



FINCH: Benchmarking Finance & Accounting across Spreadsheet-Centric Enterprise Workflows

FINCH members*

finworkbench@gmail.com

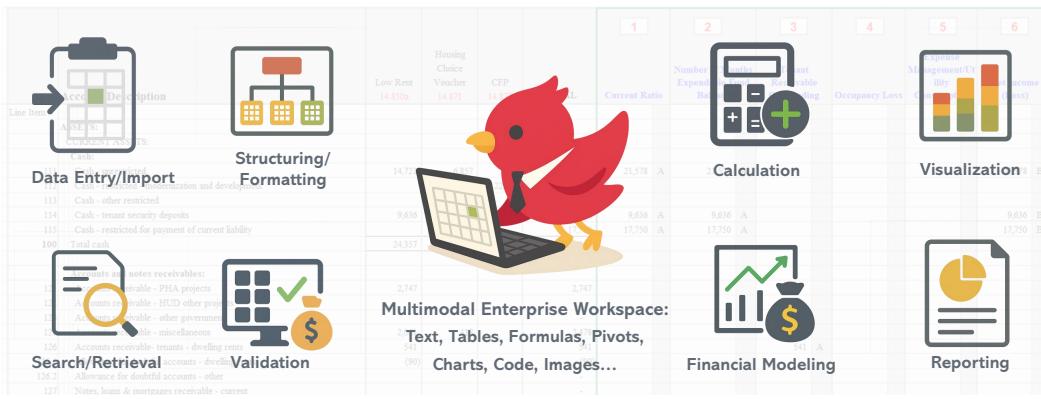


Figure 1: Real-world F&A work is messy, spanning heterogeneous and large-scale artifacts such as spreadsheets and PDFs. It's also long-horizon and knowledge-intensive: workflows interleave multiple tasks and span diverse domains such as budgeting, trading, asset management, and operations.

Abstract

We introduce a finance & accounting benchmark (FINCH) for evaluating AI agents on real-world, enterprise-grade professional workflows—interleaving data entry, structuring, formatting, web search, cross-file retrieval, calculation, modeling, validation, translation, visualization, and reporting. FINCH is sourced from authentic enterprise workspaces at Enron (15,000 spreadsheets and 500,000 emails from 150 employees) and other financial institutions, preserving in-the-wild messiness across multimodal artifacts (text, tables, formulas, charts, code, and images) and spanning diverse domains such as budgeting, trading, and asset management.

We propose a workflow construction process that combines LLM-assisted discovery with expert annotation: (1) LLM-assisted, expert-verified derivation of workflows from real-world email threads and version histories of spreadsheet files, and (2) meticulous expert annotation for workflows, requiring over 700 hours of domain-expert effort. This yields 172 composite workflows with 384 tasks, involving 1,710 spreadsheets with 27 million cells, along with PDFs and other artifacts, capturing the intrinsically messy, long-horizon, knowledge-intensive, and collaborative nature of real-world enterprise work.

We conduct both human and automated evaluations of frontier AI systems including GPT 5.1, Claude Sonnet 4.5, Gemini 3 Pro, Grok 4, and Qwen 3 Max, and GPT 5.1 Pro spends 48 hours in total yet passes only 38.4% of workflows, while Claude Sonnet 4.5 passes just 25.0%. Comprehensive case studies further surface the challenges that real-world enterprise workflows pose for AI agents.

Dataset: <https://huggingface.co/FinWorkBench>

*Full author list: Appendix A.

1 Introduction

Frontier AI systems are increasingly transforming professional workspaces. AI-assisted tools like ChatGPT [41], Claude [2], Gemini [21], and Copilot [39] are now embedded in daily enterprise workflows—helping professionals draft documents, explore data, manipulate spreadsheets, and generate reports. These tools are particularly impactful in finance and accounting (F&A), a high-stakes, knowledge- and labor-intensive domain critical to every organization.

However, real-world F&A work is inherently **messy** with substantial contextual complexity: artifacts are interconnected across heterogeneous spreadsheets, PDFs, and more, evolving through multiple versions with collaborative edits [28]; spreadsheets contain large, complex structures [17] with cross-sheet references, intricate layouts, inconsistent formatting, cryptic terms, erroneous formulas, and multimodal artifacts such as charts, images, and code. It is also **long-horizon** [42]: workflows demand multi-step reasoning spanning data entry, editing, retrieval, calculation, modeling, validation, reporting, and more.

This raises a key question: *Can today’s frontier AI agents actually handle the messy, long-horizon, and knowledge-intensive workflows that professionals face daily?*

To answer this, we introduce FINCH, an enterprise-grade F&A benchmark sourced from authentic enterprise environments. FINCH captures the intrinsic complexity of professional work through:

- **In-the-wild enterprise sourcing:** FINCH is built around authentic enterprise spreadsheets, emails, and PDFs from real-world enterprise workspaces—primarily Enron [19] (about 15,000 spreadsheet files and 500,000 emails from 150 executives and employees) and EUSES [20] (about 450 financial spreadsheets from various sources), along with securities and asset management firms, global organizations such as World Bank [4], and Canadian and British governments [11, 24]. Documents are large, cross-referenced, and messy—containing rich multimodal artifacts such as tables, formulas, charts, pivots, and images.
- **Rigorous construction process:** We propose a novel workflow construction pipeline grounded in real collaborative context of emails and versioned artifacts. We induce workflows from enterprise email threads and attachments, where collaborators naturally describe, discuss, and track workflows as part of their daily work. And we propose an LLM-assisted, expert-verified method to derive workflows by analyzing changes across versioned spreadsheets, surfacing the underlying goals that drive professionals’ work. Annotators must reason over large multi-sheet workbooks, and subtle version deltas to infer what the original analyst was trying to achieve, making the annotation process substantially more difficult than curating QA pairs over isolated tables.

We compile 172 meticulously annotated, enterprise-grade workflows built on 1,710 spreadsheets, along with PDFs and other artifacts, collectively capturing the compositional, messy, knowledge-intensive, and collaborative nature of real work. Each workflow spans one or more interdependent tasks—data entry, editing, retrieval, calculation, modeling, validation, translation, visualization, and reporting—mirroring how professionals actually work on artifact manipulation and creation.

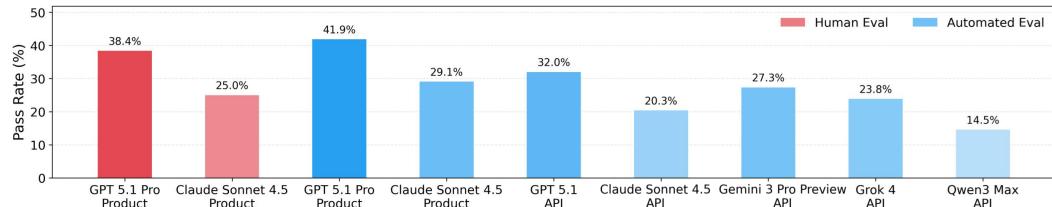


Figure 2: Model pass-rate comparison on FINCH workflows. Bars show overall workflow success rates for product-side agents and API-based models. Detailed settings can be found in Section 3.

We evaluate a spectrum of frontier AI systems—including Claude Sonnet 4.5, GPT 5.1, Gemini 3, Grok 4, and Qwen 3—using both expert evaluation and a novel automated evaluation pipeline that closely aligns with expert judgments. Our experiments reveal that even the frontier agents pass only fewer than 40% of workflows (GPT 5.1 Pro spends 16.8 minutes per workflow on average), highlighting the substantial challenges that FINCH poses for AI agents. Comprehensive case analyses further surface concrete challenges that real-world enterprise workflows pose for AI agents.

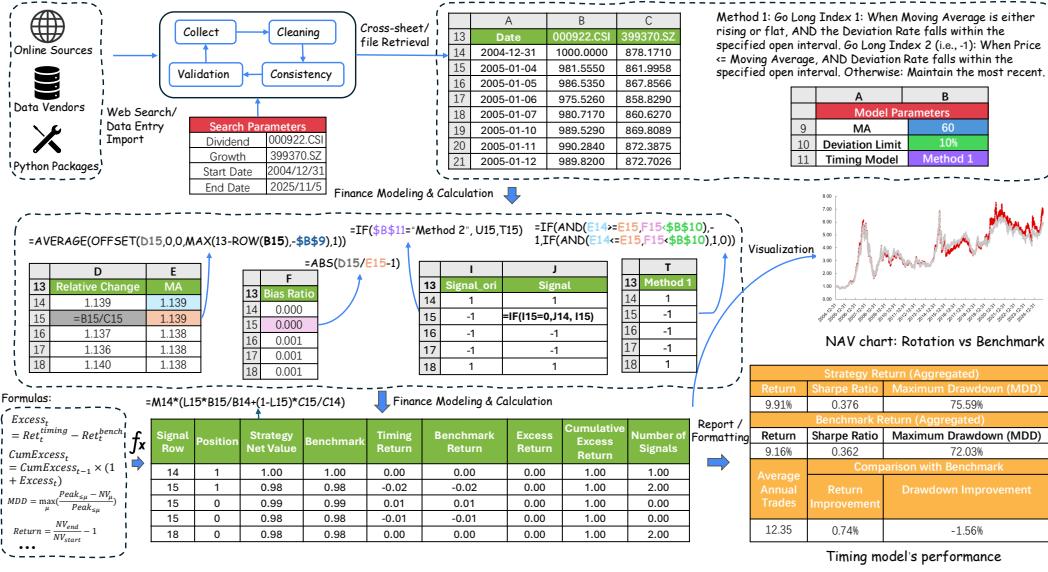


Figure 3: Illustration of an end-to-end predictive modeling workflow commonly performed by financial analysts. The workflow involves multiple steps, including web search, data import, cross-sheet and cross-file retrieval, calculation and financial modeling, visualization, and report generation. More illustrative examples in FINCH are presented in Appendix E.

2 FINCH: A Real-world Finance & Accounting Workflow Benchmark

2.1 Dataset Construction

We propose a novel workflow construction pipeline grounded in the real collaborative context of emails and versioned artifacts, as illustrated in Figure 4 (a–c). First, we induce workflows from enterprise email threads and versioned documents, where collaborators naturally describe, discuss, and track work as part of their daily routines. Second, we derive workflows by analyzing changes across versioned spreadsheets, surfacing the actual data transformations and analysis steps that professionals performed. Third, we leverage high-quality spreadsheets and reports: we design workflows, author task instructions, and revise these spreadsheets and reports so that they serve as the input files and reference solutions.

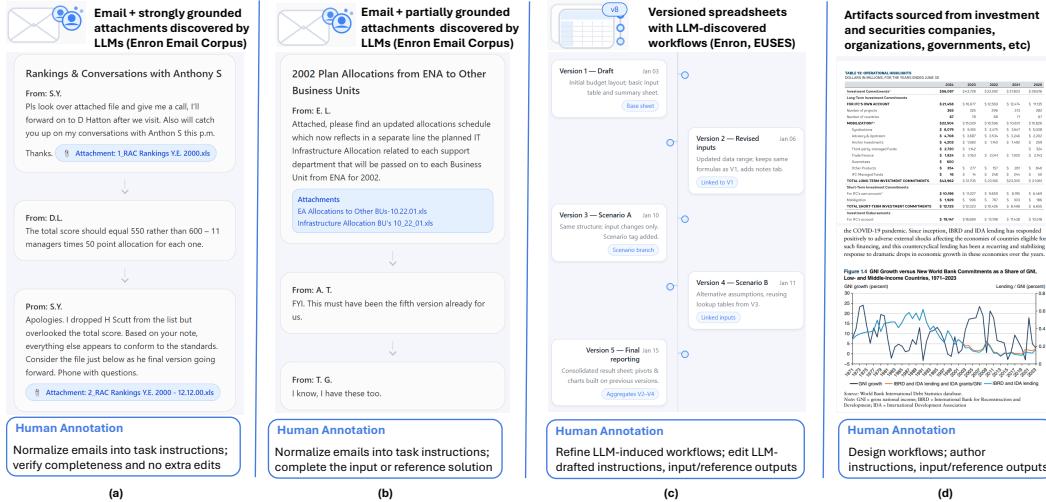


Figure 4: Illustration of our workflow construction pipeline from real-world enterprise emails, versioned spreadsheets, and high-quality artifacts.

All annotated workflows from these different channels are consolidated into a unified schema with consistent fields (NL instruction, input files, reference outputs), and each workflow is tagged with task types (e.g., data entry/import, structuring, validation) and business types (e.g., planning and budgeting, pricing and valuation, operations, asset management). Note that the reference outputs include both file-based reference answers (for most generation/editing cases) and textual reference answers (for a small number of QA and summary/visualization cases).

2.1.1 Workflow from Enterprise Email Threads

We first mine enterprise email threads to surface workflows. Starting from the Enron Email Corpus, we prompt GPT-5 to identify collaborative messages that (i) explicitly state a business goal (e.g., “update the RAC rankings” or “revise the 2002 allocations”) and (ii) reference one or more attached spreadsheets. For each selected thread, the model summarizes the communicative intent and articulates a workflow description.

We distinguish two cases. In the *strongly grounded* case illustrated in Figure 4 (a), both the input and the final reference artifacts for the workflow are already present as attachments in the thread (e.g., an initial ranking file and a corrected version). Strongly grounded cases well motivated this benchmark, but they are relatively rare. In the *partially grounded* case illustrated in Figure 4 (b), the email specifies a clear goal, but only some of the required artifacts are attached (e.g., just the updated schedule, or an intermediate report). Across both cases, human experts normalize conversational email text into task instructions and abstract away idiosyncratic details while preserving the business intent. For strongly grounded threads, annotators primarily verify that the attached files exactly implement the requested change without extra edits. For partially grounded threads, they either identify the missing artifacts from versioned spreadsheets and revise them to align with the described workflow—carefully avoiding the introduction of new changes—or create the missing artifacts themselves, which typically requires much more effort.

2.1.2 Workflow Derivation from Versioned Spreadsheets

Beyond explicit messages in email threads, we propose to discover workflows that are implicitly captured in spreadsheet version histories, as illustrated in Figure 4(c). We collect families of versioned workbooks from the Enron and EUSES repositories and apply an LLM-based differencing procedure that recognizes consecutive versions and infers the underlying workflow.

For each recognized pair (or chain) of versions, we prompt GPT-5 to propose (i) one or more workflow types (e.g., “date-stamped versioning, assumption updates, and error correction”, “data entry, structuring, and visualization”) and (ii) a detailed NL description of all changes. Human experts then validate and refine these LLM-induced workflow candidates. They first determine whether the proposed diffs constitute a coherent and meaningful workflow rather than incidental churn. For accepted cases, they (i) rewrite the draft description into a precise task instruction that describes the transformation, and (ii) edit the corresponding workbook versions so that the designated input and reference files cleanly realize the described workflow without introducing out-of-scope changes beyond the instruction. The corresponding input and reference files are thus grounded in the actual versions used in the diff, yielding workflow instances that do not rely on email context but are anchored in real enterprise spreadsheet evolution.

2.1.3 Workflow Sourced from High-quality Artifacts

Finally, we curate workflows from high-quality spreadsheets and reports drawn from the Enron and EUSES corpora, various investment and securities companies, international organizations, and national governments (e.g., the World Bank and the Canadian and British governments). Domain experts decompose these finalized financial artifacts into intermediate artifacts that serve as inputs, while treating the original documents as reference outputs, and then author realistic workflow instructions. For example, a valuation model from an investment firm can be turned into a financial modeling task; a World Bank report can be used to define a data-driven summarization and visualization task; and bilingual reports from the Canadian government can be used to construct translation and consistency-checking tasks. We additionally leverage labeled samples from existing datasets: we adapt 10 financial cases from WideSearch [47] into web-search-centric workflows and extend them into multi-step calculation and visualization pipelines, and we leverage 3 examples from DABStep [18] to construct multi-source question answering workflows.

2.1.4 Quality Control

Given the high complexity of each workflow, rigorous quality control is essential. We perform inter-annotator validation on all workflows and use the ChatGPT 5.1 Pro and Claude-Sonnet-4.5 product-side agents as secondary checkers: given the instruction and the input and reference files, the model is asked to judge whether the reference output is consistent with the instruction and whether any obvious steps are missing. LLM-based judgments are used to flag potential defects for human review. Together, these quality-control procedures yield a collection of workflows whose NL instructions, input files, and reference outputs are well aligned.

2.2 Dataset Analysis

FINCH comprises 172 meticulously annotated, enterprise-grade workflows that collectively capture the compositional, messy, multimodal, and collaborative nature of real finance and accounting work. Across these workflows, the corpus contains 1,710 spreadsheets together with 13 PDFs, 7 images, 3 Word documents, and additional files such as JSON, CSV, and Markdown. This mixture reflects how real analysts coordinate over heterogeneous artifacts rather than clean, single-table inputs.

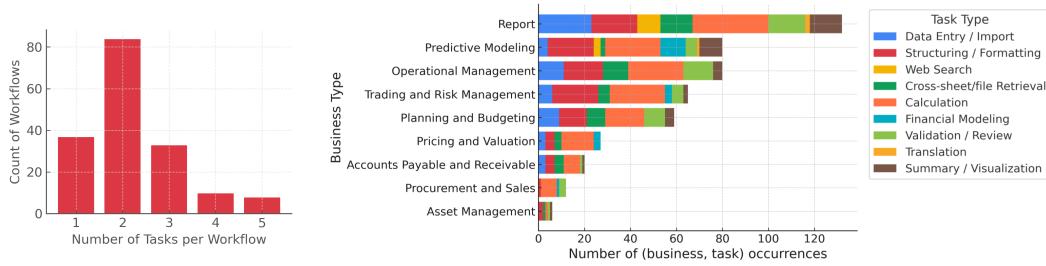


Figure 5: Distribution of number of tasks per workflow and task types across business types.

Figure 5 summarizes the coverage of task and business types. On the task side, categories are:

- Calculation (119 workflows): filling in formulas or computing figures (e.g., net value).
- Structuring / Formatting (86): reorganizing tables (e.g., adjusting hierarchies), formatting content (e.g., font size and cell fill), and inserting/deleting rows or columns.
- Data Entry / Import (44): transcribing or importing data from spreadsheets, PDFs, images, or external sources into spreadsheets.
- Validation / Review (37): checking consistency and reconciling calculations within a sheet or across sheets/files.
- Cross-sheet/file Retrieval (36): pulling values from multiple sheets or files into a target workbook.
- Summary / Visualization (33): producing summaries or charts that surface key financial insights.
- Financial Modeling (15): extending or calibrating valuation and timing models, often via scenario and sensitivity analysis.
- Web Search (11): collecting financial data from the web and integrating it into spreadsheets.
- Translation (3): translating spreadsheets or reports while preserving structure, formatting, and layout.

On the business side, workflows span reporting (48 workflows), trading and risk management (35), predictive modeling (33), operational management (36), planning and budgeting (26), pricing and budgeting (15), accounts payable/receivable (10), as well as procurement and sales (7) and asset management (3); some workflows are tagged with multiple business types. Overall, the distribution indicates that FINCH targets core finance and accounting verticals rather than curated toy tasks.

2.2.1 Task Compositionalty

FINCH is explicitly designed around composite workflows rather than isolated tasks. As shown in Figure 5, only 37 workflows (21.5%) are single-task; the remaining 135 (78.5%) involve multiple tasks. Importantly, each “task” itself typically requires substantial multi-turn reasoning: for example,

web search often entails many rounds of LLM calls to discover, filter, and verify evidence; cross-sheet retrieval requires repeated calls to read and locate key information across multiple sheets; and calculation usually spans many formulas distributed over different rows and columns.

These tasks are not independent subtasks, but are interleaved around shared spreadsheets and files. A typical workflow may begin with structuring or importing raw data, proceed to cross-sheet or cross-file retrieval, and then culminate in calculations, modeling, or reporting. The distribution in Figure 5 shows that most workflows weave multiple tasks.

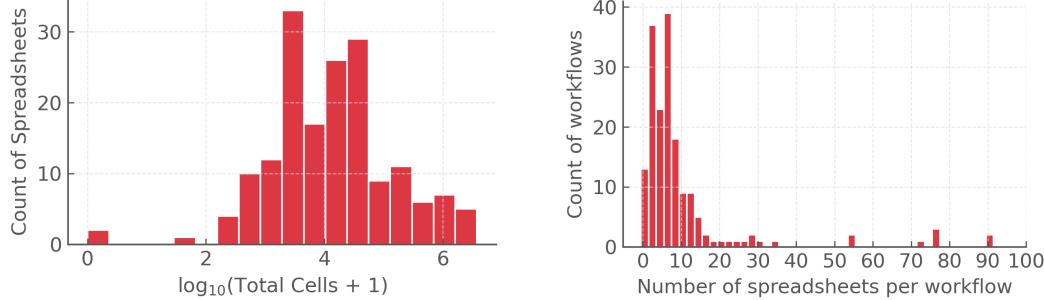


Figure 6: Distribution of the number of sheets and cells per workflow.

2.2.2 Messiness

The source files in FINCH are intentionally large, multi-sheet, and structurally complex. At the file level, 86.6% of workflows involve more than one file when counting both input and reference artifacts, and some workflows touch up to 14 distinct files. At the spreadsheet level, 92.4% of workflows involve multiple sheets, with an average of 8 sheets and a long tail reaching 91 sheets. As a result, systems must navigate cross-sheet dependencies, hidden logic, and scattered intermediate calculations rather than operating on a single “analysis” sheet. Moreover, most spreadsheets exhibit complex layouts that interleave text, tables, formulas, and charts, as well as intricate table structures with merged cells, nested headers, hierarchical data, blank rows and columns, and other irregularities.

Cell-level statistics further highlight the scale of the data. The median workflow covers 15K cells (157K on average for all workflows), with the largest workbook scaling to 3.7 million cells. Formula density is similarly skewed: while workflows contain an average of 21.5K formulas (the median is 212), the most complex ones have hundreds of thousands of formulas, reflecting deeply nested calculations and long dependency chains. Taken together, these properties create a challenging regime in which models must reason over large, noisy, and highly irregular spreadsheet layouts, rather than clean, rectangular tables.

2.2.3 Multimodality

Although FINCH is spreadsheet-centric, the workflows are inherently multimodal. Around 10.5% of workflows link spreadsheets with additional non-spreadsheet artifacts such as PDFs, Word documents, and images, and 7.6% explicitly require reasoning over PDFs or images. Within the spreadsheets themselves, 20.3% of workflows include charts and 2.3% feature pivot tables, so models must understand not only raw cell values but also derived visual summaries and explicit aggregation structures (and most workflows involve implicit aggregation structures). This multimodal, cross-artifact structure stands in contrast to prior benchmarks that operate purely on isolated tables, and better reflects the environments in which enterprise finance and accounting tasks actually occur.

2.3 Evaluation Method

2.3.1 Human Evaluation

We conduct human evaluation on all workflows to directly assess model performance. For each workflow, annotators read the NL instruction, inspect the input, reference, and model output files side by side (typically by aligning spreadsheets or documents in adjacent tabs), and determine whether the model has faithfully completed the requested job. A workflow is marked as successful only if

the model generates or revises the content and structure in accordance with the instruction and no critical errors, omissions, or unintended changes are introduced; otherwise, it is labeled as a failure. Importantly, evaluation is based on whether the instruction has been satisfactorily fulfilled rather than on a purely mechanical comparison between model and reference outputs, since there may be multiple acceptable solutions for summarization, visualization, formatting, formulas, and related aspects. To reduce subjectivity and ambiguity, annotators ultimately assign a binary pass/fail label for each workflow. These human judgments serve as the gold standard for measuring model performance and for validating the reliability of our automatic evaluation method.

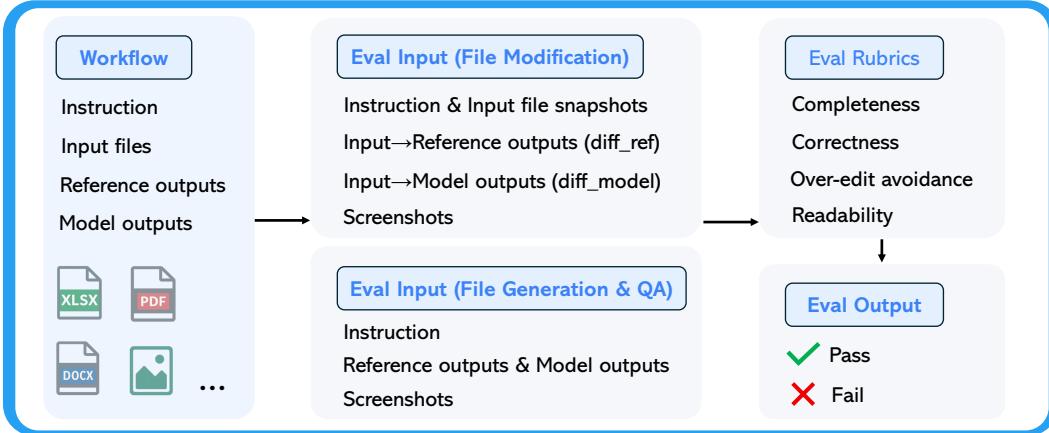


Figure 7: Illustration of our automated evaluation pipeline. Here, diff_ref denotes the diff between the input file and the reference output, and diff_model denotes the diff between the input and the model output. We categorize all workflows into file modification, file generation, and file QA. This categorization is independent of the task types in Section 2.2; for example, a *calculation* task may generate a new file, modify an existing one, or simply return a textual answer.

2.3.2 LLM-as-Judge Evaluation

To scale evaluation, we employ an LLM-as-judge framework that supports the three high-level task types in FINCH: *modify* (editing input artifacts), *generate* (creating new files such as workbooks and documents), and *QA* (answering questions based on one or more artifacts). The framework accepts heterogeneous inputs—including `.xlsx`, `.txt`, `.docx`, `.md`, `.pdf`, and images—and normalizes them into a sequence of textual inputs and screenshot images for the judge model.

For modification tasks, especially on spreadsheets, the framework computes structured diffs between the input and the reference output (diff_ref) and between the input and the model output (diff_model), and then builds a compact “input snapshot” (`snapshot`) that, for each modified sheet, retains only the first and last ten rows and the first five columns (which typically capture table headers and layout) together with rows and columns that contain edited cells. This preserves the crucial context for diff_ref and diff_model while dramatically reducing token length. In parallel, the framework renders screenshots (`screenshot`) of sheets containing changes from the input, reference, and model output, so that the judge perceives merged cells, conditional formatting, charts, and other layout-sensitive properties that are difficult to encode as text alone.

For generation tasks involving spreadsheets, the framework extracts all cell values and formulas from both the reference and the model output, and captures screenshots of every sheet, since the entire generated artifact must be verified rather than just localized edits. For QA tasks, it feeds a rich text-and-image sequence constructed from the reference answer and the model’s response, optionally augmented with relevant input files when the question requires grounding in external artifacts.

We design three task-specific judge prompts for modify, generate, and QA, respectively, but they share a common evaluation rubric. In all cases, the judge is instructed to focus on (i) *completeness* with respect to the NL instruction, (ii) *numerical and logical correctness* of derived values and formulas, (iii) the *over-edit avoidance*, penalizing unnecessary and unexpected changes of the workbook beyond instruction, and (iv) *readability* of the formatting and structure. Exact cell-by-cell equality with the reference is not required when multiple solutions are acceptable (e.g., alternative layouts, equivalent

formulas, or different but semantically equivalent summaries); instead, the judge decides whether the model has satisfactorily fulfilled the instruction. To reduce subjectivity, the judge outputs a binary score (pass/fail) along with a short NL rationale. In some web search tasks, the rubric permits small tolerance bands when comparing values, allowing for discrepancies in data from different sources.

This LLM-as-judge framework not only automates large-scale evaluation but also surfaces subtle spreadsheet errors (such as formulas silently replaced with static values) that are difficult to catch with GUI-based human inspection alone. In Section 3.2, we report the consistency between human and automated evaluations and show that the LLM-as-judge scores closely track human judgments.

3 Experiments

3.1 Agents and Models

3.1.1 Product-side Agents

We evaluate two frontier product-side agents: (i) *ChatGPT* using the GPT 5.1 model in Pro mode, and (ii) *Claude* using the Sonnet 4.5 model in thinking mode. We focus on these two systems rather than alternatives such as Gemini or Grok because they natively support returning downloadable files (e.g., spreadsheets) as outputs, rather than emitting code or markdown-formatted tables that are not intuitive for human evaluation. For both agents, we enable their external web browsing, but disable using historical chats so that each workflow is evaluated independently and without cross-run leakage. Since model updates are frequent and manual evaluation is very time-consuming, we used the latest model from our final round of experiments and did not consider subsequent updates.

3.1.2 API-based Models

We evaluate five frontier large language models (LLMs), shown in Table 1, through API interfaces. Our evaluation pipeline builds upon the open-source SpreadsheetBench framework [37]. Details of the execution paradigm and prompting strategies are deferred to the Appendix B.1. While SpreadsheetBench provides only a baseline implementation, we introduce several practical enhancements to improve its applicability to real-world tasks.

Model	Provider	Context	Max Output	Vision	Native PDF
GPT 5.1 [40]	OpenAI	400K	128K	✓	✓
Claude Sonnet 4.5 [3]	Anthropic	1M [†]	64K	✓	✓
Grok 4 [49]	xAI	256K	256K	✓	—
Qwen 3 Max [52]	Alibaba	256K	32.8K	—	—
Gemini 3 Pro Preview [22]	Google	1.05M	65.5K	✓	✓

Table 1: API-based model configurations. Context and output limits are measured in tokens. Vision indicates native image input support, while Native PDF refers to direct PDF file ingestion via the provider’s API without explicit text extraction. [†]Available via long-context beta API mode.

Spreadsheet Encoding. SpreadsheetBench only produces simple text tables without preserving cell addresses, data types, or formulas. However, these details are essential for tasks in FINCH. We extend SpreadsheetLLM [17] encoding and introduce a *semantic-rich tuple encoding* that preserves full structural and semantic fidelity. Each sheet begins with its name and the corresponding data bounding box (e.g. ## Sheet: [name] (A1:Z100)). We then serialize the bounded region using a Markdown-based format. Each cell is encoded as a tuple (Address, Value, Type, Formula), where Address denotes the cell reference (e.g. A3), Type indicates the data type (T = Text, I = Integer, F = Float, D = Date, B = Boolean), and Formula records the cell formula (e.g., =SUM(A1:A10)->100).

Multimodal Input Handling. We extend the framework to support multimodal inputs involving images and PDFs. For vision-capable models (GPT 5.1, Claude Sonnet 4.5, Grok 4, and Gemini 3 Pro), we use each provider’s official multimodal API to transmit visual inputs alongside text prompts. For PDF documents, we adopt a tiered strategy. Models with native PDF support—GPT 5.1, Claude Sonnet 4.5, and Gemini 3 Pro Preview—directly ingest PDF files via their file upload interfaces, enabling analysis of both textual and visual elements without pre-extraction. For Grok 4, which lacks

native PDF support, we extract text using PyMuPDF and include it in the `pdf_content` field. For Qwen 3 Max, which lacks multimodal support entirely, both image and PDF content are converted to textual descriptions. While this fallback retains semantic cues, it loses layout and visual context.

Context Management. To handle large spreadsheets that may exceed model context limits, we implement automatic truncation. We reserve 32K tokens for model output—sufficient for comprehensive code generation and analysis while remaining within the output limits of all evaluated models. Truncation is triggered when input exceeds the remaining capacity, removing content from the end of spreadsheet data with an explicit notice appended to inform the model of data loss.

3.2 Experimental Results

Web agents (ChatGPT 5.1 Pro vs. Claude Sonnet 4.5). As shown in Figure 2 and Table 3, ChatGPT 5.1 Pro achieves the best overall pass rates on FINCH. Their advantage largely comes from rich interactive affordances: they can iteratively inspect spreadsheets, revise intermediate states, and recover from partial errors over many tool calls. However, it still solves fewer than 40% of the FINCH workflows, suggesting that real-world finance and accounting work remains far from “solved” even for frontier agents. The detailed analysis in Figure 8 further highlights that long-horizon composition is a key bottleneck: when a workflow contains more than two tasks, the pass rate drops sharply—GPT 5.1 Pro decreases from 44.3% (workflows with ≤ 2 tasks) to 23.5% (workflows with > 2 tasks), and Claude Sonnet 4.5 decreases from 30.3% to 11.8%. This indicates that error accumulation across steps and missing intermediate affordances disproportionately hurt multi-step execution.

Pass rate also varies substantially by task type (Figure 8). Data Entry / Import and Structuring / Formatting are consistently among the most challenging categories, which aligns with FINCH spreadsheets exhibiting messy layouts, irregular tables, and nontrivial structural constraints. Moreover, Data Entry / Import workflows are frequently entangled with web search or PDF parsing, introducing multimodal dependencies that amplify failure modes. Notably, Translation—a task where modern LLMs typically excel in standard NLP settings—performs surprisingly poorly in FINCH. In finance-heavy tables, translation can easily distort or drop critical structure and layout cues (e.g., header hierarchies, row/column alignment), and large grids make omissions more likely, leading to systematic failures. Detailed error analysis can be found in Section 3.3.

# Tasks per workflow	# Workflows	Pass Rate (%)	Avg. time (min)
1	37	48.6	13.1
2	84	42.4	17.4
3	33	33.3	18.7
4	10	0	21.3
5	8	12.5	13.6

Table 2: Average GPT 5.1 Pro completion time across workflows with different numbers of tasks.

We further analyze how GPT 5.1 web completion time scales with the number of tasks in a workflow (Table 2). The longest individual workflow run takes roughly 60 minutes to complete—but fails, highlighting how challenging workflows can be for current agents. On average, single-task workflows take 13.1 minutes, while workflows with two, three, and four tasks require 17.4, 18.7, and 21.3 minutes, respectively, reflecting the increased compositional complexity of multi-task settings. Interestingly, five-task workflows show a lower average time (13.6 minutes), most of which involve web search and almost all of which result in failure.

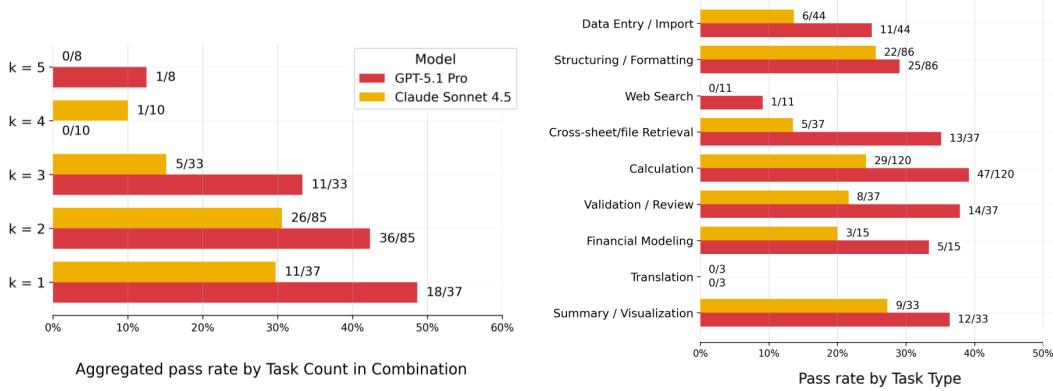


Figure 8: Pass rate comparison for GPT 5.1 Pro and Claude Sonnet 4.5 across different task combinations and task types. The left chart visualizes the aggregated pass rate based on task combinations, revealing the models’ capabilities in handling multi-step workflows commonly seen in professional finance and accounting tasks. The k in the right chart represents the number of tasks included in a workflow. For example, a “ $k=3$ ” workflow involves three distinct tasks, and its pass rate is calculated based on the collective performance of those tasks. The right chart shows the pass rates for individual tasks performed by both models in the FINCH benchmark. For workflows that involve multiple tasks (i.e., composite workflows), the correctness is counted for each individual task, as it is difficult to isolate which specific task caused the failure.

API-based. Figure 2 further shows that our API-based method remains competitive despite using only a single LLM call. In automated evaluation, for example, GPT 5.1 Pro achieves 41.9%, while GPT 5.1 with our API-based agent design reaches 32.0%—a relatively small gap given the much weaker interaction budget. Our agent design narrows the gap by supplying more efficient spreadsheet representations and task-appropriate tool outputs within the single-call budget, thereby reducing brittle reasoning over complex layouts and mitigating cascading mistakes.

3.2.1 Consistency Between Human and Automated Evaluation

We adopt a lightweight multimodal model, GPT-5-mini, as the judge to validate the performance of our automated evaluation framework. As shown in Table 3, the automated evaluation largely aligns with human judgments: for GPT 5.1 Pro and Claude Sonnet 4.5, the judge agrees with human labels on 82.1% and 90.2% of workflows, respectively. The judge also achieves high recall (83.3% and 88.4%), meaning it recovers most human-labeled passes, and reasonably strong precision (73.6% and 76.0%), indicating that the majority of automatically predicted passes are also accepted by human evaluators. Overall, this suggests that automated evaluation may overestimate accuracy by several percentage points.

Table 3: Comparison of human and automated evaluation on GPT and Claude product-side agents. “Automated Eval” shows pass counts/rates under the LLM-as-judge framework. “Agreement w/ Human Eval” reports how well the automated judgments match human labels: accuracy (Acc), recall (human-pass recall), and precision (human-pass precision).

Model (Product)	Automated Eval		Human Eval		Agreement w/ Human Eval		
	Pass	Pass Rate (%)	Pass	Pass Rate (%)	Acc (%)	Recall (%)	Precision (%)
GPT 5.1 Pro	72/172	41.9	66/172	38.4	82.1	83.3	73.6
Claude Sonnet 4.5	50/172	29.1	43/172	25.0	90.2	88.4	76.0

On the model side, the LLM judge can occasionally miss nuances in the rubric—either failing to catch subtle visual or numerical errors in large spreadsheets or, conversely, being overly literal about certain instructions (e.g., penalizing benign formula-to-constant conversions). However, we also observe cases where the LLM-based judge is correct but human raters are wrong—for example, when formulas are silently replaced with static values, which are difficult to detect through GUI-based inspection alone. On the system side, limitations of our spreadsheet tooling and data pipeline (e.g., incomplete support for corrupted but human-readable workbooks or uncommon file formats) can

cause valid outputs to be marked as failures. Taken together, these factors mean that our automated scores should be interpreted as approximate rather than exact, and that human review remains important for borderline or high-impact workflows.

3.3 Error Analysis

To understand the sources of failure on FINCH, we conducted a qualitative error analysis of GPT 5.1 Pro and Claude Sonnet 4.5 in both their web-agent and API configurations. For all failed workflows in our evaluation, we manually inspected the trajectories and annotated the primary cause of failure.

From a workflow-centric perspective, we identify five dominant categories of error.² Take Claude Sonnet 4.5 web-agent as an example. Across all examined failures, 10% stem from *task misunderstanding*: enterprise tasks often rely on implicit context in enterprise artifacts (e.g., spreadsheets), which models frequently overlook, leading them to misinterpret what is being asked and the required deliverable. 25% are *data retrieval errors*, including selecting the wrong cross-sheet, cross-table, or intra-table row/column ranges. 35% arise from *formula reasoning errors*, such as failing to reconstruct the latent business logic encoded in formulas or deriving incorrect new formulas. 25% are due to *code generation errors*, where generated scripts (e.g., Python with spreadsheet APIs) are syntactically invalid or misaligned with the spreadsheet layout. The remaining 5% correspond to *data rendering errors*, including incorrect formatting, misconfigured charts, or flawed final reports that deviate from the requested layout or narrative—for example, creating a brand-new spreadsheet instead of modifying the original one as requested. We also compare error patterns between web-based agents and API-based setups, with details provided in Appendix C.

Notably, all of these error types correspond to *generic capabilities* that modern LLMs already appear to master on many existing benchmarks. The question, then, is why these ostensibly strong base abilities degrade so sharply on FINCH. Our analysis points to five intertwined properties of real-world enterprise Finance & Accounting workflows that make failures more likely and more catastrophic. First, FINCH workflows routinely involve *large, fragmented spreadsheet ecosystems*: dozens of interlinked workbooks and thousands of rows distributed across many sheets. Executing these workflows accurately requires long-range cross-sheet navigation and precise referencing, which substantially increases the likelihood of small retrieval errors. Second, the *content is dense and semantically homogeneous*: many cells contain domain-specific financial concepts that are subtly different yet lexically similar (e.g., variants of revenue/expense items, adjusted vs. unadjusted metrics), making entity disambiguation and cell grounding unusually difficult. Third, the *table layouts and structures are complex and often irregular*, including multi-level headers, merged cells, nested subtotals, and bespoke layouts that force the model to infer structure from noisy contents and ad hoc formatting. For example, at the code level, even tiny misinterpretations of these layouts (e.g., off-by-one errors when specifying ranges) can then propagate into globally incorrect outputs, especially when logic is applied in batch across many such sheets. Fourth, *formulas encode latent structure and logic*. In the FINCH dataset, each sheet contains a large number of formulas that encode latent business logic, temporal assumptions, and fine-grained dependencies that are not visible from displayed values alone; yet models typically prioritize cell values and under-use formulas, leading to systematic misinterpretations. For example, in a pricing sheet with the column header `IF NGPL MidContinent index (@ Baker)`, the apparent semantics from the header alone suggest a daily exposure metric. However, inspecting the associated formula ($25 * V21 + C41 * C22$) reveals that the column in fact encodes a 55-day payment timing. Models that ignore or under-utilize formulas systematically misinterpret such columns' roles in downstream calculations, and this misinterpretation then propagates through subsequent steps. Finally, many workflows involve *multimodal artifacts and chat-centric tasks* such as combining spreadsheets with PDFs, charts, and screenshots requiring the agent to jointly reason over heterogeneous formats. For example, tables embedded in PDFs are often only partially referenced, with key entries missing or truncated.

Many of these factors have been examined in prior work (e.g., multi-spreadsheet settings, complex table structures, formula reasoning, and multi-step workflows), and state-of-the-art models can perform reasonably well on benchmarks that emphasize a limited subset of these factors. In FINCH, however, these factors co-occur within the same workflow in real-world enterprise data, and our results suggest this COMPOSITION is what drives the sharp performance drop. FINCH does not

²For this study, once we identify the first clear error in a failed workflow, we stop further analysis for that workflow.

demand fundamentally new abilities; rather, it probes these abilities under an enterprise “extreme” regime of high complexity, noise, and long-horizon dependencies—closely mirroring real Finance & Accounting work. Progress on compositional capability, therefore, requires training and evaluation on long-running, computation- and reasoning-intensive workflows over large, messy multimodal enterprise artifacts.

4 Related Work

The integration of LLMs into enterprise productivity tools has accelerated dramatically recently. The recently launched ChatGPT Agent [41] extends these capabilities to autonomous task completion, enabling multi-step workflows across web browsing, code execution, and spreadsheet manipulation. Microsoft Copilot [39] embeds AI capabilities across the Microsoft 365 suite, enabling users to draft documents, analyze spreadsheets, and automate workflows through natural language interaction. Similarly, Google has integrated Gemini [21] into Google Workspace, providing AI-assisted features in Docs, Sheets, and Gmail. Anthropic’s Claude Excel has also entered the enterprise space with spreadsheet automation capabilities [2], while remarkable tools like Shortcut AI [1] focus specifically on AI-powered spreadsheet manipulation.

The emergence of agentic AI systems marks a significant shift from understanding and QA to autonomous task completion [30, 7, 37]. However, there are long-standing challenges such as messy inputs and multimodal processing. SpreadsheetLLM [17] introduces novel encoding and compression methods to help LLMs understand large and messy spreadsheet structures, and further addresses this challenge through spreadsheet post-training. Beyond structural understanding, multimodal processing remains challenging for spreadsheet AI systems. On the formatting front, early work explored neural approaches for table formatting [15]. Recent advances in formula generation have progressed from pretraining with numerical reasoning [9] to natural language-driven formula synthesis [54], contrastive learning-based recommendation [6], and interactive formula prediction through hierarchical expansion [23].

Recent years have seen significant progress in benchmarks for financial reasoning [8, 27, 5, 51, 35, 10, 44, 29, 36, 42, 25, 53], spreadsheet reasoning [30, 7, 37, 26, 33, 13, 56, 17, 48, 34, 16, 12, 31, 54, 43, 45], and multimodal document [38], table [32, 55], chart [46], table-chart [14], and spreadsheet [50, 13] reasoning, driving advances in LLM-based agents for enterprise tasks. However, FINCH proposes a new benchmark for the messy artifacts and long-horizon workflows in wild F&A enterprise settings.

5 Conclusion

We introduced FINCH, a new benchmark for real-world F&A enterprise workflows. FINCH combines workflows induced from enterprise email threads, version histories of spreadsheets, and high-quality financial artifacts with rigorous expert annotation and a calibrated LLM-as-judge framework, enabling systematic evaluation of agents on diverse workflows that operate over large, messy, and multimodal enterprise artifacts and require long-horizon, spreadsheet-centric reasoning. Our experiments show that even the strongest frontier systems pass fewer than 40% of workflows after spending 48 hours on the benchmark, revealing a substantial gap between current AI capabilities and the demands of real enterprise practice. We hope FINCH will serve as a foundation for developing agents to tackle real, messy and long-horizon professional work.

References

- [1] Shortcut AI. Shortcut ai for spreadsheets. <https://www.tryshortcut.ai/>, 2024.
- [2] Anthropic. Claude for excel. <https://claude.com/claude-for-excel>, 2025.
- [3] Anthropic. Introducing claude sonnet 4.5. <https://www.anthropic.com/news/claude-sonnet-4-5>, 2025. Accessed: 2025-12-14.
- [4] World Bank. *International Debt Report 2024*. World Bank, Washington, DC, 2024. World Bank’s annual publication on external debt statistics.
- [5] Antoine Bigeard, Langston Nashold, Rayan Krishnan, and Shirley Wu. Finance agent benchmark: Benchmarking llms on real-world financial research tasks. *arXiv preprint arXiv:2508.00828*, 2025.
- [6] Sibei Chen, Yeye He, Weiwei Cui, Ju Fan, Song Ge, Haidong Zhang, Dongmei Zhang, and Surajit Chaudhuri. Auto-formula: Recommend formulas in spreadsheets using contrastive learning for table representations. *Proceedings of the ACM on Management of Data*, 2(3):1–27, 2024.
- [7] Yibin Chen, Yifu Yuan, Zeyu Zhang, Yan Zheng, Jinyi Liu, Fei Ni, Jianye Hao, Hangyu Mao, and Fuzheng Zhang. 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, 2025.
- [8] Zhiyu Chen, Wenhua Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, et al. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, 2021.
- [9] Zhoujun Cheng, Haoyu Dong, Ran Jia, Pengfei Wu, Shi Han, Fan Cheng, and Dongmei Zhang. Fortap: Using formulas for numerical-reasoning-aware table pretraining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1150–1166, 2022.
- [10] Chanyeol Choi, Jihoon Kwon, Alejandro Lopez-Lira, Chaewoon Kim, Minjae Kim, Juneha Hwang, Jaeseon Ha, Hojun Choi, Suyeol Yun, Yongjin Kim, et al. Finagentbench: A benchmark dataset for agentic retrieval in financial question answering. In *Proceedings of the 6th ACM International Conference on AI in Finance*, pages 632–637, 2025.
- [11] Department of Finance Canada. Fiscal reference tables, november 2025. Technical report, Government of Canada, Ottawa, Canada, 2025. Provides annual data on the financial position of the federal, provincial-territorial and local governments.
- [12] Haoyu Dong, Yue Hu, and Yanan Cao. Reasoning and retrieval for complex semi-structured tables via reinforced relational data transformation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1382–1391, 2025.
- [13] Haoyu Dong, Shijie Liu, Shi Han, Zhouyu Fu, and Dongmei Zhang. Tablesense: Spreadsheet table detection with convolutional neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 69–76, 2019.
- [14] Haoyu Dong, Haochen Wang, Anda Zhou, and Yue Hu. Ttc-quali: A text-table-chart dataset for multimodal quantity alignment. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 181–189, 2024.
- [15] Haoyu Dong, Jinyu Wang, Zhouyu Fu, Shi Han, and Dongmei Zhang. Neural formatting for spreadsheet tables. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 305–314, 2020.
- [16] Haoyu Dong, Pengkun Zhang, Mingzhe Lu, Yanzhen Shen, and Guolin Ke. Machinelearninglm: Scaling many-shot in-context learning via continued pretraining. *arXiv preprint arXiv:2509.06806*, 2025.

- [17] Haoyu Dong, Jianbo Zhao, Yuzhang Tian, Junyu Xiong, Shiyu Xia, Mengyu Zhou, Yun Lin, José Cambronero, Yeye He, Shi Han, et al. Spreadsheetllm: encoding spreadsheets for large language models. *arXiv preprint arXiv:2407.09025*, 2024.
- [18] Alex Egg, Martin Iglesias Goyanes, Friso Kingma, Andreu Mora, Leandro von Werra, and Thomas Wolf. Dabstep: Data agent benchmark for multi-step reasoning. *arXiv preprint arXiv:2506.23719*, 2025.
- [19] EnronData.org. Edo enron email pst dataset. <https://enrondata.readthedocs.io/en/latest/data/edo-enron-email-pst-dataset/>. Creative Commons Attribution 3.0 United States License. To provide attribution, please cite to “EnronData.org.”.
- [20] Marc Fisher and Gregg Rothermel. 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, 2005.
- [21] Google. Gemini for google workspace. <https://workspace.google.com/solutions/ai/>, 2024.
- [22] Google DeepMind. Gemini 3 pro. <https://deepmind.google/models/gemini/pro/>, 2025. Accessed: 2025-12-14.
- [23] Wanrong He, Haoyu Dong, Yihuai Gao, Zhichao Fan, Xingzhuo Guo, Zhitao Hou, Xiao Lv, Ran Jia, Shi Han, and Dongmei Zhang. Hermes: Interactive spreadsheet formula prediction via hierarchical formulæt expansion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8356–8372, 2023.
- [24] HM Treasury. Public expenditure statistical analyses 2023. Technical report, HM Treasury, London, United Kingdom, 2023. UK public expenditure statistical release (PESA).
- [25] Liang Hu, Jianpeng Jiao, Jiashuo Liu, Yanle Ren, Zhoufutu Wen, Kaiyuan Zhang, Xuanliang Zhang, Xiang Gao, Tianci He, Fei Hu, et al. Finsearchcomp: Towards a realistic, expert-level evaluation of financial search and reasoning. *arXiv preprint arXiv:2509.13160*, 2025.
- [26] Amila Indika and Igor Molybog. Sodbench: A large language model approach to documenting spreadsheet operations. *arXiv preprint arXiv:2510.19864*, 2025.
- [27] Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. Financebench: A new benchmark for financial question answering. *arXiv preprint arXiv:2311.11944*, 2023.
- [28] Bryan Klimt and Yiming Yang. The enron corpus: A new dataset for email classification research. In *European conference on machine learning*, pages 217–226. Springer, 2004.
- [29] Haohang Li, Yupeng Cao, Yangyang Yu, Shashidhar Reddy Javaji, Zhiyang Deng, Yueru He, Yuechen Jiang, Zining Zhu, Kp Subbalakshmi, Jimin Huang, et al. Investorbench: A benchmark for financial decision-making tasks with llm-based agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2509–2525, 2025.
- [30] Hongxin Li, Jingran Su, Yuntao Chen, Qing Li, and Zhao-Xiang Zhang. Sheetcopilot: Bringing software productivity to the next level through large language models. *Advances in Neural Information Processing Systems*, 36:4952–4984, 2023.
- [31] Jinyang Li, Nan Huo, Yan Gao, Jiayi Shi, Yingxiu Zhao, Ge Qu, Yurong Wu, Chenhao Ma, Jian-Guang Lou, and Reynold Cheng. Tapilot-crossing: Benchmarking and evolving llms towards interactive data analysis agents. *arXiv preprint arXiv:2403.05307*, 2024.
- [32] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. Tablebank: Table benchmark for image-based table detection and recognition. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1918–1925, 2020.
- [33] Peng Li, Yeye He, Cong Yan, Yue Wang, and Surajit Chaudhuri. Auto-tables: Synthesizing multi-step transformations to relationalize tables without using examples. *Proceedings of the VLDB Endowment*, 16(11):3391–3403, 2023.

- [34] Zheng Li, Yang Du, Mao Zheng, and Mingyang Song. Mimutable: A multi-scale spreadsheet benchmark with meta operations for table reasoning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2548–2560, 2025.
- [35] Shu Liu, Shangqing Zhao, Chenghao Jia, Xinlin Zhuang, Zhaoguang Long, Jie Zhou, Aimin Zhou, Man Lan, and Yang Chong. Findabench: Benchmarking financial data analysis ability of large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 710–725, 2025.
- [36] Zhaowei Liu, Xin Guo, Haotian Xia, Lingfeng Zeng, Fangqi Lou, Jinyi Niu, Mengping Li, Qi Qi, Jiahuan Li, Wei Zhang, et al. Visfineval: A scenario-driven chinese multimodal benchmark for holistic financial understanding. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24099–24157, 2025.
- [37] 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. *Advances in Neural Information Processing Systems*, 37:94871–94908, 2024.
- [38] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021.
- [39] Microsoft. Microsoft 365 copilot. <https://www.microsoft.com/en-us/microsoft-365/copilot>, 2024.
- [40] OpenAI. Gpt-5. <https://openai.com/index/introducing-gpt-5/>, 2025. Accessed: 2025-12-14.
- [41] OpenAI. Introducing chatgpt agent. <https://openai.com/index/introducing-chatgpt-agent/>, 2025.
- [42] Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang, Olivia Watkins, Simón Posada Fishman, Marwan Aljubeh, Phoebe Thacker, Laurance Fauconnet, et al. Gdpval: Evaluating ai model performance on real-world economically valuable tasks. *arXiv preprint arXiv:2510.04374*, 2025.
- [43] Weixuan Wang, Dongge Han, Daniel Madrigal Diaz, Jin Xu, Victor Rühle, and Saravan Rajmohan. Odysseybench: Evaluating llm agents on long-horizon complex office application workflows. *arXiv preprint arXiv:2508.09124*, 2025.
- [44] Yan Wang, Keyi Wang, Shanshan Yang, Jaisal Patel, Jeff Zhao, Fengran Mo, Xueqing Peng, Lingfei Qian, Jimin Huang, Guojun Xiong, et al. Finauditing: A financial taxonomy-structured multi-document benchmark for evaluating llms. *arXiv preprint arXiv:2510.08886*, 2025.
- [45] Zilong Wang, Yuedong Cui, Li Zhong, Zimin Zhang, Da Yin, Bill Yuchen Lin, and Jingbo Shang. Officebench: Benchmarking language agents across multiple applications for office automation. *arXiv preprint arXiv:2407.19056*, 2024.
- [46] Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 37:113569–113697, 2024.
- [47] Ryan Wong, Jiawei Wang, Junjie Zhao, Li Chen, Yan Gao, Long Zhang, Xuan Zhou, Zuo Wang, Kai Xiang, Ge Zhang, et al. Widesearch: Benchmarking agentic broad info-seeking. *arXiv preprint arXiv:2508.07999*, 2025.
- [48] Pengzuo Wu, Yuhang Yang, Guangcheng Zhu, Chao Ye, Hong Gu, Xu Lu, Ruixuan Xiao, Bowen Bao, Yijing He, Liangyu Zha, et al. Realhitbench: A comprehensive realistic hierarchical table benchmark for evaluating llm-based table analysis. *arXiv preprint arXiv:2506.13405*, 2025.
- [49] xAI. Grok 4. <https://x.ai/news/grok-4>, 2025.

- [50] Shiyu Xia, Junyu Xiong, Haoyu Dong, Jianbo Zhao, Yuzhang Tian, Mengyu Zhou, Yeye He, Shi Han, and Dongmei Zhang. Vision language models for spreadsheet understanding: Challenges and opportunities. In *Proceedings of the 3rd Workshop on Advances in Language and Vision Research (ALVR)*, pages 116–128, 2024.
- [51] Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, et al. Finben: A holistic financial benchmark for large language models. *Advances in Neural Information Processing Systems*, 37:95716–95743, 2024.
- [52] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengan Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025.
- [53] Zhihan Zhang, Yixin Cao, and Lizi Liao. Xfinbench: Benchmarking llms in complex financial problem solving and reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8715–8758, 2025.
- [54] Wei Zhao, Zhitao Hou, Siyuan Wu, Yan Gao, Haoyu Dong, Yao Wan, Hongyu Zhang, Yulei Sui, and Haidong Zhang. NI2formula: Generating spreadsheet formulas from natural language queries. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 2377–2388, 2024.
- [55] Mingyu Zheng, Xinwei Feng, Qingyi Si, Qiaoqiao She, Zheng Lin, Wenbin Jiang, and Weiping Wang. Multimodal table understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9102–9124, 2024.
- [56] Ruiyan Zhu, Xi Cheng, Ke Liu, Brian Zhu, Daniel Jin, Neeraj Parihar, Zhoutian Xu, and Oliver Gao. Sheetmind: An end-to-end llm-powered multi-agent framework for spreadsheet automation. *arXiv preprint arXiv:2506.12339*, 2025.

Contents

1	Introduction	2
2	FINCH: A Real-world Finance & Accounting Workflow Benchmark	3
2.1	Dataset Construction	3
2.1.1	Workflow from Enterprise Email Threads	4
2.1.2	Workflow Derivation from Versioned Spreadsheets	4
2.1.3	Workflow Sourced from High-quality Artifacts	4
2.1.4	Quality Control	5
2.2	Dataset Analysis	5
2.2.1	Task Compositionality	5
2.2.2	Messiness	6
2.2.3	Multimodality	6
2.3	Evaluation Method	6
2.3.1	Human Evaluation	6
2.3.2	LLM-as-Judge Evaluation	7
3	Experiments	8
3.1	Agents and Models	8
3.1.1	Product-side Agents	8
3.1.2	API-based Models	8
3.2	Experimental Results	9
3.2.1	Consistency Between Human and Automated Evaluation	10
3.3	Error Analysis	11
4	Related Work	12
5	Conclusion	12
A	Author List	19
B	Experiment Details	19
B.1	API-based Models	19
C	Detailed Analysis	20
D	Ethics Statement	20
E	Examples	21
E.1	Example 1	21
E.2	Example 2	22
E.3	Example 3	22

E.4 Example 4	23
E.5 Example 5	23
E.6 Example 6	24
E.7 Example 7	24
E.8 Example 8	25
E.9 Example 9	25
E.10 Example 10	26
E.11 Example 11	26

A Author List

- **Haoyu Dong*** (University of Chinese Academy of Sciences)
donghaoyu82@gmail.com
- **Pengkun Zhang** (South China University of Technology)
sezhangpengkun@mail.scut.edu.cn
- **Yan Gao** (Zhongguancun Academy)
gaoyan@zgcii.ac.cn
- **Xuanyu Dong** (Harvest Fund)
qianmuxuanyu@126.com
- **Yilin Cheng** (Fudan University, Zhongguancun Academy)
ylcheng@m.fudan.edu.cn
- **Mingzhe Lu** (University of Chinese Academy of Sciences)
lumingzhe23@mails.ucas.edu.cn
- **Adina Yakefu** (Hugging Face)
adina@huggingface.co
- **Shuxin Zheng** (Zhongguancun Academy)
sz@bjzgca.edu.cn

*Corresponding author.

B Experiment Details

B.1 API-based Models

Execution Paradigm. We frame the API evaluation as a **code generation task**. Models are instructed to solve spreadsheet manipulation and generation workflows by generating executable Python scripts, which are then executed in a sandboxed environment to produce output artifacts. This paradigm aligns with SpreadsheetBench’s philosophy of treating model-written code as the primary action space, but is adapted here to accommodate long-horizon, multimodal enterprise tasks.

- **Action Space:** Models generate Python code using standard libraries including `openpyxl` (for Excel manipulation), `pandas` (for data processing), `matplotlib` (for visualization), and `scikit-learn` (for statistical analysis).
- **Output Format:** Models must produce complete, self-contained Python scripts wrapped in markdown code blocks (““python ... ““).
- **Sandboxed Execution:** Generated code is extracted via regex parsing and executed in isolated Docker containers running Jupyter Kernel Gateway. Each container mounts the dataset volume at `/mnt/data/` with a 10-minute session timeout.
- **Single-shot Protocol:** We employ a strict one-shot generation protocol without iterative refinement—each model produces exactly one solution per workflow. If the generated code fails to execute (e.g., due to syntax errors or runtime exceptions), the workflow is marked as failed without retry. This strict setting is designed to evaluate the model’s raw code generation capability under realistic deployment constraints.

This unified code-as-action setting ensures that the measured performance reflects the model’s inherent competence on complex workflows rather than benefits derived from interactive debugging.

Prompting Strategy. We employ a **zero-shot** setting with a structured system prompt comprising:

1. A role definition: “You are an expert who can manipulate spreadsheets through Python code.”
2. A detailed description of the compact spreadsheet encoding format with illustrative examples.
3. The task instruction and explicit input/output file paths.
4. Library-specific best practices (e.g., `openpyxl` chart creation patterns) to mitigate common code errors.
5. An explicit directive to generate Python code as the final output.

This structured design explicitly guides models toward generating valid, context-aligned Python code, minimizing ambiguity in task interpretation. However, for models that support reasoning traces (GPT 5.1, Gemini 3 Pro), we request explicit reasoning via the `include_reasoning` API parameter, enabling us to capture the model’s internal deliberation process for subsequent qualitative error analysis. Temperature is set to 0.7 across all models.

C Detailed Analysis

Web agents (ChatGPT 5.1 Pro vs. Claude Sonnet 4.5). ChatGPT 5.1 Pro tends to decompose workflows into more, smaller steps, with explicit reasoning, tool calls, execution, and self-checking at each step. This leads to longer traces and noticeably higher latency, but also more opportunities for intermediate validation (e.g., sanity-checking partial results). However, the code it generates is often hidden behind tool abstractions, so our error attribution is limited to observed behavior and natural language reasoning rather than the exact implementation details. Claude Sonnet 4.5 typically uses fewer steps and produces more direct solutions. In visualization-heavy workflows, its generated charts are often both more accurate and more aesthetically polished than those produced by ChatGPT 5.1 Pro, leading to relatively fewer failures in the data visualization sub-tasks.

ChatGPT 5.1 Pro and Claude Sonnet 4.5 agents can explore Excel files through many API calls within a single workflow, but their encoding methods are not well-suited to spreadsheets with complex layouts and structures. Thanks to efficient encoding and appropriate tool use, the following single-call API-based method achieves a pass rate that is much closer to that of product-side agents.

API-based Our API-based runs are single-call: they leverage the models’ underlying reasoning capabilities but lack two crucial affordances that web agents exploit. (i) interleaved code execution with feedback, and (ii) explicit reflection based on intermediate tool outputs. As a result, the API agents must generate the entire plan, code, and outputs within a single LLM call. When their initial structural assumptions about a spreadsheet are slightly off, they have no mechanism to detect or correct the mistake, leading to a significantly higher error rate, particularly in categories related to schema understanding and table manipulation. It’s desirable for future work to explore agentic methods with multiple round of API calls.

D Ethics Statement

The FINCH benchmark is constructed entirely from existing, publicly available data sources. Concretely, our workflows are derived from (1) the Enron email corpora, including the parsed Enron email dataset on Kaggle (released under the CC0 Public Domain dedication) and the Enron Email Dataset from EnronData.org (licensed under CC BY 3.0 US); (2) the EUSES spreadsheet corpus and its modified variants (CC BY 4.0); and (3) a diverse collection of enterprise-like artifacts, including documents from investment and securities companies, the World Bank (CC BY 3.0), Canadian and British government websites (Open Government License), and public corpora such as WideSearch (MIT license) and DABStep (CC BY 4.0). We respect the original licenses of all upstream resources and only redistribute content within the terms they allow.

On top of these sources, we apply additional filtering, normalization, and expert annotation to organize spreadsheets and related documents into coherent workflows with task instructions, input files, and reference outputs. We do not introduce any new personally identifiable information. During curation, we remove obviously sensitive fields when they are not necessary for the task (e.g., personal contact information or signatures) and avoid annotating workflows whose successful completion would depend on sensitive personal attributes rather than business logic. The resulting FINCH dataset is released under the Creative Commons Attribution 3.0 United States license (CC BY 3.0 US), which permits broad reuse while requiring appropriate attribution.

The language in FINCH is primarily English, reflecting the dominant language of the underlying Enron and EUSES corpora and many of the public institutional sources. Because some artifacts originate from funds and securities institutions and from Canadian government materials, a small fraction of workflows include Chinese or French content.

E Examples

E.1 Example 1



E.2 Example 2

Table 5.1 Public sector expenditure on services by departmental group and function, 2024-25

Figure 1 Public and publicly guaranteed debt, by creditor and creditor type in 2023, including IMF credit

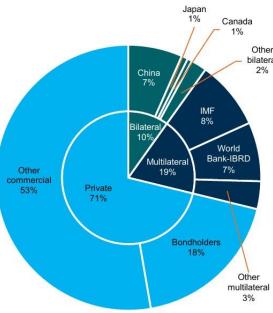


Figure 10: An example of extracting data from tables and charts in PDFs and saving it to a spreadsheet. AI agents must understand the layout, parse hierarchical structures, interpret how values map to specific cells, and reconstruct formulas from aggregated values.

E.3 Example 3

A	B	C	D	E	F	G	H	J	K	L	M	N	O	P
4						between								
5						Enron Corp. "Seller"								
6						and								
7						AEP Energy Services Gas Holding Company "Buyer"								
8						dated as of December 27, 2000								
9						Articles 2.4 (b) (e) and 2.5 (b), (e)								
10														
11														
12														
13														
14						MMBtu	Enron Amounts	AEP Amounts	Difference					
15						Rate								
16						Value								
17														
18														
19														
20														
21														
22														
23														
24														
25														
26														
27														
28														
29														
30														
31														
32														
33														
34														
35														
36														
37														
38														
39														
40						(1) Estimated Bammel Storage volume as of 05/31/01								
41						(2) Actual Bammel Storage volume as of 05/31/01								
42						(3) Published Price in Inside FERC (HSC) for the month of July 2001								
43						(4) Interest Payment calculated based on an assumed pay date of 10/31/01 for Daily Prime for 153 days (Enron calculate on "Additional								
44						Payment" and AEP calculated on "Working Gas Adjustment"								
<	Inv & WC Value Adj	Sched I - Cannon & TCO GLO	Sched II Lyondell	Sched III Centana	Prime Rate Calc - GROSS	Prime Rate Calc - NET (3)	+							

Figure 11: Cross-sheet reference validation. This example is relatively easy for frontier AI agents.

E.4 Example 4

A		B		F	
7		Base Case Variables			
8 Plant Capacity (PC)		263,000			
9 Discount Rate		5.00%			
10					
11 Question 1					
12 Include Interest Rate Adjustment?		No			
13 Include Apache Savings?		No			
14					
15 Question 2					
16 Include Interest Rate Adjustment?		Yes			
17 Include Apache Savings?		Yes			
18					
19 Question 3					
20 Include Interest Rate Modification?		Yes			
21 Include Apache Savings?		Yes			
22					
23 Question 4					
24 Include Interest Rate Adjustment?		No			
25 Include Apache Savings?		Yes			

A		B		F	
7		Base Case Variables			
8 Plant Capacity (PC)		263,000			
9 Discount Rate		5.00%			
10					
11 Question 1					
12 Include Interest Rate Adjustment?		No			
13 Include Apache Savings?		No			
14					
15 Question 2					
16 Include Interest Rate Adjustment?		Yes			
17 Include Apache Savings?		Yes			
18					
19 Question 3					
20 Include Interest Rate Modification?		Yes			
21 Include Apache Savings?		Yes			
22					
23 Question 4					
24 Include Interest Rate Adjustment?		No			
25 Include Apache Savings?		Yes			

A		B		F	
7		Base Case Variables			
8 Plant Capacity (PC)		263,000			
9 Discount Rate		5.00%			
10					
11 Question 1					
12 Include Interest Rate Adjustment?		No			
13 Include Apache Savings?		No			
14					
15 Question 2					
16 Include Interest Rate Adjustment?		Yes			
17 Include Apache Savings?		Yes			
18					
19 Question 3					
20 Include Interest Rate Modification?		Yes			
21 Include Apache Savings?		Yes			
22					
23 Question 4					
24 Include Interest Rate Adjustment?		No			
25 Include Apache Savings?		Yes			

A		B		F	
7		Base Case Variables			
8 Plant Capacity (PC)		263,000			
9 Discount Rate		5.00%			
10					
11 Question 1					
12 Include Interest Rate Adjustment?		No			
13 Include Apache Savings?		No			
14					
15 Question 2					
16 Include Interest Rate Adjustment?		Yes			
17 Include Apache Savings?		Yes			
18					
19 Question 3					
20 Include Interest Rate Modification?		Yes			
21 Include Apache Savings?		Yes			
22					
23 Question 4					
24 Include Interest Rate Adjustment?		No			
25 Include Apache Savings?		Yes			

A		B		F	
7		Base Case Variables			
8 Plant Capacity (PC)		263,000			
9 Discount Rate		5.00%			
10					
11 Question 1					
12 Include Interest Rate Adjustment?		No			
13 Include Apache Savings?		No			
14					
15 Question 2					
16 Include Interest Rate Adjustment?		Yes			
17 Include Apache Savings?		Yes			
18					
19 Question 3					
20 Include Interest Rate Modification?		Yes			
21 Include Apache Savings?		Yes			
22					
23 Question 4					
24 Include Interest Rate Adjustment?		No			
25 Include Apache Savings?		Yes			

A		B		F	
7		Base Case Variables			
8 Plant Capacity (PC)		263,000			
9 Discount Rate		5.00%			
10					
11 Question 1					
12 Include Interest Rate Adjustment?		No			
13 Include Apache Savings?		No			
14					
15 Question 2					
16 Include Interest Rate Adjustment?		Yes			
17 Include Apache Savings?		Yes			
18					
19 Question 3					
20 Include Interest Rate Modification?		Yes			
21 Include Apache Savings?		Yes			
22					
23 Question 4					
24 Include Interest Rate Adjustment?		No			
25 Include Apache Savings?		Yes			

A		B		F	
7		Base Case Variables			
8 Plant Capacity (PC)		263,000			
9 Discount Rate		5.00%			
10					
11 Question 1					

E.6 Example 6

Exposition liée aux comptes commerciaux et souverains

	31 mars 2024			31 mars 2023		
(en millions de dollars canadiens)	Prêts commerciaux	Prêts souverains	Total commerciaux	Prêts commerciaux	Prêts souverains	Total
Prêts						
Concessional - CUEC	8 508	-	8 508	40 153	-	40 153
Concessional	11	422	433	11	455	466
Non concessionnel	425	16 992	17 417	130	16 252	16 382
Total	8 944	17 414	26 358	40 294	16 707	57 001
Engagements de financement et passifs éventuels						
Engagements de prêts	941	2 273	3 214	1 007	3 039	4 046
Garanties de prêts	841	20 773	21 414	1 007	14 539	15 546
Total	9 885	38 187	5	40 072	41 301	5
Pourcentage	21%	79%	100%	57%	43%	100%

Commercial and Sovereign Exposure

	Mar 2024			Mar 2023		
(en millions de Canadian dollars)	Commercial	Sovereign	Total	Commercial	Sovereign	Total
Loans receivable						
Concessional - CEBA	\$ 508	-	\$ 508	40 153	-	40 153
Concessional	11	422	433	11	455	466
Non-concessional	425	16 992	17 417	130	16 252	16 382
Total	8,944	17,414	26,358	40,294	16,707	57,001
Financing commitments and contingent liabilities						
Loan commitments	941	2 273	3 214	1 007	3 039	4 046
Loan guarantees	941	20 773	21 414	1 007	14 539	15 546
Total	\$9,885	\$38,187	5	\$40,072	\$41,301	5
Percentage	21%	79%	100%	57%	43%	100%

Exposition liée aux comptes commerciaux et souverains, par industrie et par pays

	31 mars 2024			31 mars 2023		
(en millions de dollars canadiens)	Total	%	Total	%	Total	%
Prêts commerciaux :						
CUEC (diverses industries)	8 508	18	40 153	55		
Services publics	3	1 000	2			
Fabrication	169	-	-			
Information	39	-	35	-		
Autres	38	-	113	-		
Total	9 885	21	41 301	57		
Prêts souverains :						
Canada	37 670	78	30 670	42		
Chine	255	1	279	1		
Turkey	68	-	72	-		
Moroc	51	-	54	-		
Iraq	46	-	58	-		
Inde	31	-	32	-		
Autres	66	-	81	-		
Total	38 187	79	31 246	43		
Total	48 072	100	72 547	5	100	5

La baisse de l'exposition relative aux comptes commerciaux s'explique surtout par la diminution des prêts du CUEC. Quant à l'exposition relative aux prêts souverains, elle a augmenté en raison surtout de l'augmentation des garanties de prêts pour l'oléoduc Trans Mountain.

Figure 14: A workflow that translates a French report into English while preserving its format and structure. The report contains many tables to translate, along with text, notes, and even charts.

E.7 Example 7

TRANSWESTERN PIPELINE - SUMMARY OF OBA BALANCES

Operator	Dollars	Volume	Date	Imbal Type	Mktg Rep	MS rep
	PNM	\$879,575	422,873	2/19	Dollar Valued	
Conoco	\$484,069	223,110	2/19	Dollar Valued		
Mojave Pipeline	\$360,739	173,432	2/19	Volumetric		
OneOK Westex-Ward	\$328,563	157,963	2/18	Volumetric		
Newmorne	\$326,518	156,980	2/18	Dollar Valued		
NGPL	\$186,353	89,593	2/19	Volumetric		
Longhorn Gas Ventures	\$186,353	81,376	2/19	Dollar Valued		
Americo Abo	\$167,694	89,579	2/19	Dollar Valued		
SoCal	\$168,015	79,815	2/19	Volumetric		
El Paso Field Services	\$143,091	68,750	2/19	Dollar Valued		
Rod Cedar	\$129,691	62,053	2/18	Volumetric		
Agave	\$108,157	51,599	2/19	Dollar Valued		
Amarillo Nat Gas	\$102,894	49,372	2/17	Dollar Valued		
Citizens-Griffith	\$96,354	46,339	2/19	Dollar Valued		
Lonestar	\$87,495	42,065	2/18	Volumetric		
PG&E Topock	\$87,416	42,027	2/19	Volumetric		
Pains Gas Farmers Co-Op	\$63,242	30,405	1/31	Dollar Valued		
Capline	\$50,583	24,319	2/18	Dollar Valued		
Panhende Eastern	\$49,402	23,751	2/18	Volumetric		
Statland Exploration	\$48,490	23,313	1/31	Dollar Valued		
El Paso	\$48,144	23,343	2/18	Volumetric		
Continental	\$46,769	22,485	2/18	Dollar Valued		
Receivable imbalances	\$4,257,409	2,046,730				
Operator	Dollars	Volume	Date	Imbal Type	MS rep	MS rep
Others	(\$593,447)	(270,889)	2/17	Dollar Valued		
North Star - Westex	(\$209,759)	(100,701)	2/18	Dollar Valued		
Midland Richardson Gas Treating	(\$192,280)	(92,445)	1/31	Dollar Valued		
CrossTex Energy Serv	(\$134,414)	(64,022)	2/17	Dollar Valued		
Duke Energy Field Services	(\$128,990)	(62,014)	2/17	Dollar Valued		
Burlington	(\$56,993)	(27,229)	2/18	Dollar Valued		
SW Gas Transmission	(\$27,823)	(13,379)	2/18	Dollar Valued		
ACI	(\$1,000)	(500)	2/18	Dollar Valued		

TRANSWESTERN PIPELINE - SUMMARY OF OBA BALANCES

Operator	Dollars	Volume	Date	Imbal Type	Prod Month	Accum Prod	As of
		\$ Value	Equivalent	Mo Volume	Date		
Positive=due Transwestern							
Negative = due operator							
Williams	\$2,09						
SoCal	\$2,08						
NTXPH	\$2,08						
DOLLAR VALUED IMBALANCES							
West of Thoreau							
Capline	\$50,583						
North Star Steel	(\$269,783)						
Citizens Communications	(\$563,447)						
Total WOT	(\$714,091)						
West Juan							
TransColorado	(\$374)						
Williams Field Services	(\$18,950)						
Burlington	(\$56,909)						
Total SJ	(\$76,234)						
VOLUMETRIC IMBALANCES							
West of Thoreau							
Midvale Pipeline	173,432						
SoCal	79,815						
El Paso - Window Rock	64,269						
PGE Topock	42,027						

Figure 15: Transforming a table from one structure to another requires reorganizing data and retrieving information across sheets (e.g., area_info and summary). This example poses additional challenges: (1) distinguishing value-driven operators from volume-driven operators, and (2) performing aggregation and validation over the reorganized data.

E.8 Example 8

	C	D	E	F	G	H	I	
	IF NGPL MidContinent index (@ Forgan)	IF NGPL MidContinent index (@ Baker)	IF NGPL MidContinent Index minus \$0.01	IF CIG Rocky Mtns. index minus \$0.03	Gas Daily PG&E Topock index minus \$0.02	Gas Daily EI Paso- San Juan index minus \$0.10	Gas Daily NWPL Wyoming Pool index minus \$0.10	
31	Dec-01	155,147	155,147	-	1,387,125	898,812	480,067	431,317
31	Jan-02	370,351	370,351	-	3,536,910	2,117,920	1,149,369	1,058,919
28	Feb-02	401,692	401,692	-	4,112,550	2,257,575	1,258,026	1,180,826
31	Mar-02	380,019	380,019	-	3,880,530	2,119,363	1,199,536	1,109,736
30	Apr-02	392,883	392,883	-	3,843,585	2,167,842	1,235,554	1,128,504
31	May-02	386,271	386,271	-	3,596,625	2,121,738	1,200,934	1,088,184
30	Jun-02	406,844	406,844	1,702,339	2,057,625	2,295,482	684,037	620,487
31	Jul-02	415,605	415,605	3,807,843	-	2,441,730	-	-
31	Aug-02	437,663	437,663	3,995,353	-	2,606,812	-	-
30	Sep-02	445,239	445,239	4,059,253	-	2,615,090	-	-
31	Oct-02	444,413	444,413	4,047,111	-	2,548,440	-	-

Figure 16: The apparent semantics from the headers suggest a monthly/daily exposure metric. However, inspecting the underlying formula (e.g., C5=\$A4*Volumes!B6*Curves!G7+25*Volumes!B7*Curves!G8) reveals that it actually encodes a 55-day payment timing schedule. Models that ignore or underutilize formula information, therefore systematically misattribute the column's role in downstream computations, and this misinterpretation then propagates through subsequent steps.

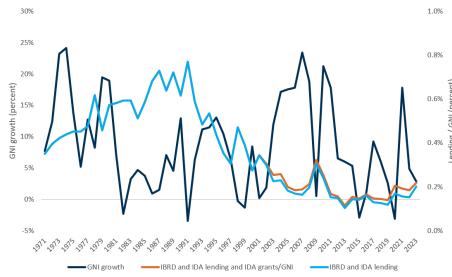
E.9 Example 9

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1 SUMMARY OF CANADA'S TRADING INCOME BY TRADER - IN US\$															
3 FX - AVG OF MNTH	1.4511	1.4493	1.4607	1.4677	1.4943	1.4761	1.4778	1.4822	1.4833	1.5108	1.5429	1.5396	To Dec 6	1.48	
4	Jan-2000	Feb-2000	Mar-2000	Apr-2000	May-2000	Jun-2000	Jul-2000	Aug-2000	Sep-2000	Oct-2000	Nov-2000	Dec-2000	MTD		
5															
6															
7	BY BOOK:														
8	TERM:														
9	Alberta Term	3,458,625	5,547,777	2,686,484	469,449	4,631,088	(1,097,125)	(5,566,310)	5,913,482	777,220	(3,599,231)	(7,518,121)	2,503,740	8,203,978	
10	BC Term	(879,015)	8,011	902,041	710,899	-	-	-	-	-	126,089	(5,525,759)	\$7,648,004	2,990,260	
11	EOL - Term	-	-	-	7,208	936,819	(1,582,742)	(512,592)	471,676	2,538,389	1,319,205	(298,040)	2,011,553	4,891,486	
12	Options	(224,086)	685,015	(711,171)	=+Summary!E10/(SUM-USDI!\$33	(1,140,535)	(2,145,933)	(5,398,385)	(1,050,467)	2,645,718	(1,111,159)	(2,863,509)			
13	CASH														
14	Alberta Cash	(1,649,853)	692,072	785,943	703,120	2,078,946	2,115,658	1,174,214	1,353,510	503,658	743,525	766,044	(162,227)	9,102,609	
15	BC Cash	(143,902)	176,401	(519,854)	50,999	73,039	317,584	295,608	327,887	673,449	522,897	341,336	238,176	2,853,619	
16	BC Pipe Cash	-	-	-	-	-	-	-	-	-	-	(385,732)	3,523,563	140,464	
17	Alberta Term - GD	(242,131)	300,898	30,625	88,895	398	(142)	150	(365,403)	(962,508)	124,583	(560,897)	(4,123)	(1,589,654)	
18	BC Term - GD	(615,175)	21,265	(4)	-	-	-	-	-	-	-	-	-	(593,915)	
19	Options - GD	286,232	517,629	348,138	424,851	354,022	1,007,495	853,328	555,567	1,935,097	(175,224)	(1,554,150)	2,667,285	7,220,271	
20	Power	246,906	183,840	1,261,789	1,453,448	1,254,449	(998,132)	(416,028)	(1,002,687)	1,471,034	(136,978)	47,101	-	3,364,742	
21	PMA	-	-	-	-	-	(812,960)	-	-	-	-	-	-	(812,960)	
22	TOTAL CANADA	237,501	8,129,908	5,283,991	4,248,065	13,745,994	278,691	(5,312,165)	5,108,098	1,037,955	(2,511,333)	(8,133,204)	13,931,724	36,045,223	
23															
24															
25	BY RISK TYPE:														
26	Total Term	2,355,424	6,237,804	2,877,354	1,526,752	9,987,140	(1,350,812)	(7,219,438)	4,239,225	(2,582,776)	(3,204,404)	(10,696,202)	11,052,148	13,222,216	
27	Check	-	-	-	-	-	-	-	-	-	-	-	-		
28	Total Cash	(2,117,923)	1,892,104	2,406,637	2,721,313	3,758,853	1,629,502	1,907,272	868,873	3,620,730	693,071	2,562,998	2,879,576	22,823,007	
29	Check	-	-	-	-	-	-	-	-	-	-	-	-		
30	TOTAL CANADA	237,501	8,129,908	5,283,991	4,248,065	13,745,994	278,691	(5,312,165)	5,108,098	1,037,955	(2,511,333)	(8,133,204)	13,931,724	36,045,223	
31															
32															
33	BY AREA/TRADER:														
34	West Term - Lavorato	3,216,394	5,845,675	2,717,109	-	-	-	-	-	-	-	-	11,779,178		
35	West Term - McKay	(1,638,092)	205,677	902,037	1,269,232	4,631,487	(1,097,266)	(5,566,160)	471,676	2,538,389	1,059,563	(2,300,237)	9,800,031	10,276,335	
36	West Term - Lambie	-	-	-	7,208	936,819	(1,582,742)	(512,592)	5,548,079	(185,288)	(3,474,648)	(8,079,018)	2,499,617	(4,842,564)	
37	Options - Disturnal	62,146	1,202,645	(363,033)	764,058	4,773,255	2,336,550	(287,207)	(1,590,366)	(3,963,287)	(1,225,692)	1,091,564	1,556,126	4,356,762	
38	Alberta Cash - Cowan	(1,649,853)	692,072	785,943	703,120	2,078,946	1,302,698	1,174,214	1,353,510	503,658	743,525	766,044	(162,227)	8,289,648	
39	BC Cash - Clark	-	(19,854)	50,999	73,039	-	317,584	295,608	327,887	673,449	522,897	341,336	238,176	2,821,121	
40	Power - Greenizan	248,906	183,840	1,261,789	1,453,448	1,254,449	(998,132)	(416,028)	(1,002,687)	1,471,034	(136,978)	47,101	-	3,364,742	
41	TOTAL CANADA	(2,978,893)	2,284,233	2,566,882	4,248,065	13,745,994	278,691	(5,312,165)	5,108,098	1,037,955	(2,511,333)	(8,133,204)	13,931,724	36,045,223	

Figure 17: This workflow requires creating a new spreadsheet with all values converted to USD. It also requires correct in-sheet and cross-sheet formula references while preserving the original spreadsheet layout.

E.10 Example 10

Percent	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987
GNI growth	8%	12%	23%	24%	14%	5%	13%	8%	19%	19%	7%	-2%	3%	5%	4%	1%	2%
IBRD and IDA lending and IDA grants/GNI	0.35%	0.40%	0.42%	0.44%	0.45%	0.45%	0.48%	0.62%	0.46%	0.57%	0.58%	0.59%	0.51%	0.59%	0.68%	0.73%	
IBRD and IDA lending	0.35%	0.40%	0.42%	0.44%	0.45%	0.45%	0.48%	0.62%	0.46%	0.57%	0.58%	0.59%	0.51%	0.59%	0.68%	0.73%	



but also implicit ex ante debt relief and financial support. Most IDA credits carry a zero or very low interest rate, and repayments typically extend over 30–50 years; however, more than one-third of IDA-eligible countries receive all or part of their IDA resources in the form of grants that carry no repayments in the future. Whereas IDA focuses on the most impoverished nations, the World Bank's other lending arm, IBRD, has played a crucial role in coordinating responses to regional and global challenges by providing loans and financial services to middle-income and creditworthy low-income countries (figure 1.4). IBRD was created to support countries rebuilding after World War II and has continued its crisis and emergency support through increased lending to countries affected by other crises since then, including the 2008–09 financial crisis, the 2014 Ebola outbreak, and the COVID-19 pandemic. Since inception, IBRD and IDA lending has responded positively to adverse external shocks affecting the economies of countries eligible for such financing, and this countercyclical lending has been a recurring and stabilizing response to dramatic drops in economic growth in these economies over the years.

Figure 18: Generating reports from tabular data requires financial knowledge of data analysis, financial events, and visualization. For example, one may plot two series with different units on a single chart (e.g., using a secondary y-axis) to reveal their correlation.

E.11 Example 11

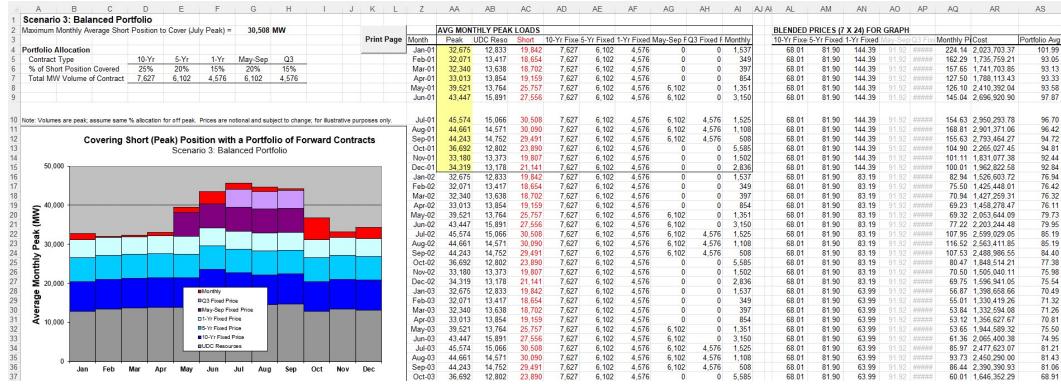


Figure 19: This Excel sheet shows an assumption-update workflow, where a mix of forward contracts is used to cover monthly peak-load short positions. It lists the contract allocations and MW volumes, along with monthly peak loads and the resulting short MW. A table on the right computes blended prices and portfolio costs, and the stacked chart visualizes coverage by contract type over the year.