FinMRAGBench: A Realistic and Complex Benchmark for Multi-Modal RAG in Financial Document Analysis

Shouqing Yang[1,2]*, Qi Zhang[1]*, Yuhang Yang[1,2], Ruikang Xu[1], Yuwei Hou[1], Zhulin Jia[4], Lirong Gao[1], Haobo Wang[1,2]†, Jinglei Chen[3], Jiexiang Wang[3], Sheng Guo[3], Bo Zheng[3], Gang Chen[1]

[1]State Key Laboratory of Blockchain and Data Security, Zhejiang University,
[2]Innovation and Management Center,
School of Software Technology(Ningbo), Zhejiang University,
[3]MYbank, Ant Group, [4]Guanghua School of Management, Peking University

Abstract

Retrieval-augmented generation (RAG) has become a widely adopted paradigm for realistic financial analysis over financial documents. However, existing benchmarks fail to capture realistic financial analysis settings that involve cross-document retrieval, multi-page evidence integration, and diverse analytical tasks. To address this gap, we introduce **FinMRAGBench**, a comprehensive multi-modal financial RAG benchmark in which most questions require retrieving evidence scattered across multiple pages and documents, constructed from large-scale real-world annual reports and comprising 887 expert-verified QA pairs spanning five representative financial analysis tasks. Moreover, we introduce **FinMRAGAgent**, an agent trained on high-quality agentic trajectories following the reasoning-and-acting (ReAct) paradigm, capable of dynamic tool invocation and multi-step financial analysis. Our extensive experiments show that current multi-modal RAG systems still struggle with incomplete retrieval and complex financial reasoning. In contrast, FinMRAGAgent achieves the strongest overall performance across all models, demonstrating that our structured reasoning approach significantly enhances multi-modal RAG in realistic financial scenarios. The code and data are available at https://github.com/sqyangit/FinMRAGBench.

1 Introduction

Retrieval-Augmented Generation (Yang et al., 2024; Zhang et al., 2025) has emerged as an effective paradigm for mitigating hallucinations (Huang et al., 2025) and improving overall large language model performance by integrating external knowledge with internal parametric knowledge (Zhang et al., 2024; Gao et al., 2023). Its success has led to broad adoption across various vertical domains.

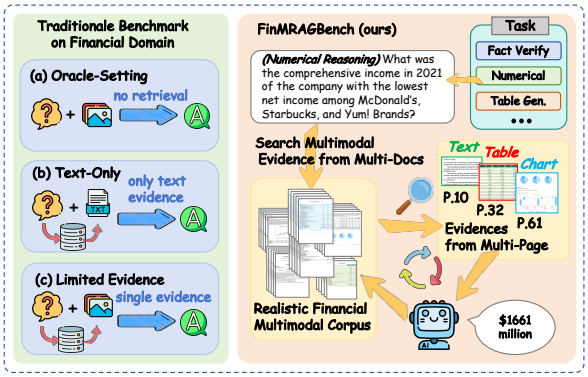


Figure 1: Comparison of Traditional Financial Benchmarks and FinMRAGBench. Traditional benchmarks rely on oracle context, text-only evidence, or single-page retrieval, limiting realistic evaluation. In contrast, FinMRAGBench supports end-to-end multi-modal RAG with cross-document, multi-page financial evidence across diverse financial analysis tasks.

The financial domain, in particular, places pressing demands on multi-modal RAG: financial documents are typically long, information-dense, and contain complex multi-modal content such as text, tables, and charts (Lai et al., 2025; Krumdick et al., 2024). In this context, accurate retrieval becomes critical, as even minor errors in financial question answering can lead to significant consequences in investment decisions, regulatory compliance, and risk management (Hopkin, 2018; Haeri et al., 2025). Therefore, building a reliable multi-modal RAG system tailored to the characteristics of financial documents is of critical importance.

To evaluate financial RAG systems, existing benchmarks have made notable efforts toward grounding question answering in financial documents. However, they still fall short of capturing the complexity of real-world financial analysis in several key aspects: some focus exclusively on text-based retrieval, neglecting critical visual information (e.g., FinAgentBench (Choi et al., 2025b) and OmniEval (Wang et al., 2025c)), while others

*Equal contribution
†Corresponding Author: wanghaobo@zju.edu.cn

Findings of the Association for Computational Linguistics: ACL 2026, pages 3822–3867
July 2-7, 2026

incorporate multi-modal inputs only under oracle-context settings that circumvent end-to-end retrieval evaluation (e.g., FinChart-Bench (Shu et al., 2025) and FinMME (Luo et al., 2025)). Although FinRAGBench-V (Zhao et al., 2025) integrates both multi-modality and retrieval, the majority of its questions rely on evidence from a single page or trivially adjacent pages within the same document. In contrast, real-world financial analysis requires practitioners to routinely retrieve and synthesize heterogeneous evidence across multiple filings, non-adjacent pages, and modalities to support a diverse range of analytical tasks. Moreover, existing benchmarks typically cover only a narrow set of task types such as numerical calculation and binary fact verification, failing to capture the full spectrum of analytical activities performed by financial professionals. Collectively, these limitations underscore the urgent need for a more realistic, complex, and comprehensive multi-modal RAG benchmark tailored to financial document analysis.

In this work, we propose **FinMRAGBench**, a realistic and complex benchmark that encompasses five task families to evaluate the core analytical capabilities essential for real-world financial report analysis and exhibits four key characteristics: **(1) Realistic Financial Evidence**: All evidence is drawn from authentic financial reporting documents, reflecting the information sources actually used by professional analysts. The evidence covers multi-modal content in financial reports, including text, tables, and charts. **(2) Diverse Task Types**: The benchmark spans five task families grounded in practices, enabling comprehensive evaluation of multi-modal RAG capabilities in authentic scenarios. **(3) Complex Retrieval Process**: Nearly **88.73%** of questions require evidence from multiple non-adjacent pages, and **75.20%** further necessitate cross-document synthesis. **(4) Multi-step Generation Demand**: Solving tasks requires multiple rounds of agent-environment interaction, averaging **7.49** interaction steps per question. FinMRAGBench is built upon a large-scale retrieval corpus sourced from 10-K, 10-Q, and 8-K filings of publicly listed companies on the SEC EDGAR platform, comprising 723 documents and 96,549 pages, ultimately resulting in 887 expert-verified QA pairs, each linked to its evidence pages.

We conducted a comprehensive evaluation of FinMRAGBench by benchmarking three multi-modal retrievers and 15 multi-modal generators, from which we draw the following key findings: (1) multi-modal retrieval across documents remains highly challenging, with an average Recall@10 of only **42.46%** across all tasks. (2) Current models consistently struggle with generation across all task categories in this complex and realistic financial setting, with the best-performing model achieving an average score of only **47.13%** across all tasks. To address these challenges, we further introduce **FinMRAGAgent**, an agent trained on high-quality agentic trajectories following the *reasoning-and-acting* paradigm, enabling it to dynamically invoke search and computation tools during inference for multi-step financial analysis. As a result, (3) FinMRAGAgent achieves the strongest overall performance, with an average score of **59.75%**, demonstrating that the reasoning-and-acting paradigm can significantly enhance model capabilities in realistic financial RAG scenarios.

2 Related Work

Retrieval-Augmented Generation and Multi-Modal LLM. Retrieval-Augmented Generation (RAG) extends large language models with external knowledge and has been widely adopted in domain-specific applications (Lewis et al., 2020; Yang et al., 2024). In the financial domain, RAG is particularly important for analyzing long and multi-modal documents such as annual reports, which interleave text, tables, and charts across multiple sections and filings (Loukas et al., 2025). Recent advances in multi-modal large language models enable joint reasoning over textual and visual inputs, making multi-modal RAG increasingly feasible (Abootorabi et al., 2025; Mei et al., 2025). However, existing RAG methods (Wang et al., 2025a) are still primarily designed for short contexts or limited evidence scopes, struggling to robustly handle multi-page, cross-document, multi-modal evidence integration in realistic financial analysis settings.

Financial RAG Benchmark. Following the growing adoption of RAG in financial analysis, several benchmarks have been proposed to evaluate retrieval-augmented question answering over financial documents. Prior work such as FinanceBench (Islam et al., 2023), FinDER (Choi et al., 2025a), FinDVer (Zhao et al., 2024), FinAgentBench (Choi et al., 2025b), OmniEval (Wang et al., 2025c), and XBRL-Agent (Han et al., 2024) primarily focuses on text-centric financial QA settings, overlooking the rich visual content in financial reports. Some benchmarks incorporate

Benchmark	Retrieval Corpus		Task Type					Evidence-Sources	
	Domain	Multi-Modal	FV	NR	CG	TG	KR	Cross-Docs(%)	Cross-Pages(%)
M3DocVQA	General	✓	✗	✓	✗	✗	✗	–	–
FinDER	Finance	✗	✗	✓	✗	✗	✗	0	0
FinDVer	Finance	✗	✓	✗	✗	✗	✗	0	0
OmniEval	Finance	✗	✓	✓	✗	✗	✓	0	0
FinAgentBench	Finance	✗	✗	✓	✗	✗	✗	0	0
FinRAGBench-V	Finance	✓	✗	✓	✗	✗	✓	0	12.64
FinMRAGBench	Finance	✓	✓	✓	✓	✓	✓	**75.20**	**88.73**

Table 1: Comparison of existing dataset with FinMRAGBench. In contrast to prior benchmarks, FinMRAGBench supports a diverse set of financial analysis tasks and exhibits substantially higher proportions of cross-document and cross-page evidence, reflecting more realistic financial reasoning scenarios.

multi-modal inputs (Shu et al., 2025; Luo et al., 2025), but often adopt oracle-context settings that bypass end-to-end retrieval challenges. Although FinRAGBench-V (Zhao et al., 2025) introduces multi-modal retrieval, its questions are largely grounded in single or adjacent pages within a document. In contrast, FinMRAGBench targets realistic financial analysis by requiring cross-document, multi-page multi-modal evidence and diverse analytical tasks, as summarized in Table 1.

3 FinMRAGBench

3.1 Realistic Financial Corpus Construction

Real-world financial analysis relies heavily on corporate annual reports, which contain dense, multi-modal evidence spanning narrative text, structured tables, and visual charts. To reflect this reality, we construct our retrieval corpus primarily from authentic annual filings. To support retrieval over this corpus, each page is annotated with structured metadata (*industry*, *company*, and *filing year*), enabling hierarchical retrieval and fine-grained filtering as described in Section 4.3. This realistically structured and information-dense corpus provides a strong foundation for evaluating RAG systems in authentic financial scenarios.

3.2 Diverse Task Definition

Existing financial RAG benchmarks oversimplify real-world financial analysis by focusing narrowly on numerical calculation or binary fact verification, thereby overlooking the diverse analytical capabilities required in professional practice. To bridge this gap, we define five task families grounded in authentic analyst workflows: *Explainable Fact Verification*, *Numerical Reasoning*, *Table Generation*, *Chart Generation*, and *Knowledge-Intensive Reasoning*. Crucially, each task requires models to retrieve and integrate multi-modal evidence dispersed across multiple documents and non-adjacent pages, spanning *narrative text, tables, and charts*, reflecting the complexity of real-world financial analysis. They also feature diverse output formats, from natural language explanations to executable code. This design reflects the integrative, multi-step nature of real-world financial analysis and enables a holistic evaluation of RAG capabilities. Full details are provided in Appendix C.1.

3.3 Complex QA Creation

To better capture the real-world complexity of financial analysis, we adopt a *evidence guided QA generation pipeline*. As shown in Figure 3, financial experts first select specific evidence fragments from designated pages across multiple annual reports and then construct questions that require integrating these pre-identified pieces, ensuring that each question requires multi-modal evidence retrieved from multiple pages and documents.

Analysis-driven Evidence Selection. To construct complex and realistic financial questions, we recruited financial experts to curate evidence pages by explicitly following established financial analysis practices. Guided by these practices (e.g., *time-series analysis of a single firm* and *cross-sectional comparison among industry peers*), the selected evidence is distributed across multiple documents and non-adjacent pages, spanning heterogeneous modalities. As illustrated in Figure 2(b), this analysis-driven evidence selection yields questions that require models to retrieve and align multi-modal content, including **text**, **tables**, and **charts**,

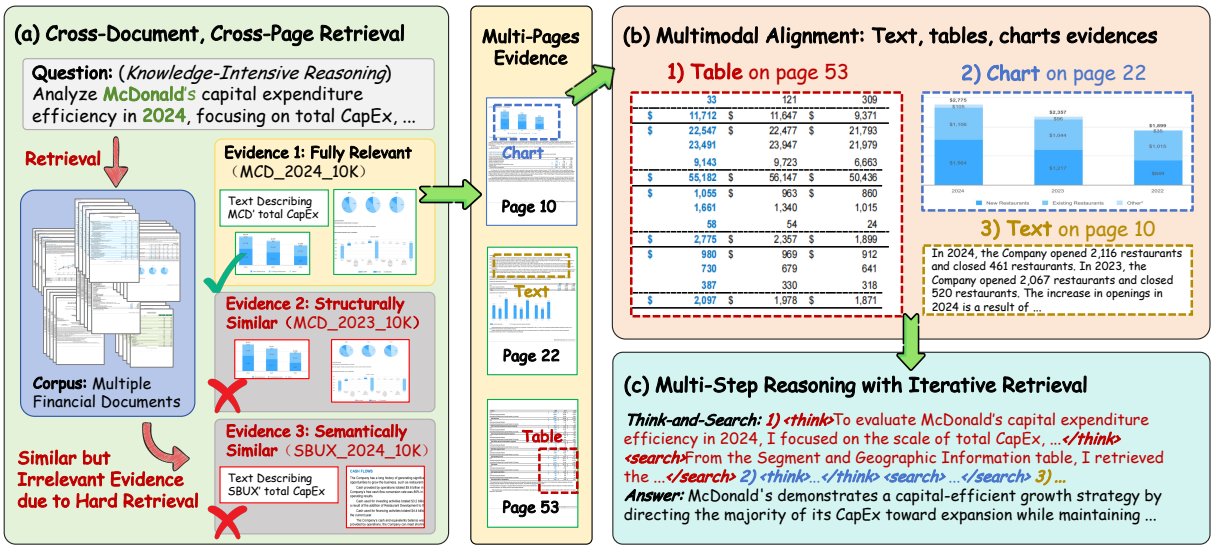


Figure 2: A representative example from FinMRAGBench. (a) Cross-document retrieval is challenged by highly similar layouts and contents across financial reports, leading to visually or textually similar but irrelevant evidence. (b) Relevant multi-modal evidence (text, tables, and charts) is distributed across multiple non-adjacent pages. (c) Answering the question requires iterative retrieval and multi-step reasoning to integrate heterogeneous evidence.

scattered across disparate pages and filings, reflecting the integrative nature of real-world financial analysis. Appendix C.2 details the financial analysis practices used to guide evidence page selection.

Evidence-guided QA Generation. Building on curated high-quality pages, we use GPT-4o (OpenAI, 2023) with task-specific prompts to generate diverse QA pairs. For each page set, the model produces candidate questions along with supporting evidence drawn from the input pages. All questions are manually reviewed against five criteria: *relevance*, *completeness*, *feasibility*, *clarity*, and *context independence*, and those failing any criterion are revised or discarded. We further enforce **round-trip consistency verification** to ensure that each question genuinely depends on its evidence and that the evidence is sufficient for answering it. *Redundant pages* are removed and *missing evidence* is manually supplemented, yielding a final set of question-evidence pairs with bidirectional consistency. As illustrated in Figure 2(c), answering these questions often requires **multi-step reasoning with iterative retrieval**, where models must resolve intermediate sub-questions by dynamically retrieving additional evidence across multiple turns before producing a final answer. To ensure high fidelity, we generate initial answers using GPT-4o with task-specific reasoning prompts and subject them to rigorous manual verification by at least two financial experts, resulting in a final set of high-fidelity, expert-validated answers.

To further ensure annotation reliability, we adopt a formal multi-stage verification protocol covering question review, evidence verification, and answer validation. We also report inter-annotator agreement statistics in Appendix B.3.

3.4 Difficulty and Complexity Analysis

To faithfully reflect the complexity of real-world financial analysis, we explicitly engineer difficulty into both retrieval and generation stages during annotation. **(1) Document Similarity**: Financial filings exhibit strong structural and semantic similarity due to standardized reporting templates and overlapping industry narratives, making it difficult to distinguish relevant evidence from plausible but irrelevant content. **(2) Cross-page and cross-document Retrieval**: Nearly **88.73%** of questions require evidence from multiple non-adjacent pages, and **75.20%** further demand retrieve across multiple documents. **(3) Multi-modal Alignment**: Relevant information is distributed across heterogeneous modalities (narrative text, structured tables, and visual charts), requiring models to align and reason over semantically equivalent but format-divergent content. **(4) Multi-step Reasoning**: Solving tasks requires multi-step agent-environment interaction, averaging **7.49** interaction steps per question as measured from expert trajectories. Moreover, the output formats are also highly diverse, ranging from natural language explanations and

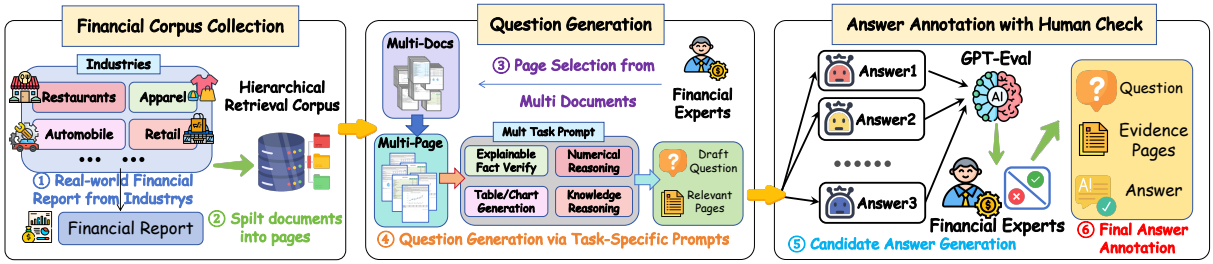


Figure 3: Overview of our dataset construction pipeline.

numerical values to executable code (e.g., Python for chart generation) and structured formats (e.g., markdown tables), reflecting the varied analytical demands of financial professionals.

3.5 Dataset Statistic

Financial Multi-Modal Corpus. The financial corpus is constructed from 723 Form 10-K filings collected from the SEC EDGAR[1] database, covering 127 publicly listed companies across 10 Global Industry Classification Standard (GICS) sectors over the last five years. Each filing is converted into PDF format and segmented into individual pages using wkhtmltopdf[2], resulting in 96,549 pages that serve as the atomic units for retrieval. This corpus design reflects realistic financial analysis settings, where analysts must retrieve and integrate evidence from large, multi-year, cross-company filings.

Question Categories. Drawing from real-world financial analysis workflows, FinMRAGBench comprises 887 questions organized into five primary task categories and 16 fine-grained subcategories. These categories capture the diverse analytical demands encountered when interpreting corporate annual reports, ranging from factual validation and numerical computation to structured content generation and in-depth reasoning. The distribution of question categories and subtypes is shown in Figure 4. Compared to existing financial RAG benchmarks, FinMRAGBench covers a broader and more realistic range of question types, particularly in terms of numerical reasoning, multimodal generation, and knowledge-intensive analysis. A detailed comparison of task coverage across benchmarks is provided in Table 1.

4 FinMRAGAgent

Beyond the benchmark, we propose FinMRAGAgent, an agentic, tool-integrated RAG framework

[1] https://www.sec.gov/search-filings
[2] https://wkhtmltopdf.org/

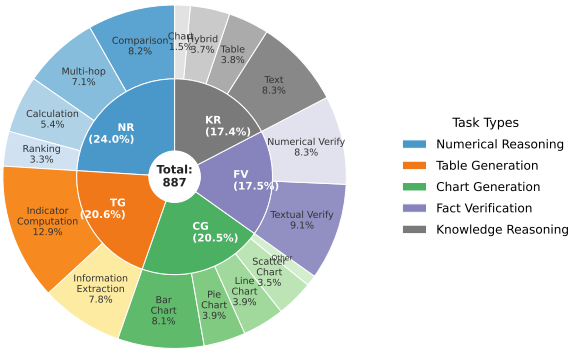


Figure 4: Statistics of Task Types in the Dataset.

for financial document analysis. As shown in Figure 5, the framework is trained on high-quality agentic reasoning trajectories to enable step-by-step reasoning with explicit tool use.

4.1 Agentic Reasoning Trajectories

In real-world financial analysis, evidence is scattered across multiple pages and documents in heterogeneous modalities, requiring experts to iteratively retrieve information, perform analytical operations, and refine their conclusions. Inspired by this workflow, we construct tool-integrated reasoning trajectories to capture the interaction between internal reasoning and external tools. Each trajectory consists of interleaved reasoning states, tool-use actions, and environment observations, following a structured protocol with `<think>`, `<search>`, `<python>`, and `<answer>` steps. Formally, a trajectory is represented as a sequence $\tau = \{(t_j, a_j, o_j)\}_{j=1}^{L}$, where t_j denotes the agent's internal reasoning at step j, $a_j \in \{\texttt{search}, \texttt{python}, \texttt{answer}\}$ denotes the executed action, and o_j is the corresponding observation returned by the external environment.

To ensure high-quality supervision, we retain a trajectory τ only if it satisfies three criteria: *answer correctness*, *reasoning consistency*, and *reasoning complexity*. Additional details of trajectory synthe-

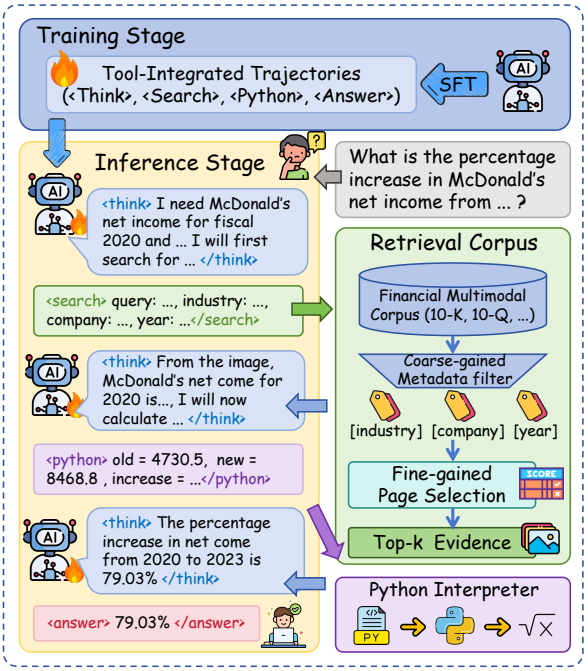


Figure 5: The overall framework of **FinMRAGAgent**.

sis and filtering are provided in Appendix C.3.

4.2 Learning and Execution of the Agent

FinMRAGAgent is trained via supervised fine-tuning on the filtered tool-integrated reasoning trajectories, allowing it to learn when and how to coordinate internal reasoning with external tool usage. Given a set of K trajectories, the i-th trajectory is denoted as $\tau^{(i)} = \{(t_j^{(i)}, a_j^{(i)}, o_j^{(i)})\}_{j=1}^{L_i}$. At each step j, the model is trained to predict the target output conditioned on the multi-modal input image $I^{(i)}$, the query $q^{(i)}$, and the interaction history $(t_{<j}^{(i)}, a_{<j}^{(i)}, o_{<j}^{(i)})$. The supervised fine-tuning objective is defined as:

$$\max_{\theta} \sum_{i=1}^{K} \sum_{j=1}^{L_i} \log p_\theta\big(y_j^{(i)} \mid I^{(i)}, q^{(i)}, t_{<j}^{(i)}, a_{<j}^{(i)}, o_{<j}^{(i)}\big) \tag{1}$$

where θ denotes model parameters and $y_j^{(i)}$ denotes the supervised target token sequence at step j.

At inference time, the agent performs a multi-step reasoning process by alternately generating intermediate reasoning steps and invoking external tools. Conditioned on the interaction history, it adaptively selects actions such as issuing `<search>` requests for evidence retrieval or invoking `<python>` for numerical computation, until a final `<answer>` is produced. Detailed inference procedures are provided in Appendix D.4.

4.3 Coarse-to-Fine Retrieval

As discussed previously, financial documents exhibit strong structural and semantic similarity, which introduces significant retrieval ambiguity. To address this, FinMRAGAgent adopts a coarse-to-fine multi-modal retrieval strategy when executing actions to retrieve relevant evidence. At the coarse stage, we restrict the search space using document-level metadata. Each page p is associated with metadata $m(p) = (\text{industry}, \text{company}, \text{year})$, and candidate pages are filtered as:

$$\mathcal{P}_{\text{coarse}}(q) = \{\, p \mid m(p) \in \mathcal{M}(q) \,\} \tag{2}$$

where $\mathcal{M}(q)$ denotes metadata constraints inferred from the query. Within this reduced space, a multi-modal retriever encodes the query and pages into vector representations $\mathbf{V}_q = E(q)$ and $\mathbf{V}_p = E(p)$ to retrieve top-k candidates. These candidates are further refined by a query-aware relevance scorer:

$$S_{\text{fine}}(p, q) = f_\phi(p, q), \tag{3}$$

which selects the most informative pages for downstream reasoning. This hierarchical design facilitates robust and precise retrieval over large-scale financial corpora. Implementation details are provided in Appendix D.3.

5 Experiment

5.1 Experimental Setup

Baselines. We evaluate a diverse set of multi-modal RAG systems spanning retrieval, generation, and advanced RAG methods. Specifically, we benchmark three multi-modal retrievers and select the strongest one to retrieve the top-k pages ($k = 10$) for downstream generation. For generation, we evaluate 15 multi-modal large language models, including both closed-source and open-source models. We further include advanced RAG baselines beyond vanilla generation, including ReAct-style methods such as IRCOT (Trivedi et al., 2023), multi-agent visual-document RAG methods such as ViDoRAG (Wang et al., 2025a), and our proposed FinMRAGAgent.

Metrics. We evaluate multi-modal RAG systems on FinMRAGBench from two complementary perspectives aligned with the RAG pipeline: retrieval quality and answer accuracy. Retrieval quality is measured using Recall to assess evidence coverage of retrieved pages. Answer accuracy is evaluated

Model	Fact Verification		Numerical Reasoning		Table Generation		Chart Generation		Knowledge Reasoning	
	LJS	HQR	F1	HQR	RMS P	RMS R	ECR	PASS	LJS	HQR
Open-source Models with Vanilla RAG										
Qwen2.5-VL-7B-Instruct	28.97	10.32	22.53	15.02	23.85	24.20	32.97	4.40	38.90	16.23
Qwen2.5-VL-32B-Instruct	42.71	28.39	30.12	22.54	33.49	33.40	92.31	24.18	**49.03**	27.27
Qwen2.5-VL-72B-Instruct	54.32	44.52	**37.45**	22.07	**40.49**	**39.97**	96.70	37.91	42.66	28.57
Qwen3-VL-4B-Instruct	42.77	25.81	23.89	10.80	24.84	24.14	88.46	18.13	37.47	14.94
Qwen3-VL-8B-Instruct	50.32	40.51	24.83	20.66	29.45	29.41	**97.25**	29.67	38.08	21.15
Qwen3-VL-32B-Instruct	**60.89**	**53.16**	33.37	**27.70**	37.25	37.12	93.41	**40.11**	48.97	**35.26**
MiniCPM-V-4.5	46.58	30.97	21.44	10.33	14.58	15.05	76.67	8.24	37.21	12.34
InternVL3.5-8B	24.32	14.84	26.60	14.08	19.30	19.26	88.46	10.44	41.88	13.64
InternVL3.5-38B	43.61	33.55	31.81	20.19	26.24	26.15	88.46	16.48	43.51	24.03
Close-source Models with Vanilla RAG										
GPT-4o	57.28	51.27	38.01	26.29	46.58	46.43	96.70	39.56	51.28	35.26
GPT-5.1	**64.18**	**56.33**	**41.90**	**38.97**	48.37	**48.44**	96.70	**47.25**	**54.10**	**47.44**
Gemini-2.5-Flash	57.74	49.03	28.52	19.25	39.70	39.32	96.15	23.63	42.66	25.32
Gemini-2.5-Pro	63.16	56.13	32.63	25.35	**50.06**	48.15	96.15	36.81	49.74	39.61
Claude-3.5-Sonnet	56.58	50.97	31.92	25.35	42.38	42.13	98.35	37.36	49.48	33.12
Claude-4.5-Sonnet	56.32	44.52	37.98	28.17	40.40	40.92	**99.45**	36.81	52.01	35.71
Qwen3-VL-8B-Instruct with Advanced Methods										
Vanilla*	57.87	47.10	47.07	50.23	38.81	38.64	**98.90**	45.60	40.65	24.68
IRCOT*	62.58	**56.13**	52.45	60.56	**55.13**	**54.48**	93.41	46.15	47.86	25.97
ViDoRAG†*	48.84	38.37	41.28	36.32	29.95	29.44	98.24	31.76	56.71	42.86
FinMRAGAgent (**Ours**)*	**63.94**	55.48	**62.39**	**65.73**	54.47	52.01	97.25	**64.29**	**63.44**	**56.49**

Table 2: Main results on FinMRAGBench across five financial analysis tasks: FV (Explainable Fact Verification), NR (Numerical Reasoning), TG (Table Generation), CG (Chart Generation), and KR (Knowledge-Intensive Reasoning). Methods marked with * are enhanced with our coarse-to-fine retrieval (C2F) strategy. ViDoRAG† is evaluated using GPT-5.1, as its multi-agent prompting relies on stronger backbone models for handling complex financial tasks.

with task-specific metrics. Specifically, for Explainable Fact Verification and Knowledge-Intensive Reasoning, we use an LLM-based judge score **(LJS)** together with the High-Quality Rate **(HQR)**. For Numerical Reasoning, we report **F1** as the primary automatic metric and additionally use LJS and HQR to complement exact matching. For Table Generation, we adopt Relative Mapping Similarity **(RMS)**, and for Chart Generation, we evaluate code executability and data correctness using **ECR** and **PASS@1**. Full metric definitions and evaluation protocols are provided in Appendix E.

For the open-ended tasks evaluated with LJS, we additionally conduct a Human-LLM agreement analysis on a randomly sampled subset to validate judge reliability, with detailed results provided in Appendix E.4.

5.2 Main Result

① **Overall generation performance remains limited across all task categories.** Across the five task categories in FinMRAGBench, generation performance remains limited (Table 2), indicating that current multi-modal LLMs struggle with realistic financial RAG. For numerically intensive tasks (Numerical Reasoning, Table Generation, and Chart Generation), the best score reaches only **50.06%**. For explanation-oriented tasks (Explainable Fact Verification and Knowledge-Intensive Reasoning), the highest score is **64.18%**. Although code executability in Chart Generation is relatively high, numerical accuracy remains a major bottleneck. As further illustrated in Figure 6, different task types stress distinct model capabilities, leading to pronounced task-dependent performance variations across models. Overall, this performance gap can be partially attributed to models' limited ability to integrate and reason over fragmented multi-modal evidence, as well as the intrinsic difficulty of precise numerical computation and visualization-oriented generation in financial analysis.

② **Realistic financial RAG is bottlenecked by multi-modal retrieval.** As shown in Table 3, current multi-modal retrievers struggle to retrieve relevant evidence from complex financial documents. Even the strongest retriever, ColQwen2 (Faysse et al., 2025), achieves only **42.46%** average recall, indicating substantial room for improvement. We observe two dominant failure modes from the retrieval results: **(i)** retrieving *visually similar but semantically irrelevant* pages due to standardized financial report templates, and **(ii)** *missing com-*

Model	FV	NR	TG	CG	KR
ColPali	33.43	20.58	23.63	40.20	28.05
VisRAG-Ret	43.74	31.22	34.34	49.08	32.41
ColQwen2	45.59	36.42	42.87	53.80	33.61
C2F (Ours)	63.67	75.16	74.41	84.94	56.75

Table 3: Recall@10 of different multi-modal retrievers. C2F denotes results of our coarse-to-fine retrieval applied to ColQwen2.

plete evidence when relevant pages are scattered across multiple documents and non-adjacent pages. These results highlight the inherent difficulty of realistic financial retrieval, where evidence is fragmented across documents, pages, and modalities.

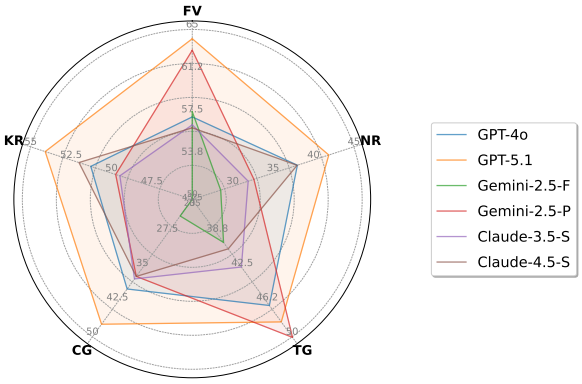


Figure 6: Radar plot comparing representative multi-modal LLMs across five financial analysis task categories in FinMRAGBench.

③ FinMRAGAgent yields substantial performance improvements in complex financial settings. Under fair comparison with enhanced retrieval, FinMRAGAgent consistently outperforms vanilla RAG, ReAct-style baselines, and visual-document RAG methods across all tasks. For example, on Chart Generation, FinMRAGAgent improves performance from **29.67%** (Vanilla), **46.15%** (IRCOT), and **31.76%** (ViDoRAG) to **64.29%**. These gains stem from explicitly integrating tools into an agentic, step-by-step reasoning process, enabling iterative evidence search and verifiable numerical computation rather than single-pass generation. Notably, while document RAG methods such as ViDoRAG perform competitively on single-page question answering, they struggle to adapt to complex cross-document, multi-page reasoning tasks, and can even underperform simpler ReAct-style approaches like IRCOT. Overall, these results highlight the importance of explicit task decomposition and tool-guided reasoning for reliable financial analysis under realistic multi-page and multi-hop settings.

5.3 Ablation Study

Impact of Coarse-to-Fine Retrieval. We conduct ablation experiments to evaluate the effectiveness of the proposed coarse-to-fine retrieval strategy. As shown in Table 4, both hierarchical retrieval and fine-grained page filtering consistently improve retrieval performance in a vanilla RAG pipeline. For Chart Generation, coarse-grained retrieval increases recall from 53.80% to 65.11%, and fine-grained filtering further boosts recall to **84.94%**. Using the improved retrieval for generation with Qwen3-VL-8B-Instruct yields corresponding gains in end-to-end performance (from **29.67%** to **45.60%**), confirming retrieval quality as a key driver of overall RAG effectiveness.

Method	FV	NR	TG	CG	KR
Vanilla	45.59	36.42	42.87	53.80	33.61
HR	55.55	56.22	64.83	65.11	46.62
HR+PS	63.67	75.16	74.41	84.94	56.75

Table 4: Ablation results of coarse-to-fine retrieval, showing the impact of hierarchical retrieval (HR) and page selection (PS) on Recall@10 across different tasks.

Impact of Supervised Fine-Tuning on Tool Use. To analyze the impact of tool-integrated supervised fine-tuning, we compare FinMRAGAgent with a ReAct-style tool-use baseline built on an untuned Qwen3-VL-8B-Instruct model (Bai et al., 2025a). As shown in Table 5, ReAct-style tool use yields moderate improvements over the vanilla setting, but consistently underperforms FinMRAGAgent across financial analysis tasks. Qualitative analysis shows that the ReAct baseline often suffers from weaker intent recognition, suboptimal tool selection, and less reliable instruction following. In contrast, supervised fine-tuning enables FinMRAGAgent to learn *when* to invoke tools and *which* tools to use, resulting in more accurate and efficient tool utilization. These results indicate that the performance gains of FinMRAGAgent arise not from tool use alone, but from supervised fine-tuning that enables more accurate and timely tool invocation.

Note: We place more experimental results in the Appendix A, including upper-bound and sanity-check experiments, model scaling analysis, and inference-time efficiency comparisons.

Model	FV	NR	TG	CG	KR
w/o SFT	63.10	57.55	49.15	46.70	46.63
w/ SFT	63.94	65.73	54.47	64.29	63.44

Table 5: Comparison between tool use with and without supervised fine-tuning on Qwen3-VL-8B-Instruct across financial analysis tasks. Reported scores correspond to task-specific primary metrics: LJS for FV and KR, HQR for NR, RMS-P for TG, and PASS for CG.

6 Conclusions

In this paper, we introduce FinMRAGBench, a multi-modal financial RAG benchmark that evaluates large language models in realistic financial analysis tasks, especially under multi-page and multi-document scenarios. Extensive experiments show that current systems struggle with cross-page and cross-document retrieval, multi-modal understanding, and complex reasoning. To address these challenges, we further propose FinMRAGAgent, a tool-integrated RAG framework trained on agentic reasoning trajectories, achieving the strongest overall performance.

Limitations

Despite its strengths, FinMRAGBench has several limitations. Due to the inherent complexity of financial analysis tasks and the need for careful expert validation, the current version of FinMRAGBench is limited in scale compared to fully automated datasets. While this design prioritizes annotation quality and realism, it constrains the overall dataset size. We leave the expansion of FinMRAGBench to larger document collections and broader coverage of financial scenarios to future work.

Acknowledgement

This paper was supported by Ningbo Key Research and Development Program (No. 2025Z190) and MYbank, Ant Group.

Ethical Considerations

FinMRAGBench is constructed entirely from publicly available financial documents, primarily Form 10-K filings released through official regulatory channels. These documents are intended for public disclosure and do not contain personal, private, or sensitive information.

The dataset is annotated by human experts with financial backgrounds under a well-defined annotation protocol. Annotators are provided with detailed task instructions that specify annotation guidelines, quality requirements, and intended research use of the data. All annotations are performed with informed consent. Annotators are recruited through academic or professional channels and are compensated appropriately for their contributions.

To ensure data integrity and ethical compliance, we adopt a multi-stage human verification process. Each question-evidence pair and its corresponding answer are reviewed by multiple annotators to ensure correctness, neutrality, and faithfulness to the source documents. Throughout the annotation process, we take care to avoid introducing misleading interpretations, speculative content, or any sensitive information beyond what is explicitly stated in the original filings.

Given that the data sources are publicly available and the annotation process does not involve personal data or vulnerable populations, this study poses minimal ethical risk and does not require institutional ethics board approval. Overall, FinMRAGBench is designed to be a clean and ethically sound benchmark for research on financial retrieval-augmented generation systems. Additional details regarding data sources and annotation procedures are provided in Section 3 and Appendix B.

References

Mohammad Mahdi Abootorabi, Amirhosein Zobeiri, Mahdi Dehghani, Mohammadali Mohammadkhani, Bardia Mohammadi, Omid Ghahroodi, Mahdieh Soleymani Baghshah, and Ehsaneddin Asgari. 2025. Ask in any modality: A comprehensive survey on multimodal retrieval-augmented generation. *arXiv preprint arXiv:2502.08826*.

Anthropic. 2024. Claude 3.5 sonnet. https://www.anthropic.com/news/claude-3-5-sonnet. Accessed: 2025-12-23.

Anthropic. 2025. Introducing claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5. Accessed: 2025-12-23.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025b. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Chanyeol Choi, Jihoon Kwon, Jaeseon Ha, Hojun Choi, Chaewoon Kim, Yongjae Lee, Jy-yong Sohn, and Alejandro Lopez-Lira. 2025a. Finder: Financial dataset for question answering and evaluating retrieval-augmented generation. In *Proceedings of the 6th ACM International Conference on AI in Finance*, pages 638–646.

Chanyeol Choi, Jihoon Kwon, Alejandro Lopez-Lira, Chaewoon Kim, Minjae Kim, Juneha Hwang, Jaeseon Ha, Hojun Choi, Suyeol Yun, Yongjin Kim, and 1 others. 2025b. 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.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2025. Colpali: Efficient document retrieval with vision language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997.

Amin Haeri, Jonathan Vitrano, and Mahdi Ghelichi. 2025. Generative ai enhanced financial risk management information retrieval. *arXiv preprint arXiv:2504.06293*.

Shijie Han, Haoqiang Kang, Bo Jin, Xiao-Yang Liu, and Steve Y. Yang. 2024. XBRL agent: Leveraging large language models for financial report analysis. In *Proceedings of the 5th ACM International Conference on AI in Finance, ICAIF 2024, Brooklyn, NY, USA, November 14-17, 2024*, pages 856–864. ACM.

Paul Hopkin. 2018. *Fundamentals of risk management: understanding, evaluating and implementing effective risk management*. Kogan Page Publishers.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

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

Michael Krumdick, Rik Koncel-Kedziorski, Viet Dac Lai, Varshini Reddy, Charles Lovering, and Chris Tanner. 2024. Bizbench: A quantitative reasoning benchmark for business and finance. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8309–8332.

Harold W. Kuhn. 2010. The hungarian method for the assignment problem. In Michael Jünger, Thomas M. Liebling, Denis Naddef, George L. Nemhauser, William R. Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence A. Wolsey, editors, *50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art*, pages 29–47. Springer.

Viet Lai, Michael Krumdick, Charles Lovering, Varshini Reddy, Craig Schmidt, and Chris Tanner. 2025. Secqa: A systematic evaluation corpus for financial qa. In *Proceedings of The 10th Workshop on Financial Technology and Natural Language Processing*, pages 221–236.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. 2023a. Deplot: One-shot visual language reasoning by plot-to-table translation. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10381–10399. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2511–2522. Association for Computational Linguistics.

Lefteris Loukas, Fabian Billert, Manos Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2025. Edgar-crawler: From raw web documents to structured financial nlp datasets. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 761–764.

Junyu Luo, Zhizhuo Kou, Liming Yang, Xiao Luo, Jinsheng Huang, Zhiping Xiao, Jingshu Peng, Chengzhong Liu, Jiaming Ji, Xuanzhe Liu, and 1 others. 2025. Finmme: Benchmark dataset for financial multi-modal reasoning evaluation. *arXiv preprint arXiv:2505.24714*.

Lang Mei, Siyu Mo, Zhihan Yang, and Chong Chen. 2025. A survey of multimodal retrieval-augmented generation. *arXiv preprint arXiv:2504.08748*.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

OpenAI. 2025. Gpt-5.1: A smarter, more conversational chatgpt. https://openai.com/index/gpt-5-1/. Accessed: 2025-12-23.

Dong Shu, Haoyang Yuan, Yuchen Wang, Yanguang Liu, Huopu Zhang, Haiyan Zhao, and Mengnan Du. 2025. Finchart-bench: Benchmarking financial chart comprehension in vision-language models. *arXiv preprint arXiv:2507.14823*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10014–10037. Association for Computational Linguistics.

Qiuchen Wang, Ruixue Ding, Zehui Chen, Weiqi Wu, Shihang Wang, Pengjun Xie, and Feng Zhao. 2025a. Vidorag: Visual document retrieval-augmented generation via dynamic iterative reasoning agents. *arXiv preprint arXiv:2502.18017*.

Qiuchen Wang, Ruixue Ding, Yu Zeng, Zehui Chen, Lin Chen, Shihang Wang, Pengjun Xie, Fei Huang, and Feng Zhao. 2025b. VRAG-RL: empower vision-perception-based RAG for visually rich information understanding via iterative reasoning with reinforcement learning. *CoRR*, abs/2505.22019.

Shuting Wang, Jiejun Tan, Zhicheng Dou, and Ji-Rong Wen. 2025c. Omnieval: An omnidirectional and automatic rag evaluation benchmark in financial domain. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5737–5762.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025d. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Pengzuo Wu, Yuhang Yang, Guangcheng Zhu, Chao Ye, Hong Gu, Xu Lu, Ruixuan Xiao, Bowen Bao, Yijing He, Liangyu Zha, Wentao Ye, Junbo Zhao, and Haobo Wang. 2025a. Realhitbench: A comprehensive realistic hierarchical table benchmark for evaluating llm-based table analysis. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 7105–7137. Association for Computational Linguistics.

Zonghan Wu, Junlin Wang, Congyuan Zou, Chenhan Wang, and Yilei Shao. 2025b. Towards competent AI for fundamental analysis in finance: A benchmark dataset and evaluation. *CoRR*, abs/2506.07315.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze D Gui, Ziran W Jiang, Ziyu Jiang, and 1 others. 2024. Crag-comprehensive rag benchmark. *Advances in Neural Information Processing Systems*, 37:10470–10490.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2025a. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Yuanqian Zhao, and 1 others. 2025b. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. *arXiv preprint arXiv:2509.18154*.

Qi Zhang, Shouqing Yang, Lirong Gao, Hao Chen, Xiaomeng Hu, Jinglei Chen, Jiexiang Wang, Sheng Guo, Bo Zheng, Haobo Wang, and Junbo Zhao. 2025. LeTS: Learning to think-and-search via process-and-outcome reward hybridization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5109–5122, Suzhou, China. Association for Computational Linguistics.

Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. Gme: Improving universal multimodal retrieval by multimodal llms. *arXiv preprint arXiv:2412.16855*.

Suifeng Zhao, Zhuoran Jin, Sujian Li, and Jun Gao. 2025. Finragbench-v: A benchmark for multimodal rag with visual citation in the financial domain. *arXiv preprint arXiv:2505.17471*.

Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. 2022. Multihiertt: Numerical reasoning over multi hierarchical tabular and textual data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6588–6600. Association for Computational Linguistics.

Yilun Zhao, Yitao Long, Tintin Jiang, Chengye Wang, Weiyuan Chen, Hongjun Liu, Xiangru Tang, Yiming Zhang, Chen Zhao, and Arman Cohan. 2024. Finver: Explainable claim verification over long and hybrid-content financial documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14739–14752.

A More Experiments and Analysis

A.1 Upper-Bound and Sanity-Check Experiments

To better contextualize the main results, we conduct two supplementary experiments that serve as an upper bound and a sanity check for multi-modal financial RAG. Results for both settings are summarized in Table 6.

Oracle setting In the oracle setting, we provide the generator with gold evidence pages annotated by experts, bypassing the retrieval stage entirely. This setting reflects an upper bound on generation performance under perfect evidence access, allowing us to isolate the limitations of the generation model itself. As expected, oracle results substantially outperform standard RAG settings across all tasks, indicating that retrieval errors remain a dominant bottleneck in realistic financial analysis. Nevertheless, performance does not saturate even under oracle evidence, suggesting that complex financial reasoning and multi-step analysis continue to pose challenges for current multi-modal LLMs.

Interestingly, on several tasks, Qwen3-VL-8B-Instruct still underperforms our ReAct-style FinMRAGAgent even with gold evidence, suggesting that step-by-step reasoning and explicit evidence integration remain critical for effectively leveraging multi-page, cross-document information. Consistent with this observation, replacing Qwen3-VL-8B-Instruct with its Thinking variant further improves performance under oracle evidence, indicating that deeper reasoning remains necessary even when perfect evidence is available.

Direct generation setting We additionally evaluate a direct generation setting, where models answer questions without any retrieval and rely solely on their parametric knowledge. This serves as a sanity check to assess the necessity of external evidence and to rule out potential data leakage. Across all task categories, direct generation performs significantly worse than RAG-based approaches, confirming that FinMRAGBench requires grounding in retrieved financial documents and cannot be solved reliably from parametric knowledge alone. Notably, GPT-5.1 and Qwen3-VL-8B-Instruct exhibit non-trivial performance on a subset of tasks, likely due to more recent pretraining data that may encode partial financial knowledge, but such parametric knowledge alone remains insufficient for robust financial analysis.

A.2 Detailed Retrieval Performance Analysis

To provide a more fine-grained understanding of retrieval behavior, we report task-wise retrieval recall across all five financial analysis tasks. As shown in Table 7, we compare vanilla retrieval, coarse-to-fine (C2F) retrieval, ReAct-style retrieval (IRCOT), and our proposed FinMRAGAgent, all built on ColQwen2 as the base retriever. Across all tasks, vanilla retrieval exhibits limited recall, highlighting the difficulty of identifying complete evidence in realistic financial settings. Applying coarse-to-fine retrieval substantially improves recall across all task categories, with particularly large gains on tasks requiring cross-document and multi-page evidence, such as Numerical Reasoning and Knowledge-Intensive Reasoning. IRCOT further improves recall by enabling iterative retrieval conditioned on intermediate reasoning steps, demonstrating the benefits of reasoning-aware retrieval. FinMRAGAgent achieves the strongest overall retrieval performance across tasks. Compared to IRCOT, it consistently attains higher recall on evidence-intensive tasks, indicating that supervised agentic trajectories enable more effective iterative evidence expansion and integration in complex financial scenarios.

A.3 Model Scaling Analysis

We additionally evaluated the effect of model capacity by training a larger 32B backbone (Qwen3-VL-32B-Instruct) with LoRA SFT using the same agentic trajectories and evaluation protocol. Table 8 compares the task-wise performance of the 8B and 32B versions of FinMRAGAgent. We observe consistent improvements across most task categories: FV improves from 63.94 to 70.26, TG from 54.47 to 64.82, CG from 64.29 to 69.46, and KR from 63.44 to 68.05, while numerical reasoning changes only slightly (from 65.73 to 64.29). Notably, gains are most pronounced on generation and knowledge-intensive reasoning tasks (TG, CG, KR), while numerical reasoning remains comparable across scales. This pattern suggests that FinMRAGAgent primarily improves structured multi-step reasoning and evidence integration rather than compensating for weak model capacity. Therefore, the effectiveness of the agent framework is not limited to small models: the reasoning-and-tool-use design remains beneficial as model scale increases.

Model	Fact Verification		Numerical Reasoning		Table Generation		Chart Generation		Knowledge Reasoning	
	LJS	HQR	F1	HQR	RMS P	RMS R	ECR	PASS	LJS	HQR
Models with Gold Evidence										
GPT-5.1	83.80	82.91	64.40	83.57	66.79	67.21	97.80	67.03	86.28	91.67
Qwen3-VL-8B-Instruct	66.39	60.65	52.31	53.52	37.18	36.94	98.90	51.65	63.96	59.09
Qwen3-VL-8B-Thinking	79.74	78.71	51.64	61.03	45.06	44.76	81.32	49.45	69.55	71.43
Direct Generation										
GPT-5.1	34.90	22.58	24.18	16.43	10.80	11.15	98.90	1.10	29.48	20.78
GPT-4o	23.68	8.39	18.41	6.10	3.89	3.86	99.45	0.00	31.56	5.85
Qwen3-VL-8B-Instruct	28.58	14.19	22.52	4.23	1.57	1.55	95.60	0.00	21.43	4.55
Qwen2.5-VL-7B-Instruct	19.29	1.94	16.04	3.76	0.80	0.78	82.97	0.00	35.13	4.55

Table 6: Results under two controlled settings on FinMRAGBench. *Models with Gold Evidence* are provided with ground-truth evidence pages, while *Direct Generation* answers questions without retrieval.

Method	FV	NR	TG	CG	KR
Vanilla Retrieval	45.59	36.42	42.87	53.80	33.61
C2F Retrieval	63.67	75.16	74.41	84.94	56.75
IRCOT*	59.35	77.89	71.17	79.25	53.76
FinMRAGAgent*	60.97	81.57	69.05	88.97	57.16

Table 7: Task-wise Retrieval Recall@10 under Different Retrieval Strategies on ColQwen2. Methods marked with * are enhanced with the proposed coarse-to-fine (C2F) retrieval strategy.

Model	FV	NR	TG	CG	KR
8B (Full SFT)	63.94	65.73	54.47	64.29	63.44
32B (LoRA SFT)	70.26	64.29	64.82	69.46	68.05

Table 8: Model scaling analysis of FinMRAGAgent. We compare an 8B backbone trained with full SFT and a 32B backbone trained with LoRA SFT under the same training trajectories and evaluation protocol.

A.4 Inference-Time Efficiency Comparison

To further contextualize computational efficiency, we measure the average per-sample inference time on a single A100 GPU using Qwen3-VL-8B-Instruct as the backbone. Table 9 reports the average runtime for Vanilla RAG, IRCOT, and FinMRAGAgent, together with the task-wise runtime breakdown. As expected, agent-based reasoning introduces additional overhead compared with single-pass vanilla generation. However, FinMRAGAgent remains substantially more efficient than IRCOT-style iterative retrieval, with an average runtime of 12.92 seconds per sample compared with 22.54 seconds, while also achieving stronger overall performance across tasks. These results suggest that FinMRAGAgent provides a more favorable efficiency-performance trade-off. It significantly improves structured multi-step reasoning while avoiding the excessive latency of fully iterative reasoning pipelines.

Method	FV	NR	TG	CG	KR	Avg.
Vanilla	4.46	5.87	3.76	4.88	7.04	5.19
IRCOT	11.93	12.99	28.31	17.94	46.36	22.54
FinMRAGAgent	11.63	8.80	14.85	12.37	18.07	12.92

Table 9: Inference-time efficiency comparison in seconds per sample on a single A100 GPU. All methods use Qwen3-VL-8B-Instruct as the backbone.

B More Dataset Details

B.1 Corpus Sources

The retrieval corpus is constructed from publicly available annual reports of publicly listed companies, covering a broad range of industries and reporting practices. In total, the corpus consists of 96,549 page-level documents extracted from 723 annual reports, spanning 10 Global Industry Classification Standard (GICS) sectors.

The collected reports exhibit substantial diversity in document length, layout structure, and information density. Pages include a mixture of narrative disclosures, financial tables, and visual elements such as charts and figures, reflecting the heterogeneous nature of real-world financial filings. This structural variability poses non-trivial challenges for document retrieval and evidence aggregation.

The corpus provides broad sectoral coverage across industries with distinct business models and reporting characteristics. Figure 7 illustrates the distribution of page volumes across industries, highlighting differences in reporting scale and disclosure intensity. Table 10 summarizes industry-level statistics, including the number of companies, reports, and pages per sector. In addition, Table 11 lists a representative subset of automobile compa-

nies included in the corpus to illustrate the diversity of firm-level coverage.

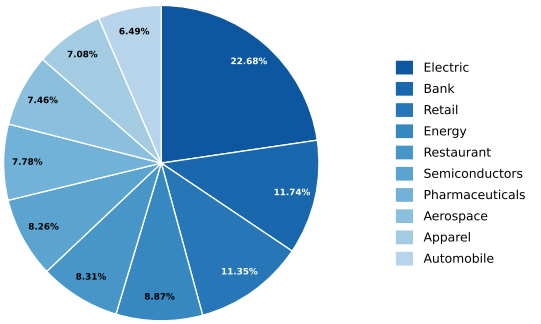


Figure 7: Industry-wise distribution of page counts in the retrieval corpus.

Industry	#Companies	#Files	#Pages	#Avg_Pages
Electric	15	88	21,894	248.8
Bank	10	60	11,337	188.9
Retail	20	115	10,961	95.3
Energy	10	60	8,560	142.7
Restaurant	15	81	8,024	99.1
Semiconductors	15	78	7,975	102.2
Pharmaceuticals	10	59	7,507	127.2
Aerospace	11	63	7,198	114.3
Apparel	11	65	6,831	105.1
Automobile	10	54	6,262	116
TOTAL	127	723	96,549	133.5

Table 10: Statistics of companies, reports, and pages across industries.

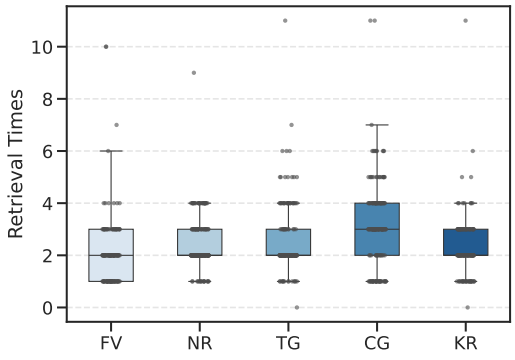
Company Name	Ticker	CIK	S&P 500	Pages
Tesla, Inc.	TSLA	1318605	Yes	733
General Motors Company	GM	1467858	Yes	654
Ford Motor Company	F	37996	Yes	1,063
Rivian Automotive, Inc.	RIVN	1874178	No	433
Oshkosh Corporation	OSK	775158	No	570
Federal Signal Corporation	FSS	277509	No	522
Lucid Group, Inc.	LCID	1811210	No	642
Harley-Davidson, Inc.	HOG	793952	No	713
REV Group, Inc.	REVG	1687221	No	458
Blue Bird Corporation	BLBD	1589526	No	474

Table 11: An Example of 10 Companies in the Automobile Industry (GICS Code: 25102010)

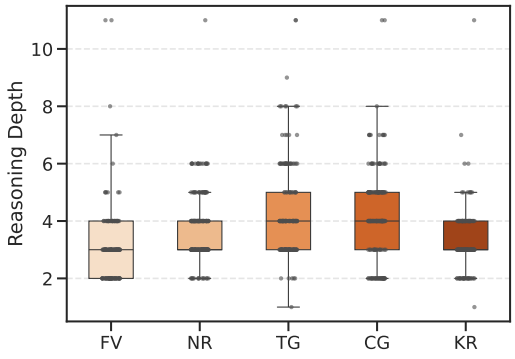
B.2 Task Composition and Dataset Complexity

FinMRAGBench consists of 887 expertly verified question–answer pairs, covering five primary task categories, including *Explainable Fact Verification (FV)*, *Numerical Reasoning (NR)*, *Chart Generation (CG)*, *Table Generation (TG)*, and *Knowledge-Intensive Reasoning (KR)*, which are further divided into 16 fine-grained subcategories, as illustrated in Figure 4. As shown in the figure, the dataset maintains a relatively balanced distribution across the five categories, without any single task type dominating the overall composition. Within each category, multiple subtypes are included to capture diverse reasoning patterns and evidence modalities, ranging from numerical computation and comparison to visual interpretation and knowledge-intensive analysis over financial filings.

Beyond task taxonomy, FinMRAGBench exhibits varying levels of retrieval and reasoning complexity across different task categories. To characterize this complexity, we annotate each question with two difficulty-related measures: *retrieval depth*, defined as the number of distinct evidence retrievals required, and *reasoning depth*, defined as the number of inference steps needed to reach the final answer. Figure 8 shows the distribution of these measures across task categories, highlighting substantial variation in both retrieval and reasoning demands induced by different task types.



(a) Retrieval Times across different tasks



(b) Reasoning depth across different tasks

Figure 8: Retrieval and reasoning complexity of FinMRAGBench.

B.3 Human Verification Protocol and Inter-Annotator Agreement

Annotation protocol. FinMRAGBench adopts an evidence-guided, multi-stage human verification workflow. First, financial experts validate the evidence pages by checking whether the selected pages are necessary and sufficient to answer the question and correcting missing or redundant evidence. During QA construction, candidate questions are manually reviewed under explicit criteria (relevance, completeness, feasibility, clarity, and context-independence), and we further apply round-trip consistency verification to ensure each question genuinely depends on its evidence.

For answer annotation, each question-evidence pair is independently verified by at least two financial experts. Disagreements are resolved through discussion and manual adjudication, and the final answers are checked for correctness and faithfulness to the source documents. This design intentionally prioritizes expert validation and evidence faithfulness rather than lightweight crowd labeling.

Inter-annotator agreement. To further quantify reliability, we conducted an additional verification study on 50 randomly sampled QA pairs following the same protocol. Two financial experts independently judged answer correctness (Correct/Incorrect), and we observed substantial agreement (Cohen's $\kappa = 0.6269$, observed agreement = 92%). This supports that the annotations are reproducible rather than subjective.

C Dataset Construction Details

C.1 Task Definition

Based on a systematic analysis of real-world usage of financial annual reports, we organize the questions into five high-level task categories and 16 finer-grained sub-task types.

Explainable Fact Verification This task category evaluates a model's ability to verify factual claims grounded in financial reports and provide concise, evidence-based explanations. Each question requires the model to determine whether a given claim is *Supported*, *Refuted*, or *Insufficient*, and to justify the decision using retrieved evidence.

- **Numerical Fact Verification** Verifies claims involving explicit numerical values or quantitative comparisons (e.g., revenues, ratios, year-over-year changes). Correct answers require accurate retrieval of relevant figures and precise numerical reasoning across tables or charts.
- **Textual Fact Verification** Verifies claims based on textual statements in financial reports, such as business descriptions, risk disclosures, or accounting policies. This sub-task emphasizes semantic understanding and evidence grounding in narrative sections.

Numerical Reasoning This task category assesses a model's ability to perform quantitative reasoning over financial data retrieved from reports. Questions require extracting numerical values from tables, charts, or text and applying arithmetic or logical operations to derive correct answers.

- **Multi-hop Numerical Reasoning** Requires combining numerical information from multiple pages, documents, or reporting periods to answer a single query, testing cross-document retrieval and multi-step reasoning.
- **Calculation** Involves direct arithmetic operations such as summation, subtraction, ratio computation, or percentage change based on retrieved financial figures.
- **Comparison** Requires comparing numerical values across companies, time periods, or financial indicators to determine relative magnitude or trend.
- **Ranking** Asks the model to order entities (e.g., companies or fiscal years) according to specific financial metrics, requiring accurate aggregation and comparison of multiple values.

Table Generation This task category evaluates a model's ability to synthesize structured tabular outputs from retrieved financial evidence. Models are required to organize relevant numerical information into well-formed tables that align with the query intent and underlying financial semantics.

- **Indicator Computation** Requires computing financial indicators (e.g., ratios or derived metrics) from retrieved values and presenting the results in a structured table format.
- **Information Extraction** Involves extracting relevant financial figures from reports and organizing them into tables without additional

computation, emphasizing accurate retrieval and structural consistency.

Chart Generation This task category evaluates a model's ability to generate visual representations of financial data based on retrieved evidence. Models are required to select appropriate chart types and correctly encode numerical information into executable or structured chart specifications.

- **Pie Chart Generation** Requires generating pie charts to illustrate proportional relationships among financial components.
- **Bar Chart Generation** Involves creating bar charts to compare financial values across categories, such as companies or reporting periods.
- **Line Chart Generation** Focuses on generating line charts to depict trends or temporal changes in financial metrics.
- **Scatter Chart Generation** Requires producing scatter plots to visualize relationships or correlations between two financial variables.
- **Other Chart Types** Covers additional chart formats, including radar charts, stacked bar charts, area charts, and bubble charts, which are grouped for conciseness.

Knowledge-Intensive Reasoning This task category evaluates a model's ability to perform long-form analytical reasoning grounded in retrieved evidence. Questions typically require synthesizing information across multiple pieces of evidence and producing coherent, well-supported explanations rather than short factual answers.

- **Text Evidence** Requires reasoning primarily over narrative text, such as business descriptions, management discussion sections, or risk disclosures.
- **Table Evidence** Involves analytical reasoning based on tabular financial data, requiring interpretation and synthesis of numerical information from tables.
- **Chart Evidence** Requires extracting and reasoning over information conveyed in charts or visual plots, emphasizing multi-modal understanding.
- **Hybrid Evidence** Combines multiple evidence types (e.g., text, tables, and charts), requiring integrated reasoning across heterogeneous modalities.

C.2 Expert-Guided Page Selection

To ensure that selected evidence pages reflect authentic financial analysis practices, financial experts guided page selection based on a set of well-established financial analysis perspectives. These perspectives correspond to common analytical workflows used by practitioners and naturally induce the need for cross-page, cross-document, and multi-modal evidence. We describe the primary perspectives below.

Temporal (Time-Series) Analysis. Time-series analysis examines period-to-period changes in a firm's key financial indicators across reporting periods. Financial experts compare financial data of the same company across different fiscal years or quarters to identify period-specific changes, growth signals, and potential anomalies.

In practice, financial experts collect filings spanning at least three fiscal years for a target company and focus on core indicators such as revenue, net income, cash flow, leverage ratios, and profitability metrics. These indicators are compared using year-over-year and period-over-period analyses, often supported by tables or simple visualizations (e.g., line or bar charts). Observed changes are then interpreted in conjunction with business context, including strategic initiatives, market conditions, and macroeconomic factors.

From a page selection perspective, temporal comparison motivates selecting pages from multiple fiscal periods, including income statements, balance sheets, cash flow statements, and their accompanying explanatory text. As a result, answering temporally grounded questions typically requires aggregating evidence across years and synthesizing both numerical tables and contextual textual explanations.

Cross-Sectional (Industry Peer) Analysis. Cross-sectional (industry peer) analysis assesses a firm's relative performance and competitive position by comparing it with peer companies within the same industry. Financial experts benchmark a target company against comparable firms with similar business models, scale, and market exposure to contextualize financial performance.

In practice, financial experts identify a set of peer companies within the same GICS sector and collect their publicly disclosed filings. Key financial ratios, such as gross margin, net margin, return on equity (ROE), and efficiency indicators, are extracted and organized into comparative tables or visual summaries. These quantitative comparisons are further interpreted alongside qualitative factors, including market share, product differentiation, and brand positioning.

From a page selection perspective, industry-level comparison motivates selecting evidence pages from multiple companies and filings, often drawn from analogous sections across different firms. As a result, questions grounded in industry comparison typically require integrating structured financial tables with narrative disclosures and visual summaries to support informed comparative reasoning.

Ratio-Based Financial Analysis. Ratio analysis is one of the most widely used techniques in financial statement analysis, as it normalizes raw financial figures into interpretable indicators of profitability, liquidity, solvency, and operational efficiency. By transforming absolute values into ratios, this perspective enables meaningful comparisons across firms and across reporting periods.

In practice, financial experts first clarify the analytical objective, such as assessing short-term liquidity, operational efficiency, or overall financial risk, and then extract relevant variables from balance sheets and income statements. Commonly used ratios include liquidity ratios (e.g., current ratio, quick ratio), profitability ratios (e.g., gross margin, net margin), and efficiency ratios (e.g., asset turnover). The computed ratios are benchmarked against industry norms or the firm's historical levels to identify potential weaknesses, strengths, or emerging risks.

From a page selection perspective, ratio analysis requires jointly accessing multiple financial statements together with their supporting notes within the same filing. Interpreting ratios often depends on both numerical tables and accompanying textual explanations, such as accounting policies or management commentary, motivating the inclusion of statement pages and related explanatory text.

Common-Size (Vertical) Structural Analysis. Common-size analysis, also known as vertical analysis or internal structural comparison, examines the internal composition of financial statements by expressing individual line items as percentages of a common base, such as total revenue or total assets. This perspective highlights the relative relationships among cost components, asset categories, and resource allocations within a firm.

In practice, financial experts organize income statements and balance sheets and normalize expense items by total revenue to analyze cost structures, and asset items by total assets to examine balance sheet composition. By comparing these percentage distributions across reporting periods or against industry benchmarks, they can identify structural imbalances, disproportionate spending, or potential inefficiencies in resource allocation. Such analysis is commonly used to assess cost optimization opportunities and strategic prioritization, for example, between sales, research and development, and administrative expenditures.

From a page selection perspective, common-size analysis requires access to complete financial statements rather than isolated figures, together with consistent reporting across sections within the same filing. Interpreting structural proportions often relies on both numerical tables and accompanying narrative disclosures that explain cost drivers or strategic intent, motivating the inclusion of full statement pages and their related explanatory text.

Longitudinal Trend Analysis. Longitudinal trend analysis focuses on identifying long-term directional patterns in a firm's financial performance by continuously tracking key indicators over extended time horizons. Unlike discrete period-to-period comparisons, this perspective emphasizes sustained trajectories and directional changes that are informative for understanding long-run performance dynamics.

In practice, financial experts collect multi-year data from filings, often spanning five to ten years, and focus on indicators such as net income growth, leverage ratios, and operating or free cash flows. These indicators are commonly examined through trend curves or time-series plots to reveal persistent upward or downward movements. Observed trends are then interpreted in light of underlying drivers, including market expansion, cost structure evolution, financing activities, and macroeconomic conditions.

From a page selection perspective, longitudinal trend analysis requires aggregating temporally distributed evidence across multiple filings and reporting periods. Relevant pages are therefore drawn

from different fiscal years and sections, combining numerical tables, visual summaries, and explanatory narrative text to support long-horizon financial reasoning.

Budget-to-Actual Variance Analysis. Budget-to-actual variance analysis is a core management control technique that evaluates execution effectiveness by comparing realized financial outcomes against pre-established budgets. This perspective shifts the focus from reported performance alone to the alignment between planned targets and actual execution.

In practice, financial experts examine budgeted figures for revenues, costs, operating expenses, and capital expenditures alongside their realized counterparts, typically on a periodic basis. Deviations are quantified using variance measures and interpreted through qualitative explanations provided in management disclosures, such as management discussion and analysis (MD&A). This analysis is commonly used to assess operational discipline, cost control effectiveness, and the feasibility of strategic plans.

From a page selection perspective, budget-to-actual analysis requires integrating numerical comparisons with narrative justifications when such information is disclosed. Relevant evidence is therefore drawn from tables and explanatory text in management-facing sections, motivating the inclusion of both quantitative summaries and contextual explanations for variance-driven reasoning.

Earnings–Cash Flow Linkage Analysis. Earnings–cash flow linkage analysis examines the consistency between reported profitability and actual cash generation, reflecting the principle that accounting earnings do not necessarily translate into sustainable liquidity. This perspective is central to financial risk assessment and investment decision-making.

In practice, financial experts jointly analyze income statements and cash flow statements to compare net income with operating cash flows, and contextualize discrepancies using balance sheet items such as accounts receivable and inventory. Persistent gaps between earnings and cash inflows may indicate aggressive revenue recognition, working capital pressure, or heightened financial risk.

From a page selection perspective, earnings–cash flow linkage analysis requires integrating information across multiple financial statements within the same filing, together with relevant explanatory notes. Supporting evidence therefore spans profit figures, cash flow components, and narrative disclosures, motivating the selection of statement pages and accompanying explanatory text for reliable financial judgment.

C.3 Trajectory Synthesis and Filtering

Trajectory Synthesis Protocol To construct training data for Tool-Integrated iterative reasoning, we synthesize multi-step reasoning trajectories using large language models (GPT-5.1) guided by ReAct-style prompts. Each trajectory is generated incrementally, allowing the model to alternate between reasoning and external tool usage as the context evolves.

Concretely, at each iteration t, the model takes as input the accumulated context history and produces one of the following structured actions:

- **`<think>` step**: the agent's intermediate reasoning or planning process, explicitly articulated to guide subsequent actions;
- **`<search>` step**: a tool invocation for retrieving additional evidence from the retrieval system;
- **`<python>` step**: a tool invocation for performing numerical computation or data processing;
- **`<answer>` step**: the final response generation that concludes the trajectory.

Trajectory Quality Filtering To ensure high-quality and effective supervision for training, we apply a multi-stage trajectory filtering procedure to the automatically synthesized reasoning trajectories. Specifically, we perform the following three filtering steps:

1. **Final Answer Verification.** We retain a trajectory τ only if its final answer matches the corresponding ground-truth answer, ensuring that the complete sequence of reasoning and tool-use steps leads to a correct solution.
2. **Step-by-Step Reasoning Consistency.** We verify the logical consistency of each intermediate step in trajectory τ. Each tool call and its corresponding observation are checked to ensure alignment with the preceding reasoning context and the overall problem-solving objective. Trajectories that exhibit incomplete reasoning, incoherent transitions, or incorrect

tool usage are discarded. This step prevents spurious trajectories in which correct answers are obtained without meaningful reasoning.

3. **Complexity-Aware Trajectory Selection.** To encourage substantive multi-step reasoning, we further apply complexity-based filtering. Trajectories with longer and more elaborate reasoning chains are retained, while those with overly short reasoning paths or lacking explicit external tool invocation are filtered out. This criterion promotes training signals that reflect realistic, process-driven analytical workflows rather than trivial or shortcut solutions.

C.4 QA Prompt

To construct high-quality question–answer pairs across diverse financial reasoning tasks, we design task-specific prompts for both question generation and answer annotation. For each of the five task categories in FinMRAGBench, we specify a dedicated question generation prompt that guides the formulation of queries grounded in financial reports, as well as a corresponding answer annotation prompt that enforces correctness, completeness, and evidence grounding.

Question Generation Prompt. The question generation prompts are designed to elicit task-appropriate questions that reflect realistic financial analysis objectives. Each prompt specifies the task scope, required evidence type, and expected reasoning pattern, while avoiding trivial surface-level queries. As summarized in Tables 20–24, the prompts vary across task categories to account for differences in analytical focus, such as factual verification, numerical computation, visual interpretation, structured information extraction, and knowledge-intensive reasoning. Together, these prompts ensure that generated questions span diverse reasoning demands and are grounded in authentic financial reporting scenarios.

Answer Annotation Prompt. For answer annotation, we design prompts that instruct annotators to produce accurate, well-justified answers based strictly on retrieved evidence. The annotation prompts emphasize evidence consistency, correct numerical calculation when applicable, and faithful interpretation of tables or figures. Tables 25–29 present the answer annotation prompts used for the five task categories, highlighting task-specific requirements such as explanation generation, intermediate computation, and structured output formatting. This prompt design helps standardize answer quality across tasks while preserving task-specific characteristics.

D Implementation Details

D.1 Experiments Details

Model Setup. We evaluate a diverse set of multi-modal RAG systems, covering retrieval, generation, and agentic reasoning baselines. For retrieval, we benchmark three multi-modal retrievers: ColQwen2 (Faysse et al., 2025), ColPali (Faysse et al., 2025), and VisRAG-Ret (Yu et al., 2025a). Unless otherwise specified, we use the strongest retriever to retrieve the top-k pages ($k = 10$) for downstream generation. For generation, we evaluate 15 multi-modal large language models, including proprietary models such as GPT-4o (OpenAI, 2023), GPT-5.1 (OpenAI, 2025), Gemini-2.5-Flash, Gemini-2.5-Pro (Comanici et al., 2025), Claude-3.5-Sonnet (Anthropic, 2024), and Claude-4.5-Sonnet (Anthropic, 2025), as well as open-source models from the Qwen2.5-VL (Bai et al., 2025b), Qwen3-VL (Bai et al., 2025a), MiniCPM-V-4.5 (Yu et al., 2025b), and InternVL-3.5 (Wang et al., 2025d) families. Beyond vanilla single-pass RAG, we include advanced baselines with explicit reasoning and tool use, including the ReAct-style IRCOT (Trivedi et al., 2023), the multi-agent visual-document RAG method ViDoRAG (Wang et al., 2025a), and our proposed FinMRAGAgent. IRCOT and FinMRAGAgent are implemented with Qwen3-VL-8B-Instruct, while ViDoRAG is evaluated using GPT-5.1, as its multi-agent prompting relies on stronger backbone models to handle complex financial tasks.

Environment. We deploy open-source models using the vLLM framework on 8 NVIDIA A100 GPUs, while proprietary models are accessed through their official APIs. For all generation experiments, we set the temperature to 0 to ensure deterministic outputs and cap the maximum number of generated tokens at 8,196. For training, we fine-tune Qwen3-VL-8B-Instruct on our high-quality agentic trajectories using LLaMA-Factory[3]. The maximum image resolution is set to 262,144 pixels, and we adopt a cosine annealing scheduler with an initial learning rate of 1e-5, training the model for

[3] https://github.com/hiyouga/LLaMA-Factory

3 epochs.

Training Details. We follow the training paradigm introduced in VRAG (Wang et al., 2025b) to model agentic reasoning with tool use through multi-round generation. Specifically, trajectories that interleave reasoning steps with external tool invocations (e.g., search and Python-based numerical computation) are converted into the ShareGPT format supported by LLaMA-Factory. Each trajectory is represented as a sequence of alternating user and assistant messages, enabling the model to learn iterative decision-making over multiple turns. Under the multi-round formulation, observations returned by the external environment are injected into the trajectory as user messages. This design choice is necessary to align with the model's pre-training interaction protocol, in which image tokens are only allowed to appear in user messages. In total, we construct approximately 6K high-quality training trajectories for supervised fine-tuning.

D.2 Baseline Prompt and Configuration

Vanilla RAG. For Vanilla RAG, we use task-specific answer-generation prompts to produce final responses for each task category. The corresponding prompt templates for the five tasks are provided in Tables 25–29.

IRCOT. For IRCOT, we adapt the original iterative retrieval–reasoning procedure to our multimodal RAG setting and implement it under a ReAct-style interaction protocol that alternates between explicit `think` and `search` steps. In this setting, task-specific instructions are injected into the main IRCOT prompt to adapt the prompt to different task requirements, such as output format and task-specific constraints. The base prompt template for IRCOT is shown in Table 13, while the task-specific prompt components are summarized in Table 15.

ViDoRAG. For ViDoRAG, we follow the original implementation and adopt its three-agent architecture, consisting of a *Seeker Agent*, an *Inspector Agent*, and an *Answer Agent*, each instantiated with the prompt templates provided in the original work. To adapt ViDoRAG to the diverse financial analysis tasks in FinMRAGBench, we inject task-specific instructions into the Answer Agent's prompt, analogous to the task-specific prompting strategy used for IRCOT (Table 15). In our implementation, we instantiate all three agents in ViDoRAG using GPT-5.1, as its multi-agent prompting relies on stronger backbone models for handling complex financial tasks.

FinMRAGAgent. FinMRAGAgent is instantiated with a unified agent prompt that defines the interaction protocol, available actions (e.g., `search`, `python`, `answer`), and response format. The full agent prompt template is provided in Table 14. For each task category, we additionally inject task-specific prompt components into the agent prompt to encourage task-appropriate outputs and constraints; these task-specific components are summarized in Table 15.

D.3 Coarse-to-Fine Retrieval Details

Hierarchical Indexing and Coarse Retrieval To support efficient retrieval from a large-scale financial corpus, we organize documents using hierarchical indexing based on structured document attributes. Each page is associated with metadata such as industry, company, and filing year. Given a query, these attributes are used to constrain the retrieval space before semantic matching, restricting candidate pages to a relevant subset of documents.

This coarse-grained filtering step significantly reduces retrieval ambiguity caused by repetitive table structures and standardized reporting templates across different companies and fiscal periods, while maintaining high recall for relevant evidence.

Fine-grained Page Filtering After coarse-grained retrieval, the candidate set may still contain redundant or weakly relevant pages. To address this, we use a two-stage fine-grained filtering strategy. We first apply a reranker to rerank the retrieved candidate pages according to their query relevance. This reranking stage provides a stronger relevance prior and reduces noise introduced by visually or structurally similar financial pages.

We then apply a prompt-based page filtering stage that assigns a relevance score to each reranked candidate page with respect to the query. The scoring model can either share parameters with the downstream generation model or be instantiated separately, depending on the evaluation setting. Pages are finally re-ranked according to these scores, and only the top-ranked pages are passed to the generation stage. The scoring prompt used for page filtering is provided in Table 16.

D.4 Inference Procedure of FinMRAGAgent

We describe the inference procedure in terms of state representation, action selection, state transition, and termination.

State Representation. During inference, the agent maintains an abstract reasoning state that summarizes the multi-modal input, query, and accumulated interaction history. Following the history-compressed formulation in the main text, the state at step j is defined as

$$s_j = \phi(I, q, t_{\leq j}, a_{<j}, o_{<j}), \tag{4}$$

where t_j denotes the internal reasoning state, a_j the executed action, and o_j the corresponding observation returned by the external environment.

Action Selection Policy. At each inference step j, the agent selects the next action by maximizing the learned policy conditioned on the current state:

$$a_j = \arg\max_{a \in \mathcal{A}} \pi_\theta(a \mid s_j), \tag{5}$$

where the action space is defined as $\mathcal{A} = \{\texttt{search}, \texttt{python}, \texttt{answer}\}$.

State Transition. After executing action a_j, the agent receives an observation o_j from the external environment. The state is then updated as

$$s_{j+1} = \phi_{\text{upd}}(s_j, a_j, o_j), \tag{6}$$

where o_j may correspond to retrieved evidence or numerical computation results.

Termination. The inference process proceeds iteratively until the agent selects the terminal action $a_j = \texttt{answer}$, at which point the final response is generated based on the current state. The abstract inference mechanism described above is instantiated with a coarse-to-fine retrieval strategy in the following subsection.

D.5 Inference with Coarse-to-Fine Retrieval

To efficiently retrieve relevant financial evidence from large-scale multi-modal corpus, FinMRAGAgent integrates a coarse-to-fine retrieval strategy into the inference process. At each `search` step, the agent first infers high-level metadata constraints from the current query and reasoning state, such as industry, company, and fiscal year, to filter the retrieval space. It then applies a multi-modal retriever followed by query-aware re-ranking to identify the most relevant evidence, which is incrementally added to the evidence set. The complete inference procedure is summarized in Algorithm 1.

Algorithm 1 FinMRAGAgent Inference with Coarse-to-Fine Retrieval

Require: Query q, retrieval corpus $\mathcal{C}$, agent policy π_θ

Initialize reasoning state $t_0 \leftarrow \emptyset$, evidence set $\mathcal{E} \leftarrow \emptyset$
for $l = 1$ to $L_{\max}$ **do**
  Generate internal reasoning $t_l \sim \pi_\theta(t_{<l}, \mathcal{E}, q)$
  Predict next action $a_l \sim \pi_\theta(t_{\leq l}, \mathcal{E}, q)$
  if $a_l = \texttt{search}$ **then**
    Infer metadata constraints $\mathcal{M}(q, t_l)$
    $\mathcal{P}_{\text{coarse}} \leftarrow \{p \in \mathcal{C} \mid m(p) \in \mathcal{M}(q, t_l)\}$
    Retrieve candidates using multi-modal retriever (e.g., ColQwen2)
    Re-rank candidates via query-aware scoring
    Update evidence set $\mathcal{E} \leftarrow \mathcal{E} \cup$ Top-$k(\mathcal{P})$
  else if $a_l = \texttt{python}$ **then**
    Execute numerical computation and observe result
    Update reasoning state with computation output
  else if $a_l = \texttt{answer}$ **then**
    Generate final answer and **break**
  end if
end for
return Final answer

E Evaluation Metrics Details

E.1 Retrieval Quality

To evaluate retrieval quality, we report Recall@10, which measures the coverage of relevant evidence within the top-10 retrieved pages. This metric reflects whether the retrieval module successfully captures the required evidence for downstream generation in realistic financial RAG settings.

E.2 Answer Accuracy

Due to the substantial diversity in answer formats across financial analysis tasks, we adopt task-specific evaluation metrics, including both automatic metrics and LLM-based evaluation. For all LLM-based evaluations, we additionally sample a subset of instances for human verification to validate the reliability of the automatic judgments.

Explainable Fact Verification For explainable fact verification, we evaluate both the predicted veracity label and the accompanying explanation.

The veracity label is a three-class classification problem (*Supported*, *Refuted*, and *Insufficient*) and can be evaluated using Exact Match (EM). In this work, we primarily focus on the quality of the generated explanations, which better reflects a model's ability to reason over and justify financial claims. Accordingly, our main reported metrics for this task are **LJS** and **HQS**, both derived from an LLM-as-judge evaluation protocol.

Inspired by the human evaluation design of FinDVer (Zhao et al., 2024), we introduce an LLM-based evaluation to assess explanation quality. Specifically, a large language model scores each generated explanation on a discrete scale from 1 to 10 based on factual correctness, reasoning soundness, and consistency with the reference answer. The resulting score is normalized to a percentage and reported as **LJS** in the main results table. We further report **HQR**, defined as the proportion of instances with scores greater than or equal to 7. The detailed judging prompt is provided in Table 17.

Numerical Reasoning For numerical reasoning tasks, we report F1 as the primary automatic metric, following MULTIHIERTT (Zhao et al., 2022). Compared to Exact Match, F1 better captures partial correctness in financial scenarios where answers often contain multiple numerical values rather than a single scalar.

Given the open-retrieval setting and frequent use of natural language descriptions, we employ an LLM-as-judge evaluation following ViDoRAG (Wang et al., 2025a), which assigns a discrete score from 1 to 5 to each generated answer based on numerical correctness and consistency with the reference answer. We define **HQR** as the proportion of predictions with a score greater than or equal to 4. The judging prompt used for this evaluation is provided in Table 19.

Table Generation For table generation, model outputs are first normalized into a canonical table format. We then evaluate them using the RMS metric proposed in DePlot (Liu et al., 2023a) and FinAR-Bench (Wu et al., 2025b), which jointly measures structural alignment and value accuracy between the generated and ground-truth tables. Detailed computation steps are provided in Appendix E.3.

Chart Generation For chart generation, we follow RealHitBench (Wu et al., 2025a) and evaluate both executability and data correctness. Specifically, we compute ECR to measure whether the generated code is executable, extract the resulting y-axis values for comparison with reference data, and report PASS@1 as the overall success rate.

Knowledge-Intensive Reasoning Knowledge-intensive reasoning requires models to generate long-form analytical responses. We evaluate answer quality using an LLM-as-judge protocol based on G-Eval (Liu et al., 2023b).

For each generated long-form answer, the LLM assigns a score from 1 to 10 based on factual correctness, reasoning coherence, and overall answer quality with respect to the reference answer. The normalized score is reported as **LJS** in the main results table. We further report **HQR**, defined as the proportion of responses receiving a score of at least 7, indicating satisfactory analytical quality. The full judging prompt is provided in Table 18.

E.3 RMS Metric for Table Generation

We adopt the RMS metric for table generation evaluation, following the formulation in FinAR-Bench, with a minor modification to better emphasize numerical accuracy. The RMS metric jointly evaluates table structure alignment and numerical correctness by matching predicted and ground-truth table entries.

Data Point Representation. Each table is represented as a collection of data points, where each point corresponds to a (row header, column header, value) tuple. The row and column headers are concatenated to form a unique key (e.g., *"Sales Expense 2022"*).

Header Matching via Textual Distance. To match predicted and ground-truth headers, we measure their similarity using the normalized Levenshtein distance:

$$\mathrm{NL}(pr\|pc,\ tr\|tc) = \frac{\mathrm{edit\_distance}(pr\|pc,\ tr\|tc)}{\max\big(\mathrm{len}(pr\|pc),\ \mathrm{len}(tr\|tc)\big)}. \quad (7)$$

Pairs with distance exceeding a threshold τ are assigned a unit cost. The resulting cost matrix is then used to perform optimal one-to-one matching via the Hungarian algorithm (Kuhn, 2010).

Numerical Error Measurement. Given an aligned pair of values, we quantify numerical dis-

crepancy using the relative error:

$$D_\theta(p,t) = \begin{cases} \frac{|p-t|}{|t|}, & \text{if } \frac{|p-t|}{|t|} \le \theta \\ 1, & \text{otherwise,} \end{cases} \tag{8}$$

where p and t denote the predicted and ground-truth numerical values.

Final RMS Precision and Recall. Following FinAR-Bench, the textual distance term is excluded from the final score to focus the evaluation on numerical accuracy after alignment. Specifically, we define:

$$D_{\tau,\theta}(p,t) = \begin{cases} D_\theta(p,t), & \text{if NL}(pr \| pc,\ tr \| tc) \\ & \le \tau, \\ 1, & \text{otherwise.} \end{cases} \tag{9}$$

The final RMS Precision and RMS Recall are computed as:

$$\text{RMS}_{\text{Precision}} = 1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} X_{ij} D_{\tau,\theta}(p_i,t_j)}{N}, \tag{10}$$

$$\text{RMS}_{\text{Recall}} = 1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} X_{ij} D_{\tau,\theta}(p_i,t_j)}{M}, \tag{11}$$

where X_{ij} denotes the assignment matrix obtained from the Hungarian algorithm, N is the number of predicted data points, and M is the number of ground-truth data points.

E.4 Human-LLM Agreement for LJS

To further examine the reliability of LJS for open-ended financial reasoning tasks (FV and KR), we conducted a human–LLM agreement analysis.

On a randomly sampled subset of 50 QA pairs, financial experts independently rated answer quality on a 1–10 scale following the evaluation rubric, without access to the LJS outputs. We then compared LJS scores against the human ratings. The LLM judge demonstrates strong agreement with human evaluators (quadratic weighted Cohen's κ = 0.9112), indicating that its assessments are closely aligned with human judgment. As shown in Table 12, most ratings concentrate near the diagonal, indicating that LJS scores closely track human ratings rather than exhibiting systematic disagreement.

We additionally examined score stability and calibration. The mean difference between LJS and human scores is 0.42 with a standard deviation of 1.25 (on a 1-10 scale), suggesting a slight tendency of LJS to assign marginally higher scores, but with limited deviation relative to the rating range. The overall dispersion of scores remains comparable, indicating stable evaluation behavior rather than erratic scoring.

Human \ LLM	1	2	3	4	5	6	7	8	9	10
1	2	5	0	1	0	0	0	0	0	0
2	0	1	1	0	0	0	0	0	0	0
3	0	1	1	3	1	0	0	0	0	0
4	0	0	2	1	1	1	0	0	0	0
5	0	0	0	1	1	0	1	0	0	0
6	0	0	0	1	0	0	0	1	0	1
7	0	0	0	0	0	1	1	2	0	0
8	0	0	0	0	0	0	1	1	2	2
9	0	0	0	0	0	0	2	1	2	2
10	0	0	0	0	0	0	0	0	1	5

Table 12: Human-LLM agreement matrix for LJS on 50 randomly sampled open-ended QA pairs. Rows denote human expert scores and columns denote LJS scores on a 1-10 scale.

F Case Study

Case Study: Retrieval Bias under Structural and Semantic Similarity. We analyze a representative example from FinMRAGBench that requires cross-document, multi-page reasoning over multi-modal financial evidence. As shown in Figure 9, the vanilla RAG baseline is misled by superficial structural and semantic similarity across financial reports, retrieving pages from incorrect companies or fiscal years with similar layouts and table schemas. This leads to incompatible evidence aggregation and incorrect conclusions. In contrast, as illustrated in Figure 10, FinMRAGAgent performs multi-round, interaction-driven retrieval guided by intermediate reasoning states, incrementally expanding and aligning evidence across entities and time periods. By coherently integrating textual, tabular, and visual information, the agent produces the correct final answer.

Case Study: Tool Use and Numerical Reasoning Efficiency. We examine a numerical reasoning example that requires retrieving net income values across multiple fiscal years and performing explicit growth-rate computation. As shown in Figure 11, the IRCOT baseline enhanced with coarse-to-fine retrieval is able to progressively collect relevant evidence through stepwise search. However, due

to inefficient and redundant use of the search tool, it performs multiple unnecessary retrievals and ultimately relies on implicit reasoning without invoking a computation tool, leading to an incorrect numerical result. In contrast, as illustrated in Figure 12, FinMRAGAgent learns when and how to invoke tools effectively, retrieving the required evidence with fewer search steps and explicitly performing growth-rate computation via a Python interpreter. This enables the agent to produce an accurate and verifiable numerical answer.

Case Study: Diverse Task Formulations in Financial RAG. Beyond numerical reasoning, we further present representative examples from the remaining tasks in FinMRAGBench, including Explainable Fact Verification, Table Generation, Chart Generation, and Knowledge-Intensive Reasoning. As illustrated in Figures 13–16, these tasks require models to reason over heterogeneous financial evidence spanning text, tables, and charts, and to produce structured or explanatory outputs grounded in retrieved documents. Together, these cases highlight the diversity of task formulations and the breadth of reasoning skills evaluated by FinMRAGBench.

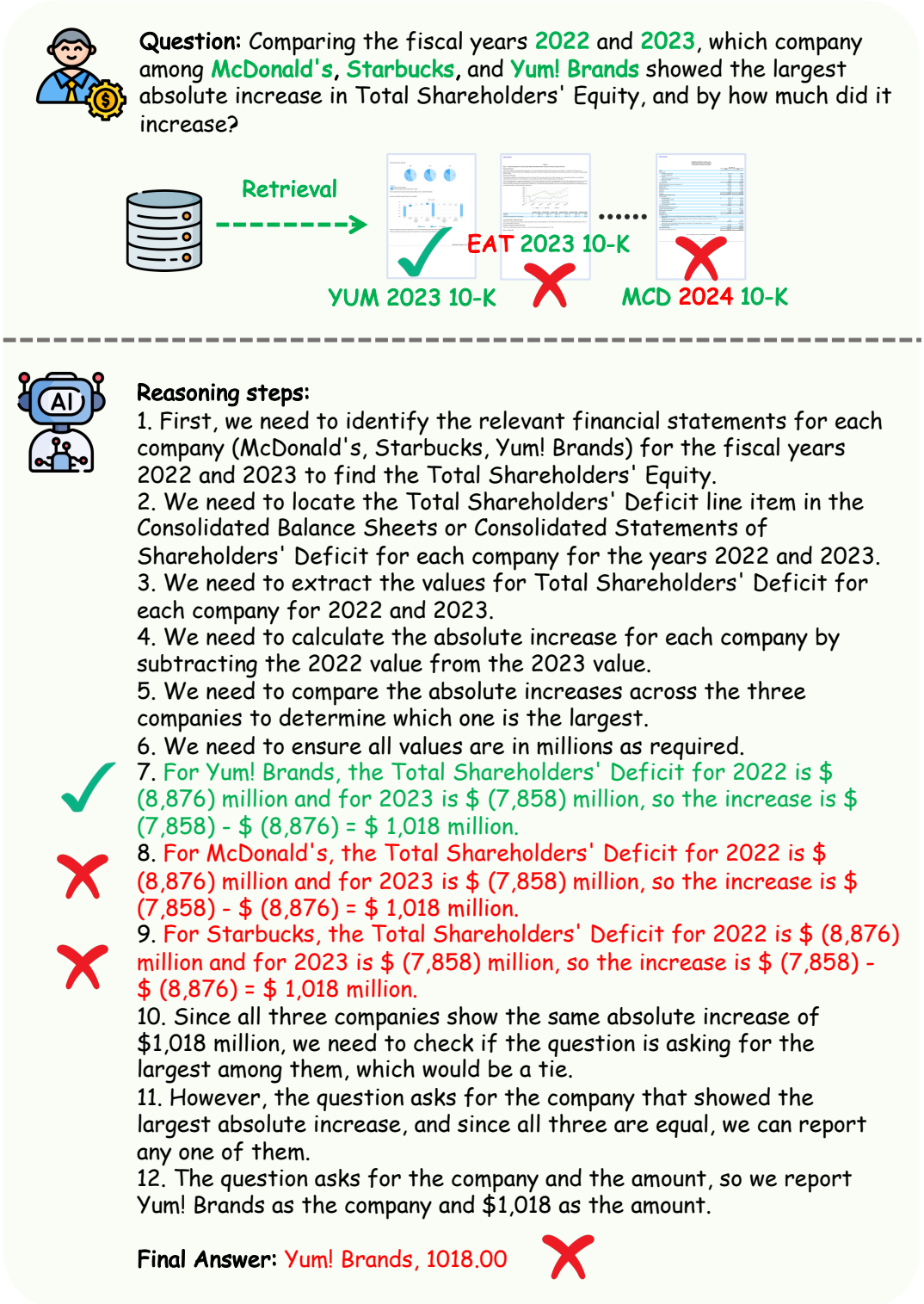


Figure 9: Case example of the vanilla RAG baseline, illustrating retrieval errors caused by structural and semantic similarity across financial reports.

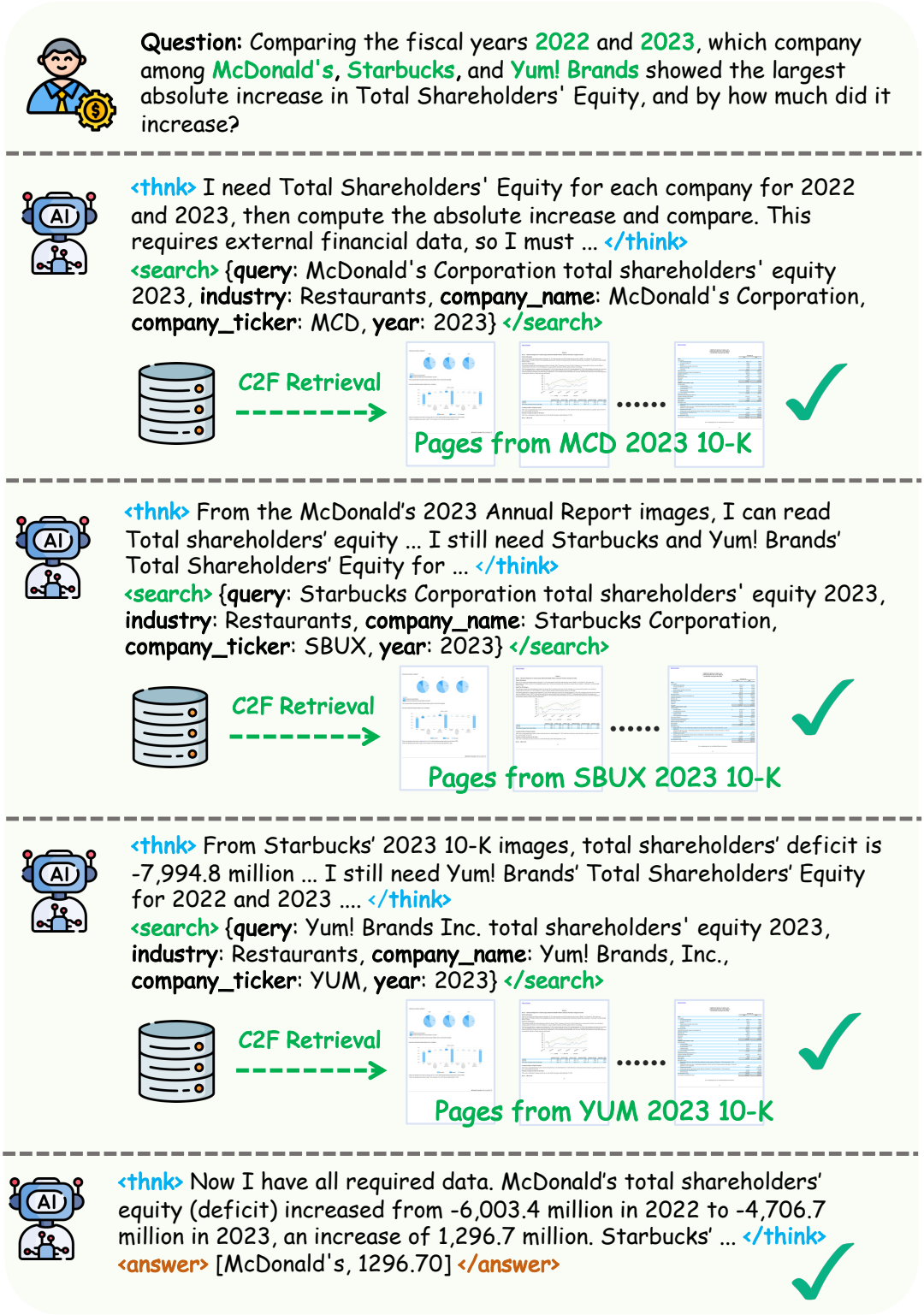


Figure 10: Case example of FinMRAGAgent, demonstrating multi-round retrieval and evidence alignment across companies and fiscal periods.

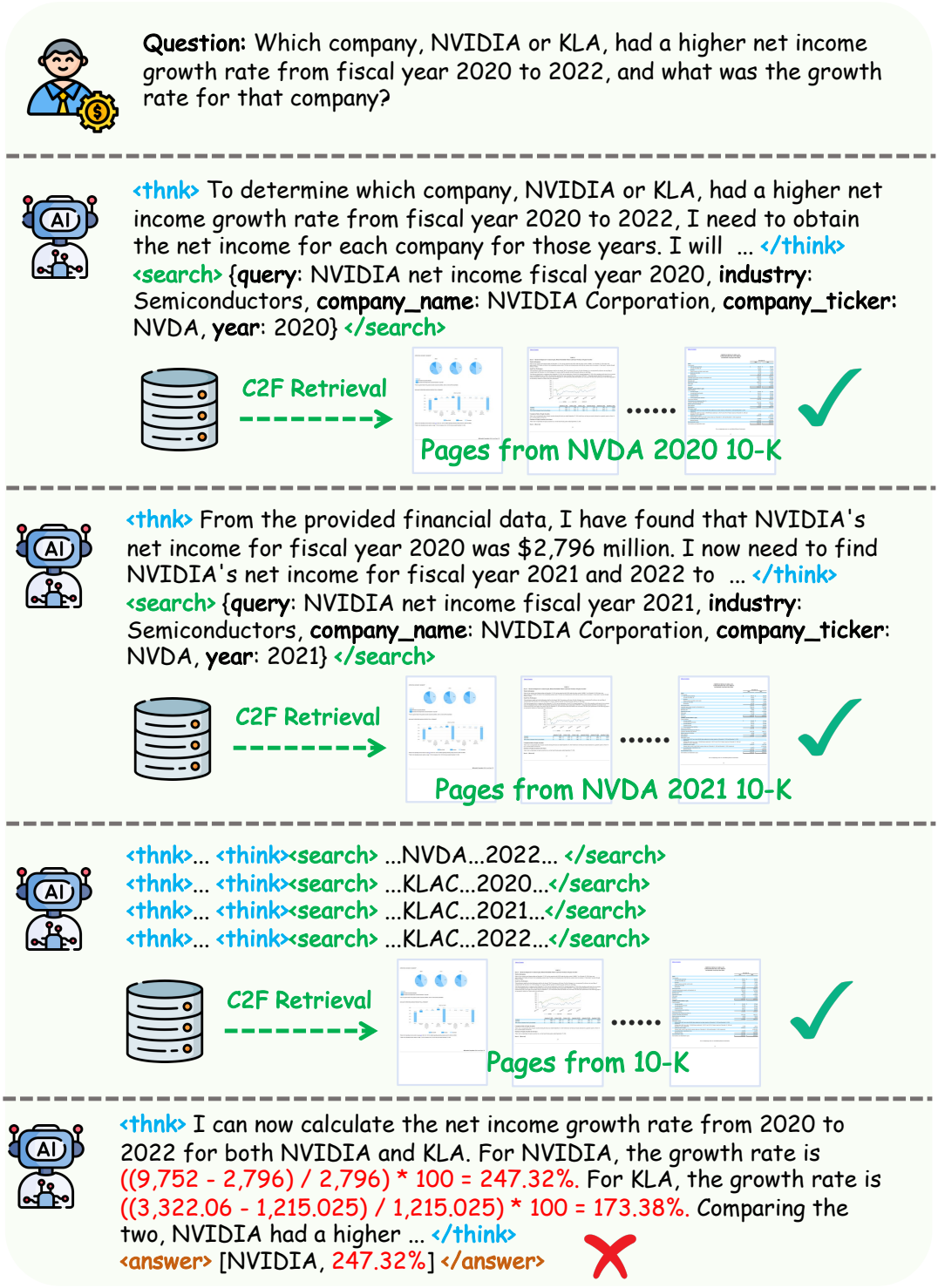


Figure 11: Case example of the IRCOT baseline with coarse-to-fine retrieval, showing redundant search steps and incorrect numerical reasoning without explicit computation.

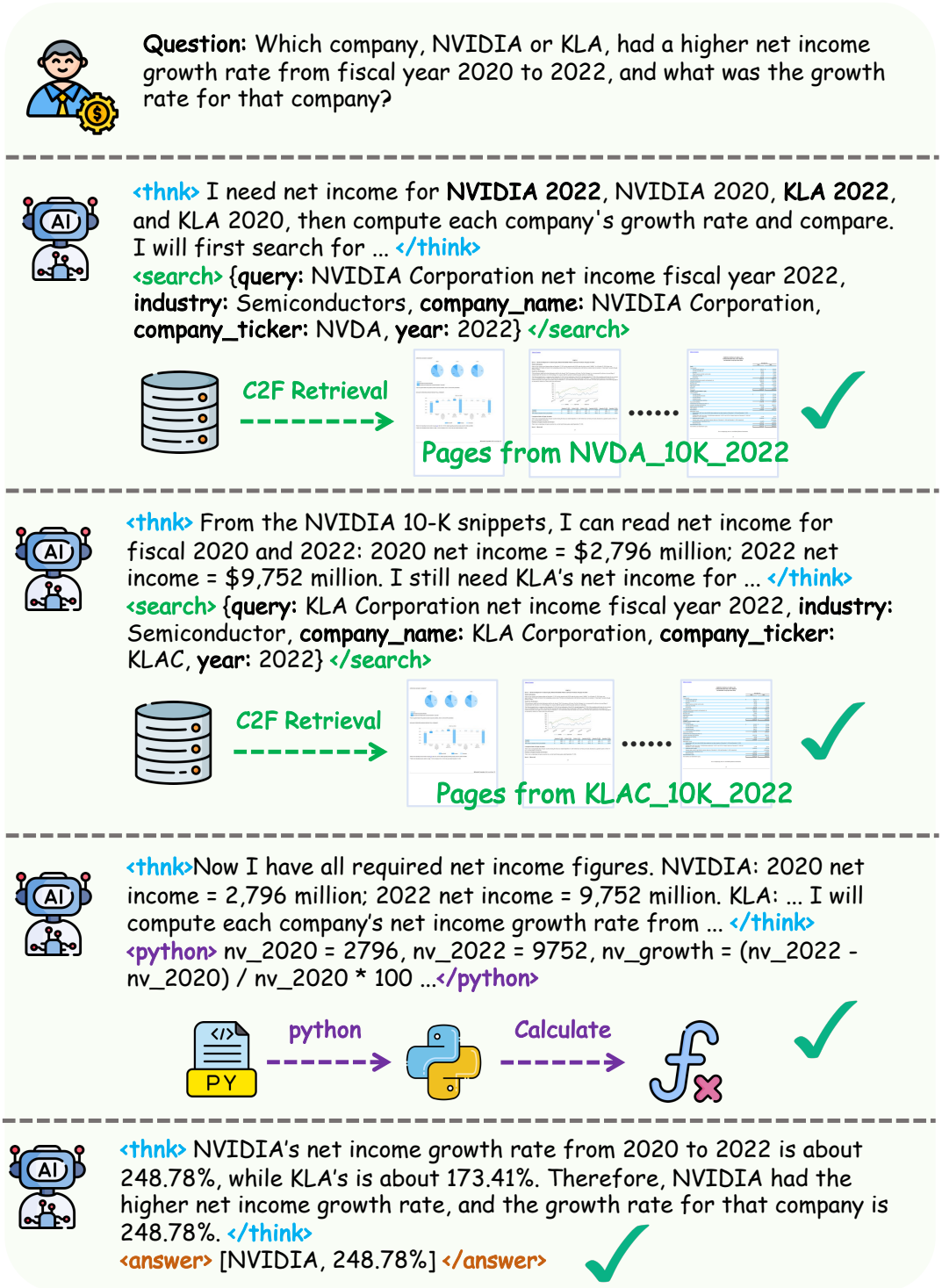
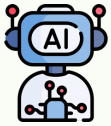


Figure 12: Case example of FinMRAGAgent, illustrating efficient tool invocation and accurate numerical computation with fewer retrieval steps.

Claim: PNC reported consolidated total assets of approximately $426 billion at year-end 2024, which was lower than both Bank of America and M&T Bank's reported consolidated totals.

Subtype: Numerical Fact Verify

Evidence_Pages: ["PNC_10K_2024_50.jpg", "MTB_10K_2024_120.jpg", "BAC_10K_2024_93.jpg"]



Answer: Refuted

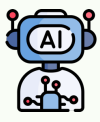
Explanation: PNC reported $560.0 billion in total assets at December 31, 2024. This is higher than M&T Bank's $208.1 billion but lower than Bank of America's multi-trillion-dollar consolidated assets. Therefore, the statement that PNC had approximately $426 billion and was lower than both banks is incorrect."

Figure 13: An example of the Explainable Fact Verification task.

Question: Calculate McDonald's Corporation's financial indicators for fiscal year 2023, including Total Revenues, Net Profit Margin, Debt-to-Asset Ratio, and Earnings Per Share (Diluted), and output the results in a markdown table.

Subtype: Indicator Computation

Evidence_Pages: ["MCD_10K_2023_38.jpg", "MCD_10K_2023_40.jpg"]



Answer:

Indicator	2023
Total Revenues	25493.7
Net Profit Margin	0.3322
Debt-to-Asset Ratio	1.0839
Earnings Per Share (Diluted)	11.5600

show

Figure 14: An example of the Table Generation task.

Question: Please draw a line chart to compare the Net income of McDonald's, Starbucks, and Yum! Brands from fiscal year 2018 to fiscal year 2023.

Subtype: LineChart Generation

Evidence_Pages: ["MCD_10K_2020_38.jpg", "MCD_10K_2023_38.jpg", "SBUX_10K_2020_50.jpg", "SBUX_10K_2023_45.jpg", "YUM_10K_2020_61.jpg", "YUM_10K_2023_56.jpg"]

Answer:

```
import pandas as pd
import matplotlib.pyplot as plt

years = [2018, 2019, 2020, 2021, 2022, 2023]

mcdonalds_net_income = [5924.3, 6025.4, 4730.5, 7545.2, 6177.4, 8468.8]
starbucks_net_income = [4518.3, 3599.2, 928.3, 4199.3, 3281.6, 4124.5]
yum_brands_net_income = [1542, 1294, 904, 1575, 1325, 1597]

plt.figure(figsize=(10, 6))
plt.plot(years, mcdonalds_net_income, marker='o', label="McDonald's")
plt.plot(years, starbucks_net_income, marker='o', label="Starbucks")
plt.plot(years, yum_brands_net_income, marker='o', label="Yum! Brands")

plt.title('Net Income Comparison (2018-2023)')
plt.xlabel('Year')
plt.ylabel('Net Income (in millions)')
plt.xticks(years)
plt.legend()
plt.grid(True)
plt.show()
```

draw

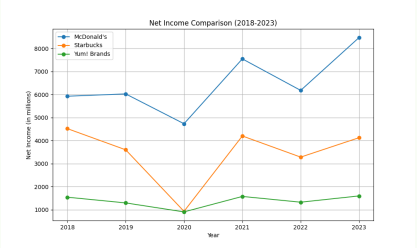
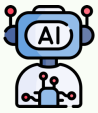


Figure 15: An example of the Chart Generation task.

Question: According to Tesla's 2023 annual report, what is the company's attitude toward research and development (R&D) investment, what innovation strategies does it pursue, and is its R&D spending primarily expensed or capitalized? Would this accounting treatment be considered aggressive?

Subtype: Hybrid Evidence

Evidence_Pages: ["TSLA_10K_2023_34.jpg", "TSLA_10K_2023_50.jpg", "TSLA_10K_2023_51.jpg"]



Answer: Tesla's 2023 annual report indicates a strong commitment to research and development (R&D) as a key component of its strategy to accelerate the world's transition to sustainable energy. The company emphasizes its focus on increasing vehicle production, enhancing battery technologies, and improving software capabilities, particularly in artificial intelligence, robotics, and automation. This strategic focus is evident in their continued investment in R&D to accelerate AI, software, and fleet-based profits. In terms of financial treatment, Tesla's R&D spending is primarily expensed, as reflected in the operating expenses section of the income statement, where R&D costs are listed as $3,969 million for 2023. This approach is generally considered conservative rather than aggressive, as expensing R&D costs immediately impacts the income statement, reducing net income, rather than capitalizing them, which would spread the costs over future periods and potentially inflate current earnings.

Figure 16: An example of the Knowledge-Intensive Reasoning task.

Prompt for IRCOT

Answer the given question. You must conduct reasoning inside <thinking> and </thinking> first every time you get new information. After reasoning, if you find you lack some knowledge, you can call a search engine by <search> query </search> and user will return the searched results. You can search as many times as your want.
Workflow:

1. Always write <thinking>. . . </thinking>.
2. Then output either:
   - <search>{...json...}</search> (ONLY ONE per turn), OR
   - <answer>. . . </answer>.
3. Do NOT output both <search> and <answer>.

IMPORTANT SEARCH FORMAT:
When you use <search>, the content inside MUST be a single valid JSON object in the following schema:

```
{
  "query": "A clear, atomic, and retrieval-ready search query.",
  "industry": "The company's primary industry sector.",
  "company_name": "The full official name of the company.",
  "company_ticker": "The company's stock ticker symbol.",
  "year": "The specific fiscal year for the metric."
}
```

Rules:

- Exactly ONE company, ONE fiscal year, and ONE simple metric per <search>.
- The JSON must be directly parseable (no markdown fences, no extra text).
- If multiple years/metrics are needed, issue multiple <search> calls.

If you find no further external knowledge needed, you can directly provide the answer inside <answer> and </answer>, without detailed illustrations. {specific_prompt} Question: {question}

Table 13: Prompt for IRCOT.

Prompt for FinMRAGAgent

Answer the given question. You must conduct reasoning inside <think> and </think> first every time you get new information. After reasoning, if you find you lack some knowledge, you can call a search engine by <search> query </search> and user will return the searched results. You can search as many times as your want. If you need to perform calculations, you can call the python interpreter by <python> code </python> and user will return the execution result. If you find no further external knowledge needed, you can directly provide the answer inside <answer> and </answer>, without detailed illustrations. {specific_prompt} Question: {question}

Table 14: Prompt for FinMRAGAgent.

Task-specific Prompt

Numerical Reasoning: For numerical reasoning tasks, your final answer inside <answer> MUST be a valid JSON array of strings. The number of items in the array MUST match the number of required outputs for the question. Each string MUST be the final value only (a number or entity name), as short as possible, without any explanation. Keep percentage signs. If a final value has decimals, retain two decimal places. You MUST NOT skip or omit search before answering.
Table Generation: For table generation tasks, your final answer inside <answer> MUST be formatted as a valid Markdown table, with no additional explanations, comments, or text outside the table. You MUST NOT skip or omit search before answering.
Chart Generation: For chart generation tasks, the plotting code inside <answer> MUST follow the specified template: import pandas as pd \n import matplotlib.pyplot as plt \n ... plt.show(). You MUST NOT skip or omit search before answering.
Explainable Fact Verification: For explainable fact verification tasks, your final answer inside <answer> MUST be a valid JSON object with the following structure:

```
{
  "answer": "Supported / Refuted / Insufficient",
  "explanation": "A concise explanation referencing the retrieved evidence."
}
```

The answer field MUST contain exactly one of: Supported, Refuted, or Insufficient. The explanation MUST be concise and based solely on retrieved evidence. You MUST NOT skip or omit search before answering.
Knowledge-Intensive Reasoning: For knowledge reasoning tasks, your final answer inside <answer> MUST be a concise but complete response in one or two short paragraphs, clearly combining reasoning and conclusion without unnecessary elaboration. You MUST NOT skip or omit search before answering.

Table 15: Task-specific prompt templates. Each task type is associated with a dedicated `specific_prompt` that enforces strict output formats and mandatory evidence retrieval.

Prompt for Page Selection

You are reviewing one page from a company's financial report. Your task is to evaluate how informative this page is for answering the following question.
Question: {query}
Output strictly in JSON format (no markdown, no explanation outside JSON):

```
{
  "score": int,
  "rationale": "Brief reason (<=20 words)"
}
```

Scoring Guidelines:

- **0–2**: Irrelevant (e.g., title page, table of contents, disclaimers)
- **3–4**: Marginal (mentions section names or headers only, no real data)
- **5–6**: Somewhat relevant (mentions related concepts but lacks specific values)
- **7–8**: Useful (contains partial data, figures, or textual discussion relevant to the question)
- **9–10**: Highly useful (contains detailed tables, numerical values, or clear evidence directly answering the question)

Rules:

- Focus on whether the page provides *substantive financial information* (numbers, ratios, results, tables).
- Do NOT assign a high score if the page only lists report sections or references other pages.
- Output only valid JSON and nothing else.

Table 16: Prompt for Page Selection.

LLM Judge Prompt for Explainable Fact Verification

You are an expert evaluator of financial reasoning explanations.
You are given:

- A financial claim
- A human-written gold explanation that correctly verifies the claim
- A model-generated explanation for the same claim

Your task is to evaluate whether the model explanation is logically consistent with the gold explanation and whether it captures the correct reasoning needed to justify the claim. Compare the reasoning structure, key analytical steps, and conclusion. Evaluate the model explanation based on the following criteria:

- **Correctness**: Does the explanation reach the correct conclusion regarding the claim?
- **Logical Alignment**: Is its reasoning process consistent with that of the gold explanation?
- **Completeness**: Does it include the essential reasoning steps presented in the gold explanation?
- **Clarity**: Is the explanation easy to understand and well-organized?

Scoring Guidelines (10-point scale):

- **1–2**: The explanation is irrelevant, incorrect, or fundamentally flawed.
- **3–4**: Marginal relevance; contains major reasoning gaps or misunderstandings.
- **5–6**: Partially correct; some alignment with the gold reasoning but missing key steps.
- **7–8**: Mostly correct and aligned; minor omissions but sound reasoning overall.
- **9–10**: Highly accurate, logically aligned, complete, and clearly articulated.

Output Format:

```
{
    "rationale": "your rationale for the score, as a text",
    "score": "your score from 1 to 10"
}
```

Inputs:

- **Claim**: `{claim}`
- **Gold Explanation**: `{gold_explanation}`
- **Model Explanation**: `{model_explanation}`

Table 17: LLM Judge Prompt for Explainable Fact Verification.

LLM Judge Prompt for Knowledge-Intensive Reasoning

You are an expert evaluator of financial question-answering systems.
You are given:

- A financial question
- A human-annotated gold answer
- A model-generated answer to the same question

Your task is to evaluate the model answer by comparing it with the gold answer, focusing on correctness, reasoning completeness, and clarity.
Evaluate the model answer on the following criteria:

- **Correctness**: Does the model answer provide factually accurate, financially sound, and contextually appropriate information?
- **Logical Consistency**: Is the reasoning coherent, logically valid, and aligned with the intent of the question?
- **Coverage**: Does the answer fully and directly address all components of the question, matching the breadth and depth of the gold answer?
- **Clarity**: Is the explanation clearly written, well-structured, and easy for a financially knowledgeable audience to understand?

Scoring Guidelines (10-point scale):

- **1–2**: Irrelevant, incorrect, or fundamentally flawed; does not address the question meaningfully.
- **3–4**: Partially relevant but contains major factual errors, omissions, or logical issues.
- **5–6**: Moderately correct; captures some key ideas but lacks important details or clarity compared with the gold answer.
- **7–8**: Mostly correct and well-aligned with the gold answer; minor omissions or slight reasoning gaps.
- **9–10**: Highly accurate, comprehensive, logically consistent, and clearly presented; closely matches or fully meets the standard of the gold answer.

Output Format:

```
{
    "rationale": "your rationale for the score, as a text",
    "score": "your score from 1 to 10"
}
```

Inputs:

- **Question**: `{question}`
- **Gold Answer**: `{gold_answer}`
- **Model Answer**: `{model_answer}`

Table 18: LLM Judge Prompt for Knowledge-Intensive Reasoning.

LLM Judge Prompt for Numerical Reasoning

You are an expert evaluation system for a question answering chatbot.
You are given the following information:

- A user query and a reference answer
- A generated answer

You may also be given a reference answer to use for reference in your evaluation.
Your job is to judge the relevance and correctness of the generated answer. Output a single score that represents a holistic evaluation.
You must return your response in a line with only the score. Do not return answers in any other format. On a separate line, provide your reasoning for the score as well.
Follow these guidelines for scoring:

- The score must be between **1** and **5**, where **1** is the worst and **5** is the best.
- If the generated answer is not relevant to the user query, assign a score of **1**.
- If the generated answer is relevant but contains mistakes, assign a score between **2** and **3**.
- If the generated answer is relevant and fully correct, assign a score between **4** and **5**.
- When comparing numerical values in the generated answer with the reference answer, allow for a reasonable margin of error; small deviations should not be penalized as mistakes.

Example Response:

```
4.0
The generated answer has the exact same metrics as the reference answer, but it is not as concise.
```

Inputs:

- **User Query**: `{query}`
- **Reference Answer**: `{reference_answer}`
- **Generated Answer**: `{generated_answer}`

Table 19: LLM Judge Prompt for Numerical Reasoning.

Question Generation Prompt for Explainable Fact Verification

 # Role Play
Suppose you are a senior financial annual report analysis expert and your task is to generate high-quality fact verification question–answer pairs based on annual report tables and textual excerpts.

 # Task Description
Fact verification tasks in financial report analysis involve checking whether a given natural language statement is consistent with the information contained in annual reports. Each statement should be specific to certain metrics, company names, fiscal years, or textual disclosures. To simulate real-world analyst workflows, each statement must depend on information that spans across multiple report sections, fiscal years, or companies, requiring retrieval from more than one page or document. The answer to each statement must be one of the following three categories:

- **Supported**: the statement is fully confirmed by the evidence in the annual report.
- **Refuted**: the statement contradicts the evidence in the annual report.
- **Insufficient**: the evidence in the annual report is not enough to judge the statement as true or false.

The input consists of one or more tables or text excerpts from annual reports. The generated statements must require retrieval of relevant information across these inputs, and the correctness must be determinable only with reference to the provided evidence.

 # Generation Restrictions
To generate five questions based on the given task description and multi-source financial tabular data, give due consideration to the following aspects:

- **Complexity**: Statements must not rely on a single data point or a single page. Each statement should require combining information across multiple companies, multiple fiscal years, or multiple report pages or documents. Include both simple factual checks and more complex reasoning such as multi-step calculations, trend verification, or multi-document textual reasoning.
- **Length**: Each statement should be between 15 and 40 words.
- **Diversity**: Cover two distinct types of fact verification tasks:
  - **Numerical Verification**: Statements that involve verifying financial indicators, which may come from tables or from textual descriptions within the annual report, either by direct extraction or through more complex calculations.
  - **Textual Verification**: Statements that involve verifying qualitative or narrative information from any part of the annual report text, including strategic goals, risk disclosures, management discussion, or other narrative sections.
- **Real-World Relevance**: Statements must reflect realistic checks that analysts or auditors would perform.
- **Writing Style**: Use professional and precise financial terminology. Explicitly mention metrics, company names, fiscal years, or textual disclosure subjects.
- **Balance**: Generate statements that are evenly distributed across the three verification labels (Supported, Refuted, Insufficient), and also balanced between Numerical Verification and Textual Verification. Refuted and Insufficient statements must remain realistic and plausible.
- **Context-Independent**: Each statement must be understandable on its own without external context.

 # Output Control
Please generate the output as a strict JSON object in the following format:

```
{
  "result":[
    {
      "question": "Your fact verification statement here",
      "answer": "Supported or Refuted or Insufficient",
     "explanation": "A brief explanation of why the answer is Supported, Refuted, or Insufficient",
      "subtype": "Numerical Fact Verify or Textual Fact Verify",
      "evidence_pages": [" ", " "]
    }, ...
  ]
}
```

Table 20: Prompt for Generating Explainable Fact Verification Questions.

Question Generation Prompt for Numerical Reasoning

 # Role play
Suppose you are a senior annual report analysis expert and your task is to generate high-quality and diverse numerical reasoning questions that can only be answered through retrieval and synthesis across multiple pages or documents of company financial reports.

 # Task Descriptions
Numerical reasoning tasks in financial report analysis involve interpreting and calculating with numerical data found in structured report excerpts, such as income statements, balance sheets, and KPIs across different companies and fiscal years. These tasks typically require retrieving relevant values from multiple report sections, recognizing trends, comparing performance, or evaluating quantitative shifts across time or entities. Each question must require multi-source retrieval to answer, simulating a real-world retrieval-augmented analysis workflow.
The input consists of one or more images from official annual reports. These images may represent different pages from the same or different companies, and may span multiple fiscal years. Based on the type of input images, design questions that match the appropriate comparison mode — use cross-company horizontal comparisons when images are from different companies, and cross-year vertical analyses when images span multiple fiscal years of the same company.

 # Generation Restrictions
To generate five questions based on the given task description and multi-source financial textual or tabular data, give due consideration to the following aspects:

- **Complexity:** Include a range of problem complexities. Simpler questions may involve a direct comparison or percentage difference. More complex questions should require multi-hop reasoning that spans across multiple documents, companies, years, and logically connected metrics — for example, identifying the company with the highest value in one metric, then retrieving another metric for that specific entity (but still be expressed as a single complete sentence).
- **Length:** Questions should be between 20 and 50 words. Keep language compact but fully informative.
- **Diversity:** The subtypes covered by the questions should include: Multi-hop Numerical Reasoning, Ranking, Comparison, and Calculation (Numerical Calculation and Time-based Calculation).
- **Real-World Relevance:** Questions must reflect the type of financial inquiry a professional analyst would perform when reviewing annual reports — such as identifying revenue leaders, cost-saving trends, or changes in capital structure.
- **Writing-Style:** Use professional, clear, and consistent financial terminology. Avoid vague wording. Specify the target metric, entities, or years clearly.
- **Question Form:** The question must be phrased as a single, grammatically complete sentence, even if it involves multi-hop reasoning. Do not split the question into multiple sub-questions.
- **Context-Independent Question:** Each question must be fully self-contained and context-independent. The question must explicitly mention all necessary information, including the specific metric name, company name or entity names, and fiscal year(s), so it can be understood and answered in isolation within a retrieval-augmented setting (e.g., a question that specifies the metric, such as revenue; the companies involved, such as Company A, B, and C; and the time frame, such as fiscal years 2021 to 2023).
- **Answer Control:** Answers must follow the format "AnswerName" or "AnswerName1, AnswerName2. . .". "AnswerName" should be a numeric value or company/entity name — short and unambiguous.
- **Retrieval Dependency:** Ensure that each question requires combining information from multiple companies, multiple years, and multiple report pages. The answer must depend on retrieval; it must not be answerable using only a single data point or page.

 # Output Control
Please generate each output as a strict JSON object in the following format:

```
{
    "result":[
        {
            "question": "Your question here",
            "answer": "AnswerName or AnswerName1, AnswerName2. . . ",
            "subtype": "Multi-hop Numerical Reasoning, Ranking, Comparison, Calculation.",
            "evidence_pages": [" ", " ", " "]
        }, ...
    ]
}
```

Table 21: Prompt for Generating Numerical Reasoning Questions.

Question Generation Prompt for Table Generation

 # Role
You are a senior financial analysis expert. Your task is to generate table-generation questions that require calculating multiple financial indicators (2–4 metrics) for the year based on financial information presented in annual report pages.

 # Task Description
Table generation in financial report analysis involves computing multiple financial ratios or performance indicators from structured financial information appearing in annual report pages, such as balance sheets, income statements, and cash flow summaries. Based on the provided report excerpts, generate questions that require calculating several financially meaningful metrics and presenting the results in a concise tabular format.
The metrics may include, but are not limited to, standard indicators such as return on equity, return on assets, gross margin, net profit margin, revenue growth rate, net profit growth rate, debt to assets, debt to equity, equity to assets, current ratio, quick ratio, inventory turnover, receivables turnover, or any other valid financial ratio derivable from typical annual report disclosures.
Each generated question should require synthesizing multiple numerical values from the report pages to compute the requested metrics.

 # Question Construction Requirements
Each generated question must satisfy the following conditions:

- Each question must require calculating **2 to 4 financial metrics** for the year.
- Each question must strictly follow the natural-language template below:

  "Calculate the {company_name}'s [financial_metric1, financial_metric2, . . .] in {year} given the attached annual report data. Output the results in a markdown-formatted table. Use 'Item' and {year} as the column headers. Express the result as a decimal, rounded to four decimal places."

- Metrics may be selected from the provided list or any other valid financial ratios computable from standard annual report data. The provided list includes: return on equity, return on assets, gross margin, net profit margin, revenue growth rate, net profit growth rate, debt to assets, debt to equity, equity to assets, current ratio, quick ratio, inventory turnover, receivables turnover, . . .
- Each question must be self-contained and fully understandable without external context.
- Each question must require computation using multiple numerical values extracted from the provided annual report pages.

 # Output Format
Return the final result as a strict JSON object:

```
{
    "result": [
        {
            "question": "Your generated question1 here",
            "metrics": ["metric1", "metric2", "..."],
            "evidence_pages": [" ", " ", " "]
        }, ...
    ]
}
```

Table 22: Prompt for Generating Table Generation Questions.

Question Generation Prompt for Chart Generation

 # Role play
Suppose you are a senior financial visualization expert and your task is to generate high-quality and diverse chart generation questions based on financial report excerpts (tabular or textual data) that require transforming structured tables into meaningful visualizations.

 # Task Descriptions
Chart generation in financial report analysis involves selecting the most appropriate visualization to clearly present relationships, trends, comparisons, and distributions of financial metrics. These tasks typically require identifying the relevant metrics from multiple company reports or fiscal years, determining the analysis objective (such as revenue growth, cost breakdown, or market share comparison), and specifying the correct chart type (e.g., line chart, bar chart, pie chart, scatter plot). The purpose is to simplify complex tabular data, enhance interpretability, and provide clear insights for decision-making. Each generated question must explicitly specify what type of chart should be drawn, the metric(s) to be visualized, and the entities or years to be compared.
The input consists of one or more tables or text blocks extracted from annual reports, potentially spanning multiple companies and multiple fiscal years. Based on the type of input, design visualization questions that require meaningful financial analysis — use cross-company comparisons when the input involves multiple companies, and cross-year analyses when it spans multiple fiscal years of the same company.

 # Generation Restrictions
To generate five questions based on the given task description and multi-source financial tabular data, give due consideration to the following aspects:

- **Complexity:** Include a range of complexities. Simple tasks may require plotting a single metric as a line or bar chart. More complex tasks should require combining multiple metrics or entities, such as plotting operating expenses and net income over time, or visualizing market share composition.
- **Length:** Each question should be between 20 and 50 words, phrased concisely but fully informative.
- **Diversity:** The subtypes covered by the questions should include LineChart Generation, BarChart Generation, ScatterChart Generation, and PieChart Generation.
- **Real-World Relevance:** Questions must reflect actual financial analysis needs, such as revenue trends across years, cost structure composition, profitability comparisons, or cross-company performance benchmarking.
- **Writing-Style:** Use professional, clear, and consistent financial terminology. Avoid vague expressions. Specify chart type, metric(s), entity names, and timeframes explicitly in each question.
- **Question Form:** Each question must be phrased as a single, grammatically complete sentence. Do not split into multiple sub-questions.
- **Context-Independent Question:** Each question must be self-contained, explicitly mentioning all necessary details (metric, company, fiscal years, chart type) so it can be understood without external context.
- **Retrieval Dependency:** Ensure that each question requires referencing multiple companies, multiple years, or multiple metrics; no single-cell or trivial table lookups are allowed.

 # Output Control
Please generate each output as a strict JSON object in the following format:

```
{
    "result":[
        {
            "question": "Your chart generation question here.",
            "subtype": "LineChart Generation, BarChart Generation, ScatterChart Generation,
             PieChart Generation",
            "evidence_pages": [" ", " ", " "]
        }, ...
    ]
}
```

Table 23: Prompt for Generating Chart Generation Questions.

Question Generation Prompt for Knowledge-Intensive Reasoning

 # Role Play
Suppose you are a senior annual report analysis expert and your task is to generate high-quality textual reasoning question–answer pairs that can only be answered through retrieval and synthesis across multiple pages or documents of company financial reports.

 # Task Description
Textual reasoning tasks in financial report analysis involve interpreting and synthesizing qualitative information found in unstructured report excerpts, such as management discussion sections, strategic outlooks, ESG disclosures, and risk factor narratives across different companies and fiscal years. These tasks typically require retrieving relevant passages from multiple report sections, identifying themes, comparing strategic directions, or evaluating qualitative shifts across time or entities. Each question must require multi-source retrieval to answer, simulating a real-world retrieval-augmented analysis workflow. The input consists of one or more images from official annual reports. These images may represent different pages from the same or different companies, and may span multiple fiscal years. Based on the type of input images, design questions that match the appropriate comparison mode — use cross-company horizontal comparisons when images are from different companies, and cross-year vertical analyses when images span multiple fiscal years of the same company.

 # Generation Rules
To generate five question–answer pairs based on the given task description and multi-source annual report data, give due consideration to the following aspects:

- **Complexity**: Questions should require combining and synthesizing information from multiple documents, companies, or fiscal years. Avoid questions that can be answered from a single paragraph or a single report page.
- **Length**: Questions should be between 20 and 50 words. Keep the language professional, precise, and analytically focused.
- **Real-World Relevance**: Questions should resemble inquiries that financial analysts, investors, or regulators would realistically make, such as tracking strategic evolution, comparing disclosures, or evaluating responses to macroeconomic or industry-specific risks.
- **Answer Form**: Answers should be natural language explanations or summaries, potentially consisting of several sentences, and must provide clear reasoning rather than short numeric or entity-only responses.
- **Context-Independent**: Each question must explicitly mention the relevant company names, fiscal years, and thematic focus, so that it is fully understandable on its own without external context.
- **Retrieval Dependency**: Ensure that each question–answer pair requires multi-document retrieval and synthesis, rather than relying on a single data point or isolated passage.

 # Output Format
Please generate five question–answer pairs and output them as a strict JSON object in the following format:

```
{
    "result":[
        {
            "question": "Your question here",
            "answer": "A full natural language explanation here,
             possibly several sentences, clearly reasoned and complete.",
            "evidence_pages": [" ", " ", " "]
        }, ...
    ]
}
```

Table 24: Prompt for Generating Knowledge-Intensive Reasoning Questions.

Answer Annotation Prompt for Explainable Fact Verification

 # Role Play
Suppose you are a senior financial annual report analysis expert and your task is to answer fact verification questions based on evidence from annual reports.

 # Task Description
You are given a statement and one or more evidence excerpts from financial annual reports. Your task is to decide whether the statement is:

- **Supported**: fully confirmed by the provided evidence,
- **Refuted**: contradicted by the provided evidence,
- **Insufficient**: the provided evidence is not enough to confirm or refute the statement.

You must also provide a short explanation that clearly summarizes the reasoning behind your decision. The explanation should cite the relevant metrics, years, companies, or textual information from the evidence.

 # Reasoning Process

1. Read the statement carefully and identify the key metric(s), company name(s), year(s), and claim being made.
2. Examine the provided evidence excerpts (tables or text) to see if the required information is present.
3. Compare the statement against the evidence:
   - If the evidence confirms all aspects, label the statement as **Supported**.
   - If the evidence contradicts the statement, label it as **Refuted**.
   - If the evidence is incomplete or does not cover all aspects, label it as **Insufficient**.
4. Summarize the reasoning in a concise explanation.

 # Output Control
Please generate the output in strict JSON format:

```
{
  "answer": "Supported or Refuted or Insufficient",
  "explanation": "Your concise reasoning here, referring to the evidence"
}
```

 # Question
The question is:

Table 25: Prompt for Answer Annotation in Explainable Fact Verification

Answer Annotation Prompt for Numerical Reasoning

 # Role Play
Suppose you are an expert in financial annual report analysis and your task is to provide precise answers to questions based on the content of annual report tables.

 # Chain-of-Thought
Let's think step by step as follows and make the most of your strengths as a financial report analysis expert:

1. Fully understand the question and extract the necessary information from it, including the metric(s), company names, fiscal years, and calculation/comparison requirements.
2. Clearly and comprehensively understand the content of the annual report tables, including the structure, meaning of each row and column header, and any summative or flag rows (such as Total, Consolidated, Average).
3. Based on the question, select the relevant rows and columns and locate the required values across the appropriate companies or years.
4. According to the requirements of the question, perform the needed operations such as calculation, ranking, comparison, or aggregation.
5. Output the reasoning steps and then the final answer in the specified format.

 # Output Control

- First, you need to output your reasoning steps according to the question and table itself. The reasoning steps should follow the format below: `[Reasoning steps for this question are as following: 1.First, we need to... 2.We need to...]` Output steps until final answers get solved.
- Then, you need to output the final answer. The final answer should follow the format below: `Final Answer: AnswerName1, AnswerName2...` Ensure the final answer format is the last output line and can only be in this form, with no other content.
- Ensure the `AnswerName` is a number or entity name, as short as possible, without any explanation. Give the final answer directly without any explanation. Note: If the final answer has multiple decimals, retain two decimal places.

I will give you multiple annual report page images and a question; please use them to answer following the above reasoning and answer format strictly.

 # Question
The question is:

Table 26: Prompt for Answer Annotation in Numerical Reasoning

Answer Annotation Prompt for Table Generation

 # Role Play
Suppose you are a senior financial annual report analysis expert and your task is to answer quantitative analysis questions by producing clean, markdown-formatted tables.

 # Task Description
You are given a question along with one or more evidence tables or textual excerpts from annual reports. Your task is to calculate the required financial indicators and output the results strictly as a markdown table. The markdown table must be formatted with clear column headers and aligned rows, so that the output can be directly rendered without additional editing.

 # Reasoning Process

1. Read the question carefully and identify the target metrics, entities, and fiscal year(s).
2. Extract the necessary numerical values from the provided evidence, making sure to interpret units, scales, and negative formatting correctly.
3. Perform any required calculations (ratios, margins, growth rates, turnovers, etc.), paying special attention to unit conversions (e.g., thousands vs. millions), and round results to the number of decimal places specified in the question.
4. Construct a markdown table with appropriate headers (e.g., Item, Year, Company) and rows corresponding to each metric.

 # Output Control

- The final output must be a markdown table only, no extra text or explanation.
- Use vertical bars (`|`) and dashes (`-`) to format the table so that it renders correctly in markdown.
- Ensure the column headers exactly match what is requested in the question (e.g., `"Item, 2023"`).
- Round all numeric results as instructed in the question (default: four decimal places if not otherwise specified).
- The final output values must be **pure numbers without any units** (e.g., output `120.4500`, not `$120.4500 millions`).
- Do not include any text before or after the markdown table.

 # Question
The question is:

Table 27: Prompt for Answer Annotation in Table Generation

Answer Annotation Prompt for Chart Generation

 # Role Play
Suppose you are an expert in financial annual report analysis, and your task is to generate executable Python code that answers the question using the content of financial annual report.

 # Chain-of-thought
Let's think step by step as follows and make the most of your strengths as a financial report table analysis expert:

1. Understand the question and extract the required information: metrics, company names, fiscal years, and calculation/-comparison requirements.
2. Carefully read the annual report tables, including the structure, row and column headers, units, and any summative rows or special flags.
3. Select the most relevant rows and columns according to the question and locate the corresponding cells across years or companies.
4. Perform the required operations (calculation, ranking, comparison, or aggregation) and prepare the data for plotting with pandas and matplotlib.

 # Output Control

- The final answer should follow the format below and ensure the first three code lines is exactly the same with the following code block:
- You only need to output the final code without any interpretation, make sure that your code can be run directly without any syntax errors.
- Please do not read external files, and the pandas and matplotlib libraries have been successfully introduced.
- Ensure that the X-axis used for drawing in the code is arranged in ascending alphabetical or numerical order.
- Ensure the last line in python code can only be `"plt.show()"`, no other from.
- Give the final answer to the question directly without any explanation.

I will give you multiple annual report page images and a question; please use them to answer the question in the specified format without any explanation: [Final Answer]: import pandas as pd import matplotlib.pyplot as plt ... plt.show()

 # Question
The question is:

Table 28: Prompt for Answer Annotation in Chart Generation

Answer Annotation Prompt for Knowledge Reasoning

 # Role
You are a professional financial analysis assistant. Your task is to answer textual reasoning questions strictly based on retrieved evidence images from company annual reports.

 # Input
The images are retrieved evidence pages from annual reports. They may contain management discussions, strategic outlooks, ESG disclosures, risk factor narratives, or other narrative sections. Treat these images as the only valid sources of information for answering the question.

 # Task

1. Carefully read the question and identify the key entities, years, and themes.
2. Extract and interpret the relevant passages directly from the retrieved evidence images.
3. Compare, synthesize, or trace changes across years or companies if multiple images are provided.
4. Provide a reasoned, well-structured answer in natural language.

 # Answer Requirements

- Base both answers only on the content visible in the retrieved evidence images; do not use outside knowledge or assumptions.
- **Long Answer**: Provide one combined long-form answer that naturally weaves together the reasoning process and the final conclusion.
- **Short Answer**: Provide a brief, clear, and direct summary consisting of 2–4 sentences.
- If the retrieved evidence is insufficient, explicitly write `Insufficient evidence` for both the long answer and the short answer.

 # Answer Format
Long Answer:
Detailed reasoning and conclusion in one long-form response.

Short Answer:
Concise summary of the conclusion in a few sentences.

 # Question
The question is:

Table 29: Prompt for Answer Annotation in Knowledge Reasoning