appendix B describes the types of files in an example engagement binder. Each synthetic company's binder includes either seven or eight workpapers, depending on industry, that can be independently completed, where a single task corresponds to completing or reviewing one workpaper. In aggregate, FinancialAuditBench comprises 45 main tasks that measure workpaper completion performance and 45 corresponding review tasks.

The tasks in FinancialAuditBench are designed to model work typically assigned to more junior (staff-level) auditors during the substantive fieldwork phase of a financial statement audit. Agents receive an audit plan and assessed risks as inputs, and the workpaper templates specify the procedures to perform. Completing these tasks requires identifying and evaluating relevant evidence, checking recorded amounts against supporting documents, performing calculations and reconciliations, and documenting supported conclusions. The synthetic client binders, moreover, are designed to provide much of the supporting documentation required to complete these procedures, while limiting common errors and ambiguities in the supporting information (although some remain as

discussed in Appendix A). Additional information on how these tasks relate to financial audits in practice can be found in the appendix as well.

Task format. As illustrated in Figure 1, each task provides an agent with a synthetic client's set of supporting documents and one workpaper template specifying the audit procedures to perform. Similar to how auditors complete these workpapers, the agent must examine the provided files and perform the specified audit procedures. We release all the tasks in the Harbor format (Harbor Framework Team, 2026), which packages task instructions, execution environments, and grading scripts.

Evaluation. Completed workpapers are graded against rubrics covering an auditor-guided sample of data points, calculations, and conclusions, using reference answers prepared and/or reviewed by CPA-licensed auditors. This grading approach is motivated by traditional workpaper review in audit firms, where reviewers exercise judgment about what to examine rather than systematically reperforming every procedure or checking every cell. The rubrics contain an average of ten scored checks per workpaper, and a workpaper receives a passing score when all applicable criteria are met; otherwise, it fails under a strict all-or-nothing grading metric. Importantly, a passing score under our proposed evaluation methodology **does not imply that a given workpaper is free of errors or omissions outside of the rubric's scope**. The percentage of criteria satisfied as a partial-credit measure of performance on the selected checks is also provided. Additional details on how evaluation and grading criteria are implemented can be found in Appendix D.

4 Benchmark Construction

4.1 Preliminaries

Privacy boundary. Before constructing the benchmark, we establish explicit rules governing access to privacy-sensitive historical audit data. Only Phase II, as outlined below, accesses confidential source binders. Later stages use the resulting differentially private statistics, public information, and general audit expertise. Auditors provide feedback on synthetic materials in environments that prohibit access to confidential source documents and are instructed to not reference any specific engagements they have worked on in the past.

Framework structure. An engagement binder contains over a hundred financial documents that describe snapshots of the same shared underlying business activities. Their amounts, dates, and relationships must therefore remain consistent across documents. To enforce these constraints, the framework represents each generated company as a computational graph whose nodes encode company characteristics and business events, and whose edges specify the relationships among them.

At a high level, the graph contains three types of nodes. Source nodes encode user-defined inputs or values sampled from predefined distributions, conditionally sampled nodes draw values from distributions determined by the values of their parent node(s), and deterministic nodes apply accounting rules or calculations. Figure 2 below illustrates these dependencies with a simple payroll example. Here, the chosen industry determines the distributions from which annual revenue and the payroll-to-revenue ratio are sampled. Multiplying the sampled revenue by the sampled ratio then determines annual payroll. This design represents dependencies across financial records and alleviates the need for estimating high-dimensional joint distributions.

4.2 Benchmark Scope

We first defined the benchmark's scope as engagement binders spanning the industries of *manufacturing* and *staffing services*. Following these decisions, we consulted fifteen auditors to identify the information used when auditing these industries including client-provided records, third-party evidence, auditor-developed materials, and public information. The final benchmark includes eight workpaper types, with supporting information identified through auditor feedback.

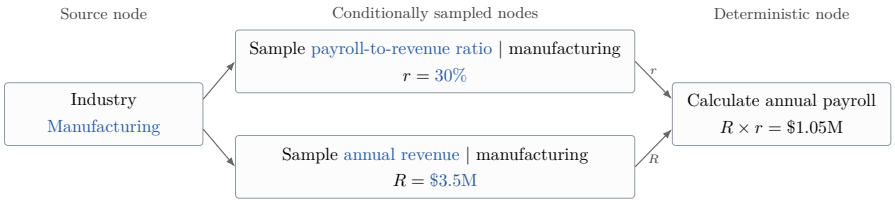


Figure 2: A toy example illustrating the three node types used in the task generation graph. The source node specifies the industry, which determines the distributions from which annual revenue and the payroll-to-revenue ratio are sampled. The deterministic node calculates annual payroll by multiplying the two sampled values.

4.3 Framework Construction

We construct the task generation framework in four phases. For each phase, we clearly outline where and how LLMs were used. Additional implementation details can be found in our released code.

Phase I: Variable selection. Using the workpaper types and supporting document categories identified during scoping, we create document drafts with input from domain experts and AI tools, refining their realism, accuracy, and presentation based on auditor feedback. At this stage, these drafts specify the content each document should contain but do not yet capture the relationships across documents needed to form a coherent engagement binder. An LLM then proposes candidate variables from these drafts, and three auditors independently assess each variable from its description and propose additions. A senior auditor, defined in this work as an auditor with over ten years of audit experience, consolidates their feedback, which an LLM then uses to rank the variables for subsequent stages.

Phase II: Private aggregation. An LLM extracts values for the selected variables from a corpus of real-world historical binders, containing both supporting documents and filled out workpapers. Each binder's contribution to each statistic is bounded, and the contributions are aggregated across binders using differentially private mechanisms. These statistics provide priors for graph construction and synthetic data generation. Section 5 describes the mechanisms and privacy guarantee. In initial construction experiments, we qualitatively found that these empirical priors provide a better basis for constructing the graph compared to those generated by an LLM or derived from publicly available data. In practice, these priors help reduce the corrective feedback required from auditors regarding implausible values and relationships in the generated documents during expert review (Phase IV). Furthermore, because the computational graph models intricate details of each synthetic company (e.g., the proportion of raw material wasted during production at a mid-sized manufacturing company), we found that auditors and financial experts alike could not always confidently assess whether these modeled values were realistic without being able to consult reference documents containing privacy-sensitive data.

Phase III: Graph construction. To scale the graph, we add edges and derived nodes to the computational graph using public resources on finance and auditing, auditor expertise, and LLMs. The fully constructed computational graph contains around 250 nodes and 1,000 edges. The generator samples values from the differentially private priors and propagates them through the computational graph to derive account counts, balances, financial events, etc. LLM-generated tests check the feasibility of the generated financials and optionally trigger resamples when infeasible company states are reached (e.g., month-end balances that are negative).

Phase IV: Expert review and revisions. In each round, the generator produces new binders, which auditors review for factual inconsistencies, implausible content, formatting issues, and missing documents. We use this feedback to revise the graph by resolving inconsistencies, modeling additional variables, and fixing presentation issues, then regenerate the affected documents for further review.

4.4 HUMAN ANNOTATION

Ground truth and rubrics. To construct the ground truth used in FinancialAuditBench, at least one CPA-licensed auditor completed and/or reviewed each workpaper. A senior auditor reviewed the proposed ground truth answers, adjudicated disagreements, and corrected errors. To develop the rubrics, a senior auditor constructed example rubrics for a set of core audit areas and then provided that example rubric structure to an AI model to replicate across the rest of the benchmark tasks. Subsequently, the rubrics were checked and revised by CPAs against multiple model submissions to assess their fairness and robustness.

Human annotation hours. Across graph construction, task generation, ground-truth preparation, and rubric refinement, auditors contributed a little over 1,100 hours. We do not report any inter-annotator agreement metrics (Cohen, 1960) because much of the feedback was developed collaboratively in discussions and collected across multiple disjoint revision cycles. For workpaper template construction, rubric construction and review, and workpaper completion, all final work products were reviewed by a senior auditor.

5 DIFFERENTIAL PRIVACY

Definition and privacy unit. The privacy unit is a single historical audit binder. Let D be a dataset of records from the set of historical audit binders we use in this project. We use binder-level adjacency, in which two datasets D and D' are adjacent if one can be obtained from the other by adding or removing all records from a single binder. A randomized mechanism M satisfies (ε, δ)-differential privacy if, for every pair of adjacent datasets D, D' and every measurable output set S (Dwork & Roth, 2014),

$$\Pr[M(D) \in S] \leq e^{\varepsilon} \Pr[M(D') \in S] + \delta.$$

Contribution bounds and mechanisms. Our privacy accounting covers six financial ratios, sixteen features describing financial records, and summary statistics for adjusting journal entries and accounts payable activity. Histograms and categorical distributions are estimated by summing bounded contributions from individual binders and applying the discrete Gaussian mechanism (Canonne et al., 2020) for ease of accounting under zero-concentrated differential privacy (zCDP). Other mechanisms such as discrete Laplace noise (Ghosh et al., 2009) could also be used for these queries. For each financial ratio, a fixed-grid variant of JointExp (Gillenwater et al., 2021) jointly estimates seven quantiles. Appendix C details the contribution bounds and privacy accounting, and our released code provides additional implementation detail.

Privacy guarantee. Before accessing any historical binders, we set a binder-level privacy ceiling after assessing with auditors the disclosure risks associated with the set of selected features as well as the theoretical upper bounds on binder membership inference. We did not empirically optimize this budget as similar data was not available in the public domain to be used for calibration.

Thus, composing the aforementioned mechanisms gives a total zCDP charge of $\rho \approx 0.2505$ (Bun & Steinke, 2016). Converting this budget at $\delta = 10^{-5}$ gives $\varepsilon \approx 3.647$, yielding a $(3.65, 10^{-5})$-DP guarantee for the addition or removal of one binder, below the predetermined ceiling. Subsequent graph construction and synthetic data generation use only these released statistics, public information, and general audit expertise, and therefore incur no additional privacy loss (Dwork & Roth, 2014).

6 EXPERIMENTS

We evaluate FinancialAuditBench across eleven frontier models on the 45 completion tasks, with eight independent runs per model-task pair. Thus, a total of 3,960 trajectories are generated and released for the main experiments in this work. We additionally evaluate ten models on the 45 review tasks, with one run per model-task pair, yielding an additional 450 review trajectories.

6.1 Models and Agent Environment

We evaluate all eleven models in a simple agent harness where the only tool provided to the agent is a *bash* tool, which executes shell commands. Table 8 lists the models, API names, reasoning effort, and temperature settings. Each trajectory is limited to 200 model calls. Additional details on the experimental setting are provided in Appendix D.

6.2 Main Results

Our main experiments reported in Table 1 evaluate *how well AI agents can autonomously complete common audit tasks end-to-end.* We report the task pass rates, partial-credit scores, cost, and runtime across eight independent runs per model-task pair. Strict pass rates measure whether all applicable rubric criteria are satisfied, while partial-credit scores indicate the proportion of criteria satisfied. Additionally, Figure 3 shows passˆk (Yao et al., 2024) and pass@k metrics to measure the reliability of these models across multiple trials. Tables 5 and 6 in Appendix D.3 report the underlying values.

Across all models, we find that while agents achieve partial-credit scores all greater than 85%, the best-performing models, Claude Opus 5.5 and Claude Fable 5.1, produce work that satisfies all applicable rubric criteria 69.17% and 65.83% of the time, respectively. Furthermore, the passˆk results demonstrate limited reliability of these agents across multiple runs with Claude Opus 5.5 and Claude Fable 5.1 passing all eight attempts on only 44.44% and 31.11% of tasks, respectively. Thus, these results suggest that deploying agents into financial audit workflows without professional oversight is ill-advised.

Table 1: Workpaper-completion performance across frontier models. Strict pass rates and partial-credit scores are averaged across tasks and runs. Cost and runtime are means per trajectory.

Model	Weights	Strict pass (%)	Partial credit (%)	Mean cost (USD)	Mean time (min)
Claude Opus 5.5	Closed	69.17	95.84	3.30	11.21
Claude Fable 5.1	Closed	65.83	94.81	6.49	13.03
Grok 4.6	Closed	47.22	91.12	1.62	11.99
GPT-5.6 Sol	Closed	44.17	89.41	0.91	3.78
GPT-6 Astra	Closed	43.61	90.95	2.96	8.00
GPT-5.6 Terra	Closed	40.28	88.66	0.40	2.06
Muse Spark 1.3	Closed	39.44	89.62	1.87	6.64
DeepSeek V4.1 Flash	Open	33.06	88.31	0.14	16.05
GLM-5.2	Closed	29.44	87.56	0.63	12.49
Kimi K3	Open	28.89	86.01	0.63	9.35
Gemini 3.8 Flash	Closed	27.50	87.88	1.80	18.52

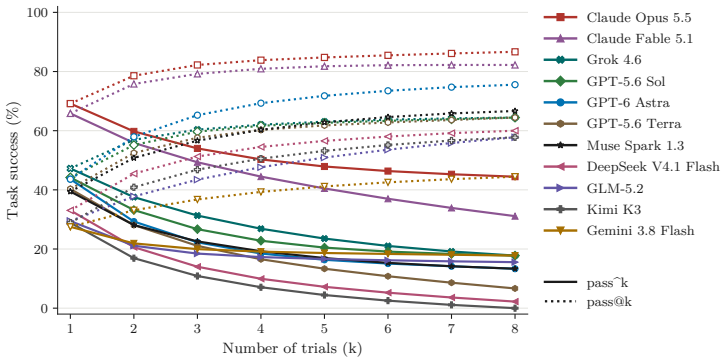


Figure 3: Reliability across eight independent runs. Solid lines show passˆk and dotted lines show pass@k. Both metrics use strict task success and are averaged across tasks. Exact values are provided in Appendix D.3.

6.3 Failure Analysis

To understand where and why models struggled and assess the fairness of the rubrics, auditors compared workpapers completed by agents with reference workpapers completed by auditors and reperformed a limited subset of procedures. This qualitative review identified four recurring types of errors: (1) failure to consider the effects of changes, (2) insufficient retrieval, (3) inadequate evaluation of evidence sources, and (4) reluctance to leave items open.

Failure to consider the effects of changes included cases where agents added information to the workpaper without assessing its downstream impacts (e.g., adding numeric details to a table when they should not have been included in the corresponding column total). Insufficient retrieval included cases where agents analyzed only the most obvious source document and concluded, based on that document alone, that the required details were not available (e.g., reviewing a custom shipping cut-off schedule that lacked the necessary detail, while the full shipping log contained it). Inadequate evaluation of evidence sources included cases where agents vouched amounts to client-provided documents rather than third-party evidence (e.g., vouching receivables to a client-prepared cash receipts report instead of a bank statement). Finally, reluctance to leave items open included cases where agents documented work using whatever information was available, even when the information the procedure required was missing, rather than noting that the procedure could not be performed as designed (e.g., documenting work over current-year transactions for a subsequent events procedure when the subsequent-period information had not been provided). These categories were not mutually exclusive.

6.4 Comparison with Estimated Auditor Completion Times

Across workpaper areas and all evaluated models, median agent runtimes range from seven to twelve minutes. In comparison, the estimated manual completion times of auditors used in this project ranged between two to twelve hours, when provided the same tasks. These estimates *exclude* senior review and corrective rework and therefore do not establish equivalent time savings in practice. However, they suggest potential time savings when agents are deployed within a well-defined scope and under appropriate professional oversight. Appendix D.1 provides a more in-depth comparison and outline of our time estimation methodology.

6.5 Review Tasks

FinancialAuditBench includes 45 review tasks evaluated across ten models, with all models receiving the same starting workpaper and supporting files for each task. In this experimental setting, models are instructed to identify and correct any potential errors that may be present in a filled out workpaper, while preserving correct work. Across the board, all models improve average strict-pass and partial-credit scores, although some individual workpapers received lower scores after review and correction. Claude Opus 5.5 achieves the largest improvement, increasing strict pass from 28.89% to 62.22% and partial credit from 87.34% to 94.57%. These results suggest that agent review can improve workpaper quality but does not reliably resolve all errors captured by the rubrics. Full results and the review setup are provided in Appendix D.2.

6.6 Computer-Use Setting

Audit workflows often involve enterprise applications, settings where agents cannot directly interact with applications programmatically and must instead interact through a graphical interface. To explore this setting, we evaluate ChatGPT Computer-Use (OpenAI, 2026) on eight completion tasks from one synthetic binder. The ChatGPT desktop application is granted computer access and instructed to complete all eight workpapers primarily via mouse and keyboard inputs. ChatGPT Computer-Use achieves 92.84% average partial-credit and a 50.00% strict pass rate, compared with 93.58% and 54.69%, respectively, for a simple agent equipped with a single bash tool on the same tasks. Both experimental settings used GPT-6 Astra. Appendix D provides the computer-use instructions and additional observations.

7 DISCUSSION

Implications. Our results show that while frontier models can complete substantial portions of audit workpapers, they do not yet produce work that can be used without professional review. Given the scope and granularity of our rubrics, the difference between partial-credit and all-or-nothing scoring highlights that models often satisfy many rubric criteria while failing others. Broader and more granular rubric coverage may reveal additional errors that are not reflected in the current scores. In financial auditing, such errors can have substantial real-world consequences if not caught. At the same time, agents help complete these tasks considerably faster than auditors, suggesting a role for AI agents as co-pilots that help auditors perform audit procedures and document their findings under professional oversight.

The value of such a co-pilot role depends not only on the accuracy of agent work, but also on the ease with which auditors can review it. We observed that agents tended to over-document how they arrived at an answer, leaving what the evidence indicated and how it affected the conclusion less clearly stated. Agents also favored writing in sentence fragments and formatted cell content poorly, which increased the effort required for a reviewer to understand what was written. Because the ability to easily understand the procedures performed, the evidence obtained, and the conclusions reached is a core element of quality audit documentation, these tendencies can detract from the overall quality of the work. As a result, such issues may reduce some of the time savings described above. Future evaluations should assess the clarity of agent documentation alongside its correctness.

Limitations and future work. FinancialAuditBench captures only part of the financial audit process. The current benchmark covers a limited number of industries, companies, workpaper types, and audit methodologies, and its synthetic engagements may not represent the full complexity of actual client work. Future work should expand the benchmark to include additional industries and audit procedures, and seek to model a wider range of realistic conditions, including incomplete, inconsistent, and poorly organized client data. Furthermore, the process of interacting with clients over the course of a financial audit is inherently a multi-turn process with auditors requesting information from multiple counterparties over the course of weeks. Extending this benchmark to support multi-turn conversational task settings would also increase the realism of the benchmark.

Our rubrics evaluate selected aspects of each workpaper and may miss errors or omissions that occur elsewhere. We emphasize that the reported metrics in this work are sufficient to reveal existing limitations in frontier model capabilities, and we *do not* make any claims regarding their readiness for autonomous deployment in real audit engagements beyond the evaluation scope we performed. Systematic and manual construction of more granular rubrics and evaluation methods extends beyond the scope of this paper and remains exciting directions for future work.

Evaluating the realism and fidelity of the generated synthetic binders also remains an open challenge, requiring substantial expert feedback. Despite extensive auditor review, resource constraints and the scale of the generated binders limited our ability to identify and resolve every issue affecting task accuracy and realism. To address these limitations, we restrict evaluation to workpaper sections that have well-defined ground truth, focusing on procedures that are common across financial statement audits in multiple industries.

Conclusion. We introduce FinancialAuditBench, a benchmark for evaluating AI agents on financial audit tasks, together with a framework for generating synthetic engagements from differentially private aggregate statistics and auditor expertise. Our results show that while agents perform well on many evaluated aspects of audit workpapers, they still do not yet consistently produce workpapers that are sufficient to support an audit conclusion without corrective feedback. These findings suggest that agents can assist auditors with workpaper preparation and review, while professional oversight remains necessary. More broadly, our construction framework shows how realistic tasks in privacy-sensitive domains can be generated for model training and evaluation.

AI USE STATEMENT

We used generative AI tools spanning AI coding agents and chatbots throughout the development of this project. In early stages of the work, authors of this work used AI tools to learn and better un-

derstand the financial audit domain and to more effectively communicate with domain experts. The use of AI tools during the construction of the benchmark itself and its corresponding task generation framework is clearly stated in Section 4. AI tools were also used for creating research figures and tables and for writing parts of the experimental code. The authors inspected and tested AI-assisted code and verified the quality and accuracy of the generated code. We take full responsibility for the final content of this work.

ETHICS STATEMENT

The benchmark contains synthetic audit records. Historical client data was used with authorization during the construction of this benchmark. The aggregate statistics released are protected by differential privacy bounds as described in Section 5. FinancialAuditBench is intended strictly for research evaluation and should not be used to provide audit assurance or replace professional judgment.

REPRODUCIBILITY STATEMENT

We release the tasks in our proposed benchmark including the data rooms, workpaper templates, rubrics, and relevant code necessary to run the core experiments in this work. For the ChatGPT Computer-Use setting, we provide sufficient detail for other researchers to reproduce our reported results. For the task generation framework, we release a development version of the framework that maintains the final structure of the computational graph used to generate the tasks presented in this paper, but omit substantial domain-specific rules and logic for generating more realistic and complex financial documents. This choice was made so that the code released to the public would be more easily understandable. We publish the query definitions, privacy ledger, and DP-derived priors used by the public generator.

REFERENCES

Kareem Amin, Alex Bie, Weiwei Kong, Alexey Kurakin, Natalia Ponomareva, Umar Syed, Andreas Terzis, and Sergei Vassilvitskii. Private prediction for large-scale synthetic text generation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7244–7262, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.425. URL https://aclanthology.org/2024.findings-emnlp.425/.

Julien Benchek, Austin Bennett, Jasmin Kern, Ryan Stevens, Rene Sultan, Charis Ching, Hayley Popiel, Vaibhav Mittal, Felix Mercier, Brendan Foody, and Bertie Vidgen. Apex-accounting, 2026. URL https://arxiv.org/abs/2607.27189.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography*, 2016. URL https://arxiv.org/abs/1605.02065.

Cambridge Centre for Alternative Finance. The 2026 global ai in financial services report: Adoption, impact and risks, 2026. URL https://www.jbs.cam.ac.uk/wp-content/uploads/2026/04/ccaf-2026-04-28-global-ai-in-financial-services-report.pdf.

Clément L. Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2004.00010.

Mark Cesar and Ryan Rogers. Bounding, concentrating, and truncating: Unifying privacy loss composition for data analytics. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, pp. 421–457. PMLR, 2021. URL https://proceedings.mlr.press/v132/cesar21a.html.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. doi: 10.1177/001316446002000104.

Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. Differentially private diffusion models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=ZPpQk7FJXF.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014. doi: 10.1561/0400000042. URL https://doi.org/10.1561/0400000042.

Marc Eulerich, Aida Sanatizadeh, Hamid Vakilzadeh, and David A. Wood. Is it all hype? ChatGPT's performance and disruptive potential in the accounting and auditing industries. *Review of Accounting Studies*, 29:2318–2349, 2024. doi: 10.1007/s11142-024-09833-9. URL https://doi.org/10.1007/s11142-024-09833-9.

Apurva Gandhi and Graham Neubig. Go-browse: Training web agents with structured exploration. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IpzRWE52yw.

Fengyu Gao, Ruida Zhou, Tianhao Wang, Cong Shen, and Jing Yang. Data-adaptive differentially private prompt synthesis for in-context learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sVNfWhtaJC.

Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms, 2009. URL https://arxiv.org/abs/0811.2841.

Jennifer Gillenwater, Matthew Joseph, and Alex Kulesza. Differentially private quantiles. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. URL https://arxiv.org/abs/2102.08244.

Niko Grupen, Gabe Pereyra, and Julio Pereyra. Open-sourcing harvey's long horizon legal agent benchmark, 2026. URL https://www.harvey.ai/blog/introducing-harveys-legal-agent-benchmark.

Harbor Framework Team. Harbor: A framework for evaluating and optimizing agents and models in container environments, 2026. URL https://doi.org/10.5281/zenodo.20953922.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. Financebench: A new benchmark for financial question answering, 2023. URL https://arxiv.org/abs/2311.11944.

Junzhe Jiang, Chang Yang, Aixin Cui, Sihan Jin, Ruiyu Wang, Bo Li, Xiao Huang, Dongning Sun, and Xinrun Wang. Finmaster: A holistic benchmark for mastering full-pipeline financial workflows with llms, 2025. URL https://arxiv.org/abs/2505.13533.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Jinu Lee, Kyoung-Woon On, Sophia Simeng Han, Arman Cohan, and Julia Hockenmaier. Evaluating legal reasoning traces with legal issue tree rubrics. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3299–3322, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.150. URL https://aclanthology.org/2026.acl-long.150/.

Zinan Lin, Tadas Baltrusaitis, Wenyu Wang, and Sergey Yekhanin. Differentially private synthetic data via apis 3: Using simulators instead of foundation model, 2025a. URL https://arxiv.org/abs/2502.05505.

Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, Harsha Nori, and Sergey Yekhanin. Differentially private synthetic data via foundation model apis 1: Images, 2025b. URL https://arxiv.org/abs/2305.15560.

Zhiwei Liu, Yueru He, Qing Ou, Tianlei Zhu, Xiaorui Guo, Xueqing Peng, and Sophia Ananiadou. Auditfraudbench: Benchmarking audit judgment in detecting fraudulent misstatements, 2026. URL https://arxiv.org/abs/2606.08345.

Zeyao Ma, Bohan Zhang, Jing Zhang, Jifan Yu, Xiaokang Zhang, Xiaohan Zhang, Sijia Luo, Xi Wang, and Jie Tang. Spreadsheetbench: Towards challenging real world spreadsheet manipulation, 2024. URL https://arxiv.org/abs/2406.14991.

Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. Aim: An adaptive and iterative mechanism for differentially private synthetic data, 2024. URL https://arxiv.org/abs/2201.12677.

Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=j5BuTrEj35.

Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pp. 111–125, 2008. doi: 10.1109/SP.2008.33.

Myeong Seok Oh, Dong-Yun Kim, Hanseok Oh, Chaean Kang, Joeun Kang, Xiaonan Wang, Hyunjung Park, Young Cheol Jung, and Hansaem Kim. Subject-level inference for realistic text anonymization evaluation. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17100–17135, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.778. URL https://aclanthology.org/2026.acl-long.778/.

OpenAI. Computer use. https://learn.chatgpt.com/docs/computer-use, 2026.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=2GmDdhBdDk.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Position: Will we run out of data? Limits of LLM scaling based on human-generated data. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 49523–49544. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/villalobos24a.html.

Alex Wang, Georg Meinhardt, Jacob Katz, Joseph H. Kim, Pratyush K. Chaudhary, Chase Blagden, and Eric Xu. Bigfinancebench: A workflow-grounded benchmark for financial-research agents, 2026. URL https://arxiv.org/abs/2606.03829.

Jingxu Xie, Dylan Xu, Xuandong Zhao, and Dawn Song. Agentsynth: Scalable task generation for generalist computer-use agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=CoBxmXThM6.

Frank F. Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Zhiruo Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, Mingyang Yang, Hao Yang Lu, Amaad Martin, Zhe Su, Leander Melroy Maben, Raj Mehta, Wayne Chi, Lawrence Keunho Jang, Yiqing Xie, Shuyan Zhou, and Graham Neubig. Theagentcompany: Benchmarking LLM agents on consequential real world tasks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=LZnKNApvhG.

John Yang, Kilian Lieret, Carlos E Jimenez, Alexander Wettig, Kabir Khandpur, Yanzhe Zhang, Binyuan Hui, Ofir Press, Ludwig Schmidt, and Diyi Yang. SWE-smith: Scaling data for software

engineering agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=63iVrXc8cC.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL https://arxiv.org/abs/2406.12045.

Thomson Yen, Julian Poeltl, Harshith Srinivas Gear, Yilin Meng, Joshua Fan, Adam Shen, Yili Liu, Ali Bauyrzhan, Patrick Shea, Siri Du, Haoyang Liu, Daniel Guetta, and Hongseok Namkoong. Mbabench: Evaluating llm agents on end-to-end spreadsheet tasks in finance, 2026. URL https://arxiv.org/abs/2605.22664.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1zk9iRqF7.

Will Zacher and Sanmukh Kuppannagari. Can llms pass the cpa exam? evaluating large language model performance on the certified public accountant test, 2024. URL https://ssrn.com/abstract=4788096.

A Audit Task Design, Scope, and Disclosures

A.1 Task Design from an Auditor's Perspective

The overall process of a financial statement audit spans the general phases of planning, risk assessment, substantive fieldwork, concluding, and reporting, with significant time and effort spent on substantive fieldwork in most audits. Audit evidence gathered during this phase serves as the foundation for the audit opinion and report. The specific class of tasks contained in FinancialAuditBench targets work done during the substantive fieldwork phase, where auditors respond to assessed risks and gather evidence through the performance of audit procedures that produce audit documentation typically in the form of workpapers. Because the planning and risk assessment phases determine the nature, timing, and extent of audit procedures and generally precede substantive fieldwork, the agent is provided with determinations made earlier in the audit, including the assessed risks of material misstatement and the audit plan (i.e., the set of chosen procedures to complete), along with necessary information on the entity and its environment. Given these facts, we assess how accurately an AI system can evaluate the provided evidence, complete the assigned procedures, and reach a well-documented conclusion.

A.2 Client-Provided Records and Third-Party Evidence

In real-world settings, files required for a financial audit are provided either directly by the client or by third-party sources such as banks and the client's customers. These files may include extraneous documents, contain errors, and/or omit essential information. Thus, an auditor must identify the necessary files, request them from the relevant parties, and follow up to resolve errors, omissions, and inconsistencies. Each task in FinancialAuditBench provides the agent with a data room of evidence files and does not permit requests for additional information from clients or third parties. Importantly, as in real-world audits, the provided engagement binders contain imperfections, such as inappropriate account coding and inconsistencies between synthetic documents. Our rubrics do not evaluate parts of each workpaper affected by missing or imperfect information so that agents are not penalized for those limitations.

A.3 Relationship to Production Audit Practice

The blank workpaper templates, outlined procedures, documentation conventions, and grading criteria we release in FinancialAuditBench were developed with the intent of evaluating how well AI systems perform on real-world audit tasks. These artifacts are not copies of proprietary production templates and do not use the exact methodology of any accounting firm. Each firm organizes its

workpapers differently and uses its own respective testing procedures, review processes, documentation processes, etc. that comply with applicable generally accepted auditing standards. We believe that measured performance on FinancialAuditBench correlates with the performance in production audit settings.

B Example Generated Binder Structure

Figure 4 shows a pruned directory tree for an example of a synthetic client binders presented in FinancialAuditBench. At a high level, the planning folder provides assumptions and scope details for the planned audit, the PBC package contains documents prepared by and provided by the client, the direct-to-auditor folder contains third-party evidence such as bank confirmations, and the workpaper templates define the audit procedures that must be performed for a given task.

```
manufacturing-binder/
|-- planning/
|   |-- Engagement Scope and Case Facts.pdf
|   |-- Materiality Memo.pdf
|   |-- Assertion-Level Risk Assessment.pdf
|   |-- Audit Selection Plan.pdf
|   `-- Planned Procedure Register.pdf
|-- pbc_package/
|   |-- Package Index.xlsx
|   |-- foundational/
|   |   |-- Chart of Accounts.xlsx
|   |   |-- TB 12 31 24.xlsx
|   |   `-- TB 12 31 25.xlsx
|   |-- accounts_receivable/
|   |   |-- Detailed AR aging.xlsx
|   |   `-- Subsequent cash receipts.xlsx
|   |-- ap_and_accruals/
|   |   |-- AP Aging (2025-12-31).xlsx
|   |   `-- Vendor Invoice Support 2025.xlsx
|   |-- inventory/
|   |   |-- Detailed inventory listing or perpetual records.xlsx
|   |   `-- Physical count instructions and results.xlsx
|   |-- revenue/
|   |   |-- Sales invoices and customer support.xlsx
|   |   `-- Shipment delivery or acceptance evidence.xlsx
|   `-- documents/
|       |-- Complete Journal Entry Population.xlsx
|       |-- Payroll Register 2025.pdf
|       `-- Audited Financial Statements 12.31.2024.pdf
|-- direct_to_auditor/
|   |-- Bank Confirmation 12.31.2025.pdf
|   |-- Loan Confirmation 2025.pdf
|   `-- AR Confirmation - [Synthetic Customer].pdf
`-- workpaper_templates/
    |-- cash/cash_workpaper.xlsx
    |-- inventory/inventory_costing_valuation_workpaper.xlsx
    |-- payroll/payroll_workpaper.xlsx
    `-- revenue/revenue_workpaper.xlsx
```

Figure 4: A snapshot of an example data room provided to agents and auditors for filling out workpaper templates.

C Privacy Accounting

Each historical binder contributes at most one vote per histogram or categorical query, giving ℓ_2-sensitivity at most one. For each structured summary, the proportions of extracted records in different categories are encoded as a list of weights whose sum is at most 60. For example, if a binder's adjusting journal entries comprise 50% depreciation adjustments, 25% amortization adjustments, and 25% tax adjustments, the corresponding weights would be 30, 15, and 15, with zero weight for all other adjustment types. These weights represent category proportions scaled to 60. Bounding their sum caps the ℓ_2-sensitivity of each structured summary query to 60.

Using standard mechanism bounds (Canonne et al., 2020; Cesar & Rogers, 2021), the 36 mechanisms in Table 2 add up to a total zCDP charge of $\rho_{\text{total}} \approx 0.2505$. Then, standard zCDP-to-DP

conversion at $\delta = 10^{-5}$ yields a $(3.65, 10^{-5})$-DP guarantee (Bun & Steinke, 2016). The public repository includes the query definitions, mechanism parameters, and privacy charges.

Table 2: Privacy accounting by query family. DG denotes the discrete Gaussian mechanism.

Query family	DG $\Delta_2 \leq 1$	DG $\Delta_2 \leq 60$	JointExp	ρ
Financial ratios	6	0	6	0.049 44
Record features	16	3	0	0.149 03
AJE and AP summaries	2	3	0	0.052 03
Total	24	6	6	0.250 51

D Additional Results and Construction Details

D.1 Comparison with Estimated Auditor Completion Times

Table 3 compares auditor time estimates with agent performance across eleven models. For each workpaper area, we report the average pass rate and median agent runtime across all models and runs in the main experiments with runtime rounded to the nearest minute. *Auditor completion times are approximate estimates informed by the time auditors spent completing the benchmark workpapers without the use of AI or automation tools.* As mentioned before, based on the definition and structure of each of these tasks, these estimates exclude waiting for client responses and external confirmations as well as senior review and any resulting rework that often arise in actual financial audits. However, auditors' reviews suggested that workpapers with lower partial-credit scores generally would require more corrective feedback and time to fix. Resource constraints limited the number of auditor-completed workpapers, and participating auditors varied in experience. Thus, we do not report an aggregate human accuracy estimate.

Table 3: Estimated manual completion times (without the use of any AI or automation tools), measured agent runtimes, and agent pass rates for the benchmark workpapers. Agent runtimes are medians rounded to the nearest minute.

Workpaper	Estimated manual workpaper time	Median agent time (min)	Average agent strict pass (%)
Accounts payable	4–6 hr	9	68.18
Accounts receivable	6–10 hr	12	4.17
Cash	3–5 hr	8	56.06
Debt	3–5 hr	8	52.08
Fixed assets	2–4 hr	8	54.17
Journal-entry completeness	2–4 hr	7	94.51
Journal-entry risk testing	6–10 hr	12	16.86
Revenue	8–12 hr	10	7.58

D.2 Review Tasks

Financial audits follow a review process in which senior auditors review work prepared by less experienced staff. The review task setting instructs an agent to review a completed workpaper using the same supporting materials and to fix any errors that may exist. In this review setting, for each task, a completed workpaper from a randomly chosen model in the main experiments serves as the starting workpaper for every reviewing model. The results in Table 4 show that all models improve average strict-pass and partial-credit scores, although some individual workpapers receive lower scores after review. Each model is run once per review task.

Table 4: Review-task results. Strict pass / partial-credit scores before and after workpaper review; changes are in percentage points. Scores before review are identical because all models receive the same starting workpapers.

Model	Before review (%)	After review (%)	Change (pp)
Claude Opus 5.5	28.89 / 87.34	62.22 / 94.57	+33.33 / +7.23
Grok 4.6	28.89 / 87.34	46.67 / 91.07	+17.78 / +3.73
GPT-5.6 Terra	28.89 / 87.34	42.22 / 89.90	+13.33 / +2.56
Muse Spark 1.3	28.89 / 87.34	42.22 / 90.48	+13.33 / +3.14
DeepSeek V4.1 Flash	28.89 / 87.34	37.78 / 89.59	+8.89 / +2.25
GLM-5.2	28.89 / 87.34	37.78 / 89.74	+8.89 / +2.39
GPT-5.6 Sol	28.89 / 87.34	35.56 / 88.69	+6.67 / +1.35
Gemini 3.8 Flash	28.89 / 87.34	33.33 / 88.67	+4.44 / +1.33
GPT-6 Astra	28.89 / 87.34	33.33 / 90.37	+4.44 / +3.03
Kimi K3	28.89 / 87.34	33.33 / 89.36	+4.44 / +2.02

D.3 Reliability across repeated runs

Tables 5 and 6 give the values plotted in Figure 3. Passˆk measures success on all k attempts, while pass@k measures at least one success within k attempts.

Table 5: Passˆk across eight independent runs: the probability that all k attempts strictly pass. Values are fractions.

Model	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8
Claude Opus 5.5	0.69	0.60	0.54	0.50	0.48	0.46	0.45	0.44
Claude Fable 5.1	0.66	0.56	0.49	0.44	0.40	0.37	0.34	0.31
Grok 4.6	0.47	0.38	0.31	0.27	0.24	0.21	0.19	0.18
GPT-5.6 Sol	0.44	0.33	0.27	0.23	0.20	0.19	0.18	0.18
GPT-6 Astra	0.44	0.29	0.22	0.18	0.16	0.15	0.14	0.13
GPT-5.6 Terra	0.40	0.28	0.21	0.17	0.13	0.11	0.09	0.07
Muse Spark 1.3	0.39	0.28	0.23	0.19	0.17	0.15	0.14	0.13
DeepSeek V4.1 Flash	0.33	0.21	0.14	0.10	0.07	0.05	0.04	0.02
GLM-5.2	0.29	0.21	0.18	0.17	0.17	0.16	0.16	0.16
Kimi K3	0.29	0.17	0.11	0.07	0.04	0.03	0.01	0.00
Gemini 3.8 Flash	0.28	0.22	0.20	0.19	0.19	0.18	0.18	0.18

Table 6: Pass@k across eight independent runs: the probability of at least one strict pass within k attempts.

Model	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8
Claude Opus 5.5	0.69	0.79	0.82	0.84	0.85	0.85	0.86	0.87
Claude Fable 5.1	0.66	0.76	0.79	0.81	0.82	0.82	0.82	0.82
Grok 4.6	0.47	0.57	0.60	0.62	0.63	0.64	0.64	0.64
GPT-5.6 Sol	0.44	0.55	0.60	0.62	0.63	0.63	0.64	0.64
GPT-6 Astra	0.44	0.58	0.65	0.69	0.72	0.73	0.75	0.76
GPT-5.6 Terra	0.40	0.52	0.58	0.60	0.62	0.63	0.64	0.64
Muse Spark 1.3	0.39	0.51	0.57	0.60	0.63	0.65	0.66	0.67
DeepSeek V4.1 Flash	0.33	0.45	0.51	0.55	0.57	0.58	0.59	0.60
GLM-5.2	0.29	0.38	0.43	0.48	0.51	0.54	0.56	0.58
Kimi K3	0.29	0.41	0.47	0.51	0.53	0.55	0.57	0.58
Gemini 3.8 Flash	0.28	0.33	0.37	0.39	0.41	0.43	0.44	0.44

D.4 Effect of reasoning effort

We evaluate GPT-6 Astra at five reasoning effort levels with eight independent runs per task at each reasoning effort level. Table 7 reports strict pass rates, partial-credit scores, mean runtime, and mean cost per trajectory across the five reasoning-effort levels.

Table 7: Reasoning effort ablation with GPT-6 Astra. Runtime and model cost are means per trajectory, with eight independent runs per task.

Reasoning effort	Strict pass (%)	Partial credit (%)	Mean runtime (min)	Mean cost ($)
Low	23.61	85.71	3.08	1.32
Medium	43.61	90.95	8.00	2.96
High	55.00	92.71	9.59	3.97
Xhigh	59.44	94.16	10.95	4.49
Max	71.11	95.83	24.94	8.27

D.5 Experimental and grading details

Model configurations. Table 8 specifies the settings used in the main experiments of this work. Reasoning effort is configured for each model and set to the default or recommended level reported in its corresponding model card. For Fable 5.1, we set the reasoning effort to medium to be more in line with other models we evaluate. We used the default API endpoints for all OpenAI and Anthropic models and OpenRouter for the remaining models. The default caching policy is used for all models. Other parameters not explicitly outlined here retain their default values.

Table 8: Models and inference settings for the main and review experiments. Note that Claude Fable 5.1 is evaluated only in the main experiments.

Model	API	Reasoning effort	Temperature
Claude Opus 5.5	`claude-opus-5-5`	medium	–
Claude Fable 5.1	`claude-fable-5-1`	medium	–
Grok 4.6	`x-ai/grok-4.6`	medium	–
GPT-5.6 Sol	`gpt-5.6-sol`	medium	–
GPT-6 Astra	`gpt-6-astra`	medium	–
GPT-5.6 Terra	`gpt-5.6-terra`	medium	–
Muse Spark 1.3	`meta/muse-spark-1.3`	medium	–
DeepSeek V4.1 Flash	`deepseek/deepseek-v4.1-flash`	medium	–
GLM-5.2	`z-ai/glm-5.2`	high	–
Kimi K3	`moonshotai/kimi-k3`	medium	1.0
Gemini 3.8 Flash	`google/gemini-3.8-flash`	medium	–

Experimental setting. In the main experimental setting, each model-task pair is run independently eight times. Every run receives the same system prompt, task instructions, client file room, and blank workpaper template in an isolated Docker workspace. The agent can interact with the workspace only through a *bash* tool. A run ends when the agent returns the required completion signal or reaches 200 model calls. Individual model requests have a 600-second timeout and up to two retries. The agent harness does not perform context compaction or summarization. Less than one percent of runs failed due to external errors such as API errors, and we reran those trajectories.

Performance by workpaper area. Table 9 reports partial-credit scores by workpaper area. For each model, reported partial-credit scores are within each area. This breakdown shows how aggregate model performance varies across the different audit areas we evaluate.

Table 9: Partial-credit scores by workpaper area (%).

Workpaper area	Claude Opus 5.5	Claude Fable 5.1	Grok 4.6	GPT-5.6 Sol	GPT-6 Astra	GPT-5.6 Terra	Muse Spark 1.3	DeepSeek V4.1 Flash	GLM-5.2	Kimi K3	Gemini 3.8 Flash
Accounts payable	100.00	98.96	99.48	98.03	94.44	96.06	98.44	90.57	91.44	94.04	92.77
Accounts receivable	89.43	87.01	81.30	83.78	80.63	79.47	84.58	80.23	80.53	84.08	81.20
Cash	99.00	99.43	97.66	96.16	93.67	96.80	94.09	95.21	91.42	88.06	89.45
Debt	99.85	99.08	96.38	94.35	92.15	95.50	94.50	94.60	91.15	89.68	95.11
Fixed assets	95.82	94.82	95.49	89.42	88.21	93.39	95.87	93.70	94.94	92.58	87.64
Journal-entry completeness	100.00	99.58	99.65	100.00	100.00	96.04	99.24	96.46	99.58	93.19	100.00
Journal-entry risk testing	97.08	96.59	84.53	82.02	91.61	78.69	81.67	82.01	82.91	81.91	83.50
Revenue	87.60	85.07	78.67	75.81	88.63	77.03	72.98	74.82	70.47	68.52	75.83
Overall	**95.84**	**94.81**	**91.12**	**89.41**	**90.95**	**88.66**	**89.62**	**88.31**	**87.56**	**86.01**	**87.88**

Rubrics and grading. Prior approaches to grading work completed in spreadsheets have included methods such as extracting and checking exact cell coordinates (Ma et al., 2024) and evaluating completed work with LLMs (Yen et al., 2026). In this work, rubrics are written by auditors who check the presence, absence, and/or correctness of specific cells or rows deemed vital in each workpaper. As the purpose of these workpapers is to both collect *and* document evidence of the audit procedure being carried out, our rubric criteria are designed to accommodate substantively equivalent answers rather than being overly restrictive as different auditors may apply their own preferences for grading dimensions such as how much documentation is sufficient. Grading criteria range from checking that particular transactions are documented and recorded correctly to verifying that the correct conclusion(s) are reached.

The grading procedure implemented in this work leverages the fact that each worksheet in each of our workpaper templates consists of one or more vertically stacked tables with fixed columns. Agents may add or remove rows from each table during each task. Yet, they are instructed not to make changes to the columns of the templates themselves. Our provided rubric criteria are then applied to each table via a keyed approach that first locates the column of the table to be evaluated and then extracts values from target row(s). This allows grading to accommodate changes in row positions without relying on fixed cell coordinates. When multiple valid ways of completing a table make the relevant rows or cells difficult to identify programmatically, the grading procedure passes the relevant table content to the grader for evaluation. *GPT-5.6 Luna* with medium reasoning effort is used as the model for LLM-based evaluation.

Rubric Construction. To facilitate efficient rubric construction and revisions and to provide visibility into how workpapers are graded, the evaluation code highlights cells that are evaluated and subsequently graded as correct or incorrect with different colors. Furthermore, each graded cell is highlighted and annotated with its corresponding grading outcome and reasoning. In practice, these annotations allow auditors to more efficiently review and revise rubrics and researchers to understand and analyze what types of errors models make. LLMs were used to translate feedback provided by auditors on the rubrics into evaluation code.

System prompt. Figure 5 shows the system prompt provided to our agent in both the preparation and review settings. The notes also clarify what Account ID refers to, as some models commonly confused that terminology. The computer-use experiment in particular uses the system prompt supplied by the ChatGPT application. Figure 8 shows the instructions supplied as a user prompt for this experiment.

Code Release We release a *development version* of the task generation framework that preserves the overall structure of the code while omitting much of the domain-specific logic. This version is intended to make the framework easier for the broader research community to understand. The complete benchmark tasks and evaluation code are released in their entirety.

System prompt

```
You are a senior auditor tasked with completing audit workpapers.
Apply professional judgment to complete the workpaper templates
assigned to you.

You may add, remove, or reorder table rows when the evidence
requires it.  Do not add, delete, rename, reorder, or shift table
columns, and do not change any provided table header.

Save the completed workpaper in the workspace, then reply with a
concise completion summary when you are done working on it.  End
that final response with the completion signal on its own final
line:

<<TASK_FINISHED>>

Use the completion signal only after the requested deliverable has
been saved.

Notes:
- You may add or remove table rows.  Do not modify columns or
  headers.
- Account ID: GL account number from the chart of accounts or
  trial balance.
- Follow the Planning Context tab and follow the guidelines in
  that tab over the information in the Planning folder if there
  are discrepancies.

Runtime:  use the bash tool with bash syntax.  Python modules:
openpyxl, pandas, pypdf, docx (python-docx), pdfplumber, PIL
(Pillow), pytesseract.  Command-line tools:  libreoffice,
pdftotext, pdftoppm, tesseract, rg.  The container has no network
access.
```

Figure 5: System prompt provided to the agent in the preparation and review settings.

Instructions. Figure 6 shows an example of the user instructions provided at the start of a workpaper-preparation task.

Task prompt: Manufacturing-01 cash

```
Complete `cash_workpaper.xlsx` using `pbc_package/`,
`direct_to_auditor/`, and `planning/`.  Follow workbook procedures
and Planning Context.
Save to `/workspace/cash_workpaper.xlsx`.
```

Figure 6: Example task prompt for a workpaper completion task.

Review instructions. Figure 7 shows the corresponding user instructions in the review setting. The agent receives a previously completed workpaper and the same supporting materials from the engagement binder, with instructions to correct errors while preserving correct work.

Review prompt: Manufacturing-01 cash

```
Review and correct `cash_workpaper.xlsx` using `pbc_package/`,
`direct_to_auditor/`, and `planning/`.  Follow workbook procedures
and Planning Context.  Preserve correct work.
Save to `/workspace/cash_workpaper.xlsx`.
```

Figure 7: Example task prompt for a workpaper review task.

Computer-use instructions. Figure 8 shows the task instructions supplied as a user prompt to ChatGPT for one of our synthetic binders. The desktop application took 283 minutes to complete the eight workpapers, compared with a mean total execution time of 65 minutes for the bash-based agent on the same tasks. During the computer-use run, ChatGPT left document windows open after use, making its activity difficult to track (Figure 9).

Computer-use prompt

```
You are a senior auditor tasked with completing audit workpapers.
Apply professional judgment to complete the workpaper templates
assigned to you.

Binder folder:
<binder-folder>

Complete every workpaper in the binder folder's templates/
directory using the supporting materials in:
- data_room/pbc_package/
- data_room/direct_to_auditor/
- data_room/planning/

Use computer-use tools to navigate the folders, read the
supporting documents, and edit the workbooks in desktop
applications.

Access only the designated binder folder and its subfolders.  Do
not read, search, or modify files outside that folder, access
other binders, or use websites or external data sources.

Follow each workbook's procedures and Planning Context tab.  If
the Planning Context tab conflicts with documents in
data_room/planning/, follow the Planning Context tab.

You may add, remove, or reorder table rows when the evidence
requires it.  Do not add, delete, rename, reorder, or shift table
columns, and do not change any provided table header.

\Account ID" means the GL account number from the chart of
accounts or trial balance.

Save each completed workbook in templates/ under its original
filename.  Continue until you have completed and saved all
assigned workpapers.

Then provide a concise completion summary.
```

Figure 8: User prompt for completing all workpapers in a binder with computer-use tools. The local binder-folder path is replaced with a placeholder.



Figure 9: Final desktop overview during the computer-use experiment with ChatGPT Computer-Use.