ExcelAudit AI: Diving deeper into Dataset's Quality & Performance

Anonymous NAACL-HLT 2021 submission

Abstract

We often encounter new benchmarks for which manual analysis of diversity and coverage is time-consuming and subjective. This problem also translates while analyzing different model performances over benchmarks post evaluation, making it hard to pinpoint categorical scenarios where the model fails more often or which areas have now improved with the newer version. That's where ExcelAudit AI comes into place. It is an easy-to-use tool that generates reports and insights for analyzing and comparing both benchmarks and evaluated results. We provide demo use cases over SheetCopilot and SpreadsheetBench datasets and agents, with validation against human insights. [1] [2]

1 Introduction

The rapid integration of large language models (LLMs) into everyday workflows, such as through AI copilots in productivity suites like Excel and Sheets [3] has fundamentally changed how tasks are solved. Users are increasingly leveraging natural language descriptions to achieve complex tasks. This significant technological shift has, in turn, spurred the research community to develop new benchmarks (Ma et al., 2024; Payan et al., 2023) that comprise natural language queries and workbooks to compare the performance of LLM-powered systems(Li et al., 2023; Chen et al., 2025).

However, the proliferation of these domain-specific benchmarks presents a challenge. Researchers and practitioners are faced with an increasingly fragmented evaluation landscape, making it difficult to effectively track the underlying differences between benchmarks or select the most appropriate one for a given application. Compounding this issue is the prevailing methodological limitation in reporting: researchers typically focus only on single, aggregate performance metrics, such as pass rate. This superficial reporting fails to account for a system's true performance across the dataset's underlying distribution, task diversity, or the subtle nuances of task complexity.

This practice severely limits the community's ability to truly understand a model's specific strengths, diagnose precise failure modes, or conduct robust, apples-to-apples comparisons between competing systems. This widespread gap highlights a critical missing piece in the current ecosystem: *a dedicated framework for systematic benchmark meta-analysis*. To the best of our knowledge, there is no existing work or tool designed to assist researchers in critically comparing multiple established benchmarks to better understand their coverage, diversity of task types, inherent complexity, and other field-specific characteristics.

In this paper, we address this gap by introducing ExcelAudit AI, a novel framework designed for systematic meta-analysis of benchmarks along with performance over a system if present. The tool provides detailed analysis on benchmark distribution, showcasing aspects like coverage diversity, reliability, etc. Furthermore, the tool can help understand the performance gap of a system by looking beyond the simple pass rates and gain a deeper, more actionable understanding of the evaluation landscape by looking into cases failing and their characteristics.

2 Related Works

Early work focused on building large corpora of realistic spreadsheet data to support various use cases, including error detection, action suggestion, data cleanup, and user study analysis. Notable examples include the Enron corpus (Hermans and Murphy-Hill, 2015), Euses (Fisher and Rothermel, 2005), and Fuse (Barik et al., 2015). These datasets provide foundational resources for understanding real-world spreadsheet usage patterns.

[1] Codebase Link

[2] Link more examples & generations

[3] ai-copilots-in-office-suites

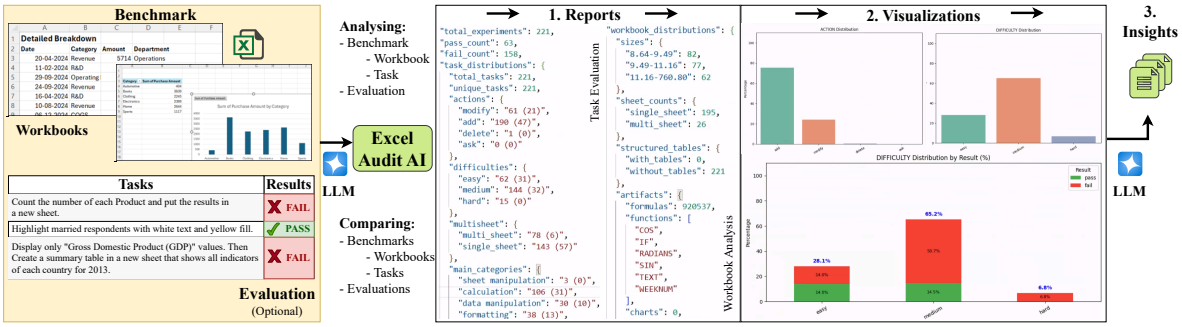


Figure 1: Overview of ExcelAudit AI tool

Table 1: Example benchmark & model analyzer and model comparison insights

Under-represented in Benchmark **Recommendations**	Evidence: Actions: ‘delete’ (1), ‘ask’ (0) out of 221 tasks. Analysis: Tasks involving deletion or inquiry are absent, indicating a lack of coverage for scenarios where users want to remove or ask questions on data. - Balance the distribution of task difficulties, ensuring more ’hard’ tasks to challenge models. - Diversify the types of actions (add, modify, delete, ask) to avoid overfitting to ‘add’ and ‘modify’. - Include more multi-sheet tasks to test cross-sheet operations.
Model Strength	Evidence: The ‘easy’ difficulty tasks have a 50% pass rate (62 total, 31 passing), which is significantly higher than medium (22%) and hard (0%). Analysis: The system demonstrates strong performance on easier tasks, indicating more reliability for straightforward Excel operations.
Model Comparison **Recommendations**	Evidence: Model A handles ’easy’ tasks with a 50% pass rate compared to Model B’s 35%. For ’medium’ difficulty, Model A achieves 22%, while Model B only manages 8%. Comparison: Model A is better suited for both easy and medium difficulty tasks, indicating a broader operational reliability, in comparison to Model B. - Prioritize on handling medium- and hard-difficulty tasks, as these represent a significant portion of the benchmark and have very low success rates. - Focus on multi-sheet and multi-step task capabilities, as these are common in practical Excel workflows and currently show very poor performance.

Though very few have focused on the underlying characteristics of these corpora. For instance, Jansen et al. (Jansen, 2015) performed a direct comparison between the Enron and Euses datasets to understand their diversity, structural overlap, and coverage of various spreadsheet features. While such manual, attribute-by-attribute analysis provides deep insights, it is time-consuming, resource-intensive, and becomes prohibitively expensive when applied to the increasing volume of new and larger datasets. Furthermore, manual analysis introduces subjectivity and is prone to human error due to the sheer size and complexity of modern spreadsheet corpora.

In parallel, with the integration of large language models (LLMs) into productivity tools such as Excel and Google Sheets, several recent works–SpreadsheetBench (Ma et al., 2024), SheetCopilot (Li et al., 2023), InstructExcel (Payan et al., 2023), and SheetAgent (Chen et al., 2025)–have explored building agentic workflows to help users solve tasks using natural language queries. Each of these efforts introduces a new benchmark along with its own evaluation setup, typically reporting results in terms of pass rates. While useful, such evaluations tend to be shallow, often lacking systematic analysis of failure cases and their underlying characteristics.

With ExcelAudit AI, we aim to bridge this gap by providing an easy-to-use framework for comparing benchmarks and evaluating system performance across multiple setups. In the sections that follow, we describe our experimental setup, followed by an implementation section.

3 Experiments

We evaluate the applicability of ExcelAudit AI on established open-source benchmarks (Li et al., 2023; Ma et al., 2024; Chen et al., 2025) and their associated agents.

3.1 Dataset

For comparative analysis, we consider the SheetCopilot and SpreadsheetBench benchmarks, which contain 221 and 907 workbook–task pairs, respectively. These tasks are designed to be complex, multi-step, and often over-specified with respect to the underlying instructions. Additionally, we use a subset of 20 instances from SheetCopilot, manually annotated by humans, to validate the quality and reliability of ExcelAudit AI.

3.2 Spreadsheet Agents

To compare agent performance on the SheetCopilot benchmark, we evaluate the SheetCopilot, SpreadsheetBench, and SheetAgent agents, each leveraging `GPT-4.1` as the underlying LLM. Beyond cross-agent comparison, we further investigate

Table 2: Results over SheetCopilot benchmark to analyze impact of agent & LLM model variations.

Agent	LLM	Pass rate
SpreadsheetBench	`GPT-4.1`	15.38
SheetAgent	`GPT-4.1`	22.62
SheetCopilot	`GPT-4.1`	28.51
SheetCopilot	`Phi-4`	25.84
SheetCopilot	`DeepSeek`	19.00

performance variations of the SheetCopilot agent across different LLM backbones. This allows us to examine how architectural consistency paired with different LLM versions influences outcomes. The resulting outputs are systematically analyzed using ExcelAudit AI.

3.3 Evaluation Metrics

To assess performance, we report pass rates obtained through the SheetCopilot evaluation framework. This system validates each generated workbook by comparing it against the corresponding ground-truth workbook, checking correctness at the level of individual components. Results are then categorized into pass–fail buckets, and the aggregated pass-rate percentages are summarized in Tab. 2. These quantitative results form the basis for deeper inspection and interpretation using ExcelAudit AI.

3.4 Models

Our tool interface is adaptive to generate using the LLM API based on the user input. However, for demonstrating the applicability, we generate all task analysis and analysis and comparison insights through `GPT-4.1` with temperature=0, n=1. The choice of LLM used for analysis and comparison does affect the quality of the responses, which has been presented within the Appendix. C. We thus choose to stick with `GPT-4.1` for the generations due to its strong capabilities in reasoning, instruction following, and key takeaway generation.

4 Implementation

The tool uses a two-step process: **1. generating report**, by extracting and summarizing information, and **2. generating insights** through individual analysis and comparisons. Fig.1 depicts the overview of the underlying architecture.

4.1 Generating Reports

Per-instance Information: Each instance within the benchmark consists of a workbook–task pair. To extract information from workbooks, we employ a heuristic method using `xlwings` and its XML contents. This allows us to obtain details about the artifacts present, including their size and language, both at the worksheet level and cumulatively for the overall workbook.

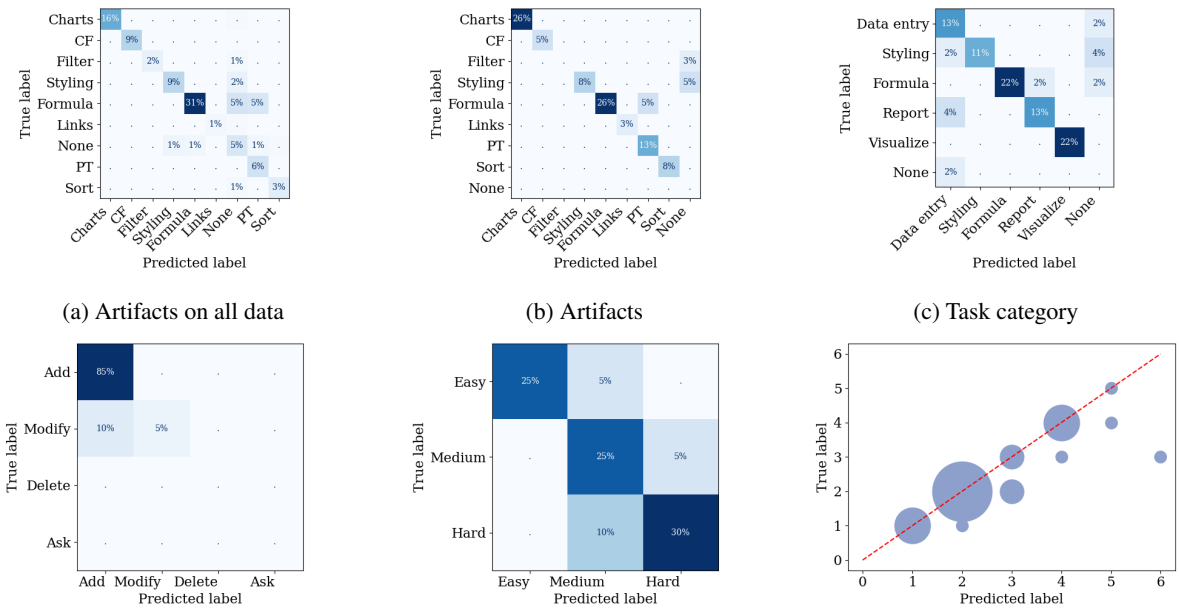
We further leverage an LLM to extract task-related information/metadata such as the underlying artifacts, atomic actions, language, and complexity. To validate the correctness of the LLM's extractions, we compare them against the hand-annotated artifacts in the 221 SheetCopilot benchmark, achieving an IoU overlap of 0.78. The distribution is shown in Fig. 2(a).

To holistically evaluate the LLM's outputs, we also compare the extracted information against a subsample of 20 manually annotated SheetCopilot ground-truth instances across all categories. We observe complete overlap in categories such as understandability, readability, and multi-sheet task annotations. Other categories including artifacts mentioned in tasks, task category, actions, and difficulty, achieve IoU overlaps of 0.88, 0.77, 0.90, and 0.80, respectively, as shown in Fig. 2. Similarly, the number of atomic actions has a mean squared error (MSE) of 0.4, highlighting the LLM's ability to analyze tasks. Overall, this suggests that the information extracted by the LLM is highly aligned with human expert annotations.

Summarizing information With the extracted per-instance information, the next step is to generate a summary. Depending on whether a system's performance over the benchmark is available, two types of reports can be produced: (1) *Benchmark Report*, which summarizes the number of artifacts per category, and (2) *Evaluation Report*, which computes the distribution of results per category across the pass and fail buckets. For example, *'chart': 4 (3)*, indicates that out of 4 samples, 3 passed. Appendix 4 illustrates the keys, along with the corresponding options within each key category, included in the report.

4.2 Generating Insights

Analysis insights Takeaway insights are generated by leveraging the LLM, which processes the reports and their numerical summaries. These insights capture key characteristics from both types of reports.



(a) Artifacts on all data (b) Artifacts (c) Task category

(d) Actions (e) Difficulty (f) # of atomic actions

Figure 2: Each captures the distribution of IoU overlap of actual vs. LLM-predicted categories for the task analyzer. Sub-figure (a) is for the 221 SheetCopilot benchmark, while the rest showcase 20 subsamples within the benchmark.

For benchmark analysis, the LLM highlights under- and over-represented categories, followed by recommendations on how to improve coverage and diversity. For evaluation analysis, it identifies the model's strengths and weaknesses in completing tasks across different categories and provides recommendations on reducing failure rates as well as on which samples should be further evaluated to enrich the overall insight report.

If the user wishes to analyze only the workbook or only the task aspects of the data, the subsequent sections are generated with both aspects treated independently.

Comparison insights We leverage LLM to first generate comparison insight per category and use these category insights and the benchmark or evaluation report to generate an overall comparison report. This report provides evidence and information on the strengths and weaknesses of each benchmark or evaluation against another, followed by summarized overall key takeaways.

We compile these results into an easily understandable HTML report for each task. We also add sections of visualizing each chart and table to further dive into the underlying distribution. Examples of extracted information & corresponding reports can be found within the Appendix. B.

5 Conclusion and Future Work

ExcelAudit AI provides a comprehensive and easy-to-use framework for analyzing and comparing benchmark results, offering insights into diversity, coverage, and failure categories. Its LLM-driven analyses are supported by human-validated annotations, strengthening its reliability and broadening its applicability across different evaluation scenarios.

Looking ahead, we aim to extend ExcelAudit AI beyond general-purpose reporting toward more domain-specific analyses. This includes allowing users to focus on targeted categories—such as charting tasks while omitting irrelevant gaps (e.g., pivot tables in certain benchmarks). More broadly, the underlying design of ExcelAudit AI can be expanded to support diverse benchmark modalities beyond spreadsheets, such as text, code, documents, and numerical data. These directions will enhance its flexibility and establish ExcelAudit AI as a general-purpose framework for benchmark analysis across domains.

6 Limitation

Though the major portion of the quantitative analysis of the report happens heuristically, the generation of textual insights and recommendations to provide easy understanding rather than having to

scratch heads with the numbers is done by the LLM. The choice of LLM thus comes into the picture, adapting based on the model's capability of generating quality analysis and comparison insights.

Moreover, the tool currently is built keeping Excel domain benchmarks in scope, revolving around metrics and information extraction over workbook or Excel task-related categories like multi-sheet queries, structured table instances, and artifacts (charts, formulas, and formatting). This hinders its application to other domains of benchmarks like code generation and can be further expanded to a generic system-based tool.

References

Titus Barik, Kevin Lubick, Justin Smith, John Slankas, and Emerson Murphy-Hill. 2015. Fuse: a reproducible, extendable, internet-scale corpus of spreadsheets. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, pages 486–489. IEEE.

Yibin Chen, Yifu Yuan, Zeyu Zhang, Yan Zheng, Jinyi Liu, Fei Ni, Jianye Hao, Hangyu Mao, and Fuzheng Zhang. 2025. Sheetagent: towards a generalist agent for spreadsheet reasoning and manipulation via large language models. In *Proceedings of the ACM on Web Conference 2025*, pages 158–177.

Marc Fisher and Gregg Rothermel. 2005. The euses spreadsheet corpus: a shared resource for supporting experimentation with spreadsheet dependability mechanisms. In *Proceedings of the first workshop on End-user software engineering*, pages 1–5.

Felienne Hermans and Emerson Murphy-Hill. 2015. Enron's spreadsheets and related emails: A dataset and analysis. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 2, pages 7–16. IEEE.

Bas Jansen. 2015. Enron versus euses: A comparison of two spreadsheet corpora. *arXiv preprint arXiv:1503.04055*.

Hongxin Li, Jingran Su, Yuntao Chen, Qing Li, and Zhao-Xiang Zhang. 2023. Sheetcopilot: Bringing software productivity to the next level through large language models. *Advances in Neural Information Processing Systems*, 36:4952–4984.

Zeyao Ma, Bohan Zhang, Jing Zhang, Jifan Yu, Xiaokang Zhang, Xiaohan Zhang, Sijia Luo, Xi Wang, and Jie Tang. 2024. Spreadsheetbench: Towards challenging real world spreadsheet manipulation. *Advances in Neural Information Processing Systems*, 37:94871–94908.

Justin Payan, Swaroop Mishra, Mukul Singh, Carina Negreanu, Christian Poelitz, Chitta Baral, Subhro Roy, Rasika Chakravarthy, Benjamin Van Durme, and Elnaz Nouri. 2023. Instructexcel: A benchmark for natural language instruction in excel. *arXiv preprint arXiv:2310.14495*.

You are an expert task analyzer for spreadsheet-related tasks. Given a natural language task description, extract and classify the following information as a JSON object with these keys:

- **artifacts**: (List[str])
  - Select a list of spreadsheet artifacts or features mentioned in task.
  - These can be picked from among the following option of artifacts only:
  - OPTIONS: ["formula", "charts", "pivot table", "filter", "conditional formatting", "links", "sort", "sparkline", "none", "formatting"]
- **action**: (List[str])
  - Select a list of actions that needs to be donw to complete the task.
  - These can be picked from among the following option of actions only:
  - OPTIONS: ["add", "modify", "delete", "ask"]
- **atomic_actions**: (int)
  - Estimates the number of intermediate actionable steps needed to finish the task.
  - Some example for actionable steps are: adding formula group, chart, pivot table to a new sheet, conditional formatting, applying a filter.
  - Each of the following steps should be counted as 1 atomic action. Cummulatively respond for the total number of atmic actions wthin the tasi
  - If it asks a question in context with Excel but does not ask to do any actionable item then atomic_actions is 0.
  - OPTIONS: [0-9]
- **difficulty**: (str)
  - The estimated difficulty of the task. Respond with only chosing from below options:
  - OPTIONS: ["easy", "medium", "hard"]
  - easy: tasks which can be completed by any generic person without any prior knowledge to Excel. Tasks like formulas like IF, SUM, or adding new sheet, sorting, chart.
  - hard: tasks which require deep understanding of Excel or are complex multi-step tasks. Complex tasks like formulas with VLOOKUP, COUNIF, pivot table, formatting and styling a chart may be present, or tasks with > 4 fairly complex steps.
  - Analyze deeply to understand the difficulty taking all the sub-actions of the task into account.
- **multisheet**: (bool)
  - Boolean indicating if the task involves selection or impact on multiple sheets.
  - OPTIONS: [true/false]
- **main_category**: (List[str])
  - Select a list of category for the task.
  - Here are some example category.
  - EXAMPLES: ["formula", "visualization", "calculation", "formatting", "reporting", "entry and manipulation"]
  - However, this list if not extensive. If you encounter a better fitting category assign that for task.
- **sub_category**: (Dict[str, List[str]])
  - Select a list of sub-category mappings for each category.
  - Here are some example sub-category per example category.

```
{
"formula": "['formula', 'function', 'calculation', 'compute', 'sum', 'if', 'count', 'vlookup', 'index', 'match']",
"visualization": "['bar', 'pie', 'line', 'scatter', 'pivot', 'histogram']",
"calculation": "['arithmetic', 'financial', 'statistical', 'logical', 'text']",
"formatting": "['cell', 'conditional', 'style', 'layout', 'theme']",
"reporting": "['dashboard', 'summary', 'presentation', 'export', 'print']",
"entry and manipulation": "['entry', 'input', 'write', 'type', 'edit']"
}
```

  - However, this list of sub-categories if not extensive. If you encounter a better fitting sub-category assign that for task.

Figure 3: Task analyzer prompt

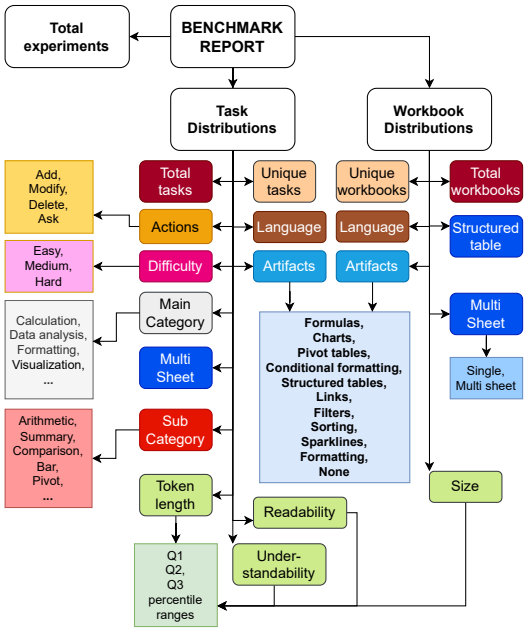


Figure 4: Benchmark report keys and categories

```
"over-represented": [
  {
    "evidence": "Actions: add = 202/221 (91%). Sub-category 'create' = 70.",
    "insights": "Create/add operations dominate, biasing models toward construction rather than
    editing, auditing, or removal."
  },
  {
    "evidence": "Main category calculation = 100/221 (45%). Sub-categories: arithmetic = 57, summary =
    53, cell = 52, organizing = 37.",
    "insights": "Benchmark is skewed toward basic arithmetic/summary and cell-level operations, which
    are simpler and less diverse."
  },
  {
    "evidence": "Difficulties: easy = 125 (57%), medium = 96 (43%), hard = 0.",
    "insights": "Task difficulty skews easy/medium; lack of hard tasks can inflate perceived
    performance."
  },
  {
    "evidence": "Artifacts: formulas = 119 (54%), formattings = 53 (24%), charts = 54 (24%).",
    "insights": "Heavy reliance on formula-centric tasks relative to other Excel ecosystems (Tables,
    filters, pivots, links)."
  },
  {
    "evidence": "Multisheet: single_sheet = 143 (65%).",
    "insights": "Single-sheet problem framing is favored; cross-sheet dependencies are underexplored."
```

Figure 5: Over-represented task categories with `GPT-5`

A Task Analyzer

Fig. 3 the following prompt used by LLM to analyze task into categories.

The same instructions are used by expert annotations in 20 samples of SheetCopilot benchmark to compute overlap in LLM capability of analyzing tasks. Expert annotators are internal scientist that have domain expertise on Excel, who were asked to do the task. This was a voluntary activity with no payment was made.

B Report structure

Fig. 4 shows the details of keys we extract information for during report generation for the task and workbook in the benchmark report to generate the insights further. These keys are also used in the evaluation report for evaluation insights.

The shared link provides in-depth examples of each of the variations of reports, visualizations, and analysis and comparison insights for each benchmark and model variation.

C Variation of LLM in Tool

Choice of LLM does impact the quality of insights you get from the tool. We experiment with other models to compare quality of results and reasoning. For smaller language models like `Phi-4` and `Mistral`, they had a hard time following instructions for response format, making it hard to parse results.

For other models like `GPT-5` reasoning and `DeepSeek`, the insights were comparable. Due to computation limitations, we didn't compute for all insights and comparisons as it was for `GPT-4.1`. However, here is an example of task insight generated from `GPT-5` reasoning over the tasks from the SheetCopilot dataset.

Table 3: Example task insight recommendations with GPT-5

Broaden action mix: target add $\leq 60\%$, modify $\geq 20\%$, delete $\geq 10\%$, ask $\geq 10\%$; include safe-destructive operations (delete rows/cols, clear formats, remove duplicates) and QA tasks that output values/explanations.
Introduce hard tasks: nested/dynamic array formulas (LET/LAMBDA, FILTER/SORT/UNIQUE, XLOOKUP, XMATCH), multi-step workflows combining pivots, charts, CF, and cross-sheet references with constraints.
Increase multi-sheet tasks: cross-sheet consolidation, 3D references, summary sheets, and sheet-level operations (move/-copy/rename/hide/protect).
Expand under-covered categories: lookups/joins or index match, date/time, text ops (TEXTSPLIT, TEXTBEFORE/AFTER, TRIM, CLEAN, SUBSTITUTE), aggregation (SUMIFS/COUNTIFS), data cleaning/transformation, statistical (CORREL, LINEST), number formats, and grouping/outline.
Enrich artifact coverage: include tasks with structured Tables ($\geq 30\%$), advanced filters ($\geq 20\%$), pivots with slicers/time-lines ($\geq 25\%$), sparklines ($\geq 10\%$), hyperlinks/external links ($\geq 10\%$), named ranges, and data validation ($\geq 20\%$).
Visualization diversity: beyond bars/lines, add waterfall, histogram, box & whisker, combo charts with secondary axes, scatter with trendlines, and formatting details (titles, labels, legends, axes, and templates).
Language and locale: add non-English prompts/workbooks (e.g., es, de, fr, and ja) and locale variations (decimal separators, date formats, and localized function names) to reach $\geq 20\%$ non-English.
Instruction diversity: vary length and style (procedural, goal-driven, narrative, ambiguous) and include incomplete/messy specs; aim for balanced readability/understandability distributions rather than near-zero.
Taxonomy hygiene: merge duplicate/overlapping subcategories (e.g., "copy" vs. "copying" and "transform" vs. "transforming"), define coverage targets per main/subcategory, and maintain consistent labeling.
Artifact-task alignment: include tasks that modify and delete existing pivots/charts/CF/filters, not just create; seed workbooks with those artifacts to enable non-add actions.
Confidence criteria: use stratified evaluation by action, difficulty, category, artifact, and sheet count; require $\geq$ 30–50 tasks per stratum and 95% CI half-width $\leq 7\%$ before claiming strong performance.
Edge-case content: hidden/merged cells, protected/hidden sheets, large ranges, spilled arrays, circular references (diagnosis), volatile functions, and mixed data types to test robustness.
Scenario/what-if: Goal Seek, Solver, and Scenario Manager tasks to cover optimization and sensitivity analysis.
Cross-ecosystem tasks: CSV import cleanup, external references, and hyperlink management to reflect real workflows.


```
"under-represented": [
  {
    "evidence": "Actions: delete = 2/221 (0.9%), ask = 0/221 (0%).",
    "insights": "Tasks overwhelmingly avoid deletion and question-answer style prompts; the benchmark
    under-exposes models to safe destructive edits and direct analytical answering."
  },
  {
    "evidence": "Difficulties: hard = 0/221 (0%), easy = 125/221 (57%).",
    "insights": "No hard tasks limits assessment of complex reasoning, multi-step logic,
    error-handling, and advanced Excel proficiency."
  },
  {
    "evidence": "Multisheet: multi_sheet = 78/221 (35%), single_sheet = 143/221 (65%).",
    "insights": "Cross-sheet navigation and 3D references are under-tested; many real-world
    spreadsheets are multi-sheet."
  },
  {
    "evidence": "Main categories: sheet manipulation = 10, visualization = 10, freeze panes = 6, entry
    and manipulation = 4 vs calculation = 100.",
    "insights": "Operational and layout workflows (sheet ops, freeze/lock, UI-centric tasks) and richer
    visualization tasks are sparse compared to calculations."
  },
  {
    "evidence": "Sub-categories with very low counts: lookup = 4, date/time = 4, aggregation = 1,
    correlation = 1, cleaning = 1, transforming = 1, grouping = 1, number format = 1, chart = 1, new
    sheet = 1.",
    "insights": "Core business skills (lookups/joins, date-time handling, data cleaning/transformation,
    statistical correlation, formatting) are underrepresented."
  },
  {
    "evidence": "Artifacts: structured_tables = 0, sparklines = 0, links = 4, filters = 12,
    pivot_tables = 23 (≈10%).",
    "insights": "Modern Excel features (Tables with structured references, sparklines, links, filters,
    pivots) are limited or absent, reducing coverage of realistic workflows."
  },
  {
    "evidence": "Languages: en = 221/221 (100%).",
    "insights": "No multilingual tasks; locale effects (function names, delimiters, date formats) are
    untested."
  },
  {
    "evidence": "Readability: 220/221 in 0.00-0.00; Understandability: 213/221 in 0.00-0.00.",
    "insights": "Instruction style diversity is extremely low; models are not stressed with varied
    phrasing, ambiguity, or narrative context."
```


Figure 6: Under-represented task categories with GPT-5