FINLFQA: Evaluating Attributed Text Generation of LLMs in Financial Long-Form Question Answering

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have demonstrated remarkable performance across various tasks. However, in long-form question answering (LFQA), they often struggle with factual accuracy, frequently generating hallucinated responses. In this work, we introduce FINLFQA, a benchmark designed to evaluate LLMs' ability to generate answers with reliable attributions. FINLFQA evaluates three key aspects: (1) evidence-supported content to enhance factual grounding and verifiability, (2) step-by-step calculations using executable code for numerical reliability, and (3) domain-specific reasoning informed by knowledge. We conduct an extensive evaluation of eight LLMs, leveraging the developed automated evaluation protocol to evaluate their performance. Our findings show that GPT-4o outperforms other models, while open-sourced models are closing the gap with proprietary models, which demonstrates that open-source models are becoming competitive alternatives for real-world applications. We also find that post-hoc and end-to-end generation perform similarly, while iterative self-feedback provides no significant improvement except external signal is provided.

1 Introduction

Long-form question answering (LFQA) presents a significant challenge in natural language processing, as it requires models to not only comprehend extensive context but also maintain factual accuracy throughout the response generation process (Xu et al., 2023). One of the prevalent issues in LFQA is hallucination (Ji et al., 2023), where models generate content that is factually incorrect, thereby undermining user trust. This challenge is particularly pronounced in finance domain for two primary reasons. First, financial content contains numerous numerical values that require models to perform accurate numerical reasoning (Zhao et al., 2024c). Second, the domain includes extensive specialized knowledge, requiring models to understand financial concepts and their relationships before executing proper reasoning tasks (Gan et al., 2024).

To address this challenge, researchers have demonstrated increasing interest in attributed text generation (Ye et al., 2024; Gao et al., 2023b). This approach enhances content trustworthiness by producing responses accompanied by supporting evidence, thereby enabling verification of generated claims. However, current attribution methods face several limitations. First, existing studies predominantly focus on evidence attribution for extractive tasks (Huang et al., 2024b), which proves insufficient for the complex requirements of financial applications. Second, contemporary LFQA in finance often generate calculations without revealing intermediate steps, presenting only final results (Zhao et al., 2024a). In such cases, mere citation attribution may inadequate in preventing hallucinations, as models might perform incorrect calculations despite citing accurate sources. Finally, even with proper citations, LLMs may generate factually inaccurate content due to insufficient domain-specific knowledge (Martino et al., 2023), particularly in specialized financial contexts.

In this work, we introduce FINLFQA[1], a comprehensive benchmark for evaluating long-form QA and attributed generation by LLMs. FINLFQA provides natural language questions alongside retrieved paragraphs from financial reports and definitions of specialized terms. The task requires generating responses with three types of attribution: source paragraphs, intermediate reasoning process, and domain-specific knowledge, which is illustrated at Figure 1.

We design automatic evaluation methods to assess both the answers and their attributions. For answer evaluation, we measure both token-level

[1]The data and code for this study can be found at https://anonymous.4open.science/r/FinLFQA, and will be released publicly upon publication.

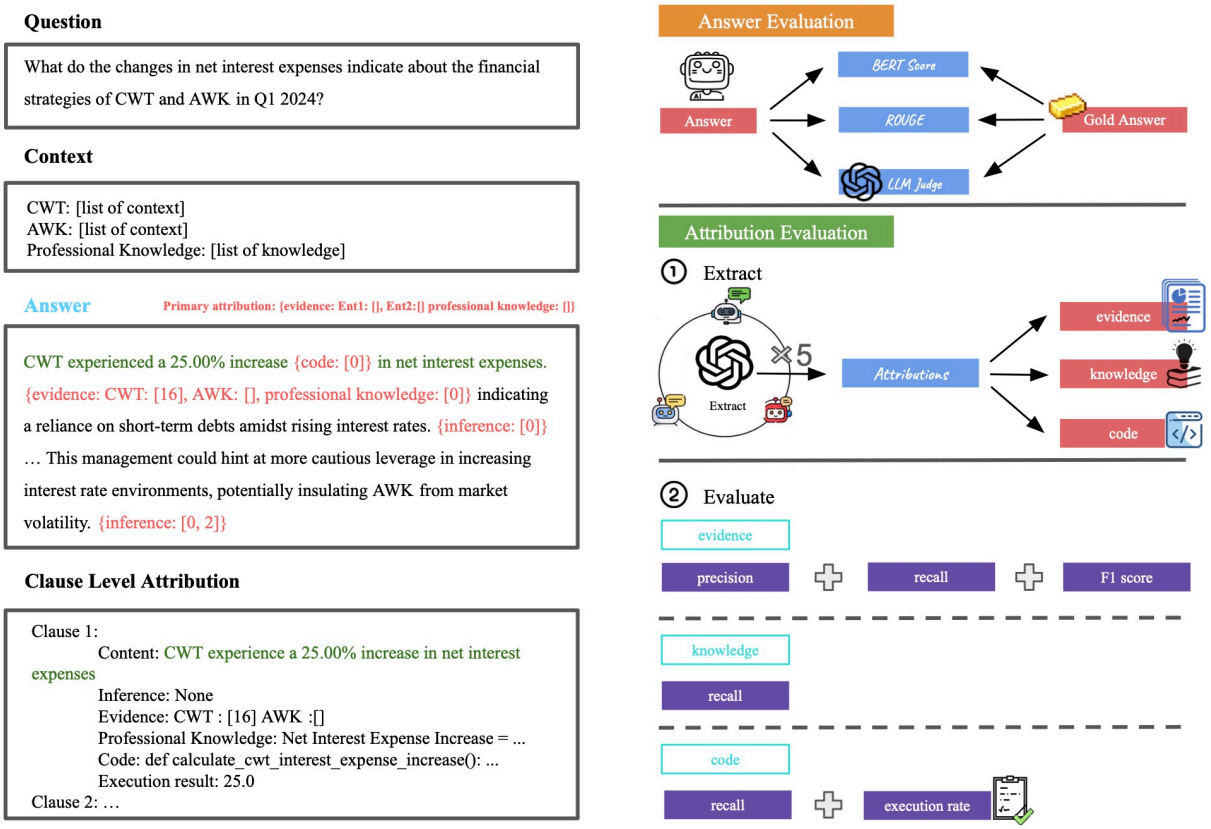

Figure 1: **(Left)** Overview of FINLFQA. **(Right)** Overview of automatic evaluations.

and semantic-level similarity using ROUGE and BERTScore. Additionally, we employ an LLM-as-a-judge approach, where the generated answer is compared against the ground truth. To assess numerical accuracy, we compute the precision, recall, and F1 score of numerical values present in the ground truth. For attribution evaluation, we measure the precision, recall, and F1 score for supporting evidence. Code attributions are assessed using the execution success rate. Finally, we evaluate professional knowledge attribution by computing recall against relevant domain-specific references.

We conduct experiments on different generation approaches, including post-hoc generation and end-to-end generation. In end-to-end generation, we evaluate two settings: (1) single-pass generation, where the model produces the answer and attributions in one step, and (2) iterative generation, where the model refines its response based on feedback from previous iteration. Additionally, we compare our results against two baseline methods: VANILLA and CoF. We evaluate FINLFQA on 8 different LLMs, including both proprietary and open-source models, providing a comprehensive evaluation of their performance across various generation strategies. Our contributions are summarized as follows:

- We propose a comprehensive benchmark, FINLFQA, specifically designed for evaluating long-form question answering and attribution generation in the financial domain. Our benchmark uniquely combines financial report paragraphs with domain-specific term definitions to support both factual and conceptual verification.

- We develop a reliable automated evaluation system that provide more fine-grained assessments, which goes beyond traditional metrics like ROUGE and BERTScore to better capture the nuances of financial reasoning and precision.

- Our extensive experiments demonstrate that end-to-end generation performs on par with post-hoc generation, indicating that current models are well-equipped to handle complex attribution tasks. Furthermore, iterative generation does not yield significant improvements over single-pass generation, except in code execution, where execution-based feedback plays a crucial role.

2 Related Work

2.1 Long-form Question Answering

Hallucination is a significant issue for generative LLMs (Xiao and Wang, 2021; Shuster et al., 2021). Many studies have explored augmenting LMs with externally retrieved information to mitigate this problem (Borgeaud et al., 2022; Izacard et al., 2023). This issue has also attracted growing interest in attributed LLMs (Hu et al., 2024; Worledge et al., 2024), which aim to enhance the verifiability of information by generating responses that attribute content to reliable sources. However, most existing work has focused on tasks such as question answering or text summarization in information-seeking contexts (Xu et al., 2024; Bohnet et al., 2023). These approaches often overlook more complex question-answering scenarios that involve numerical reasoning and knowledge-intensive tasks in real-world settings. Therefore, in our work, we will incorporate knowledge into the prompts, requiring models to generate programs that show calculation steps and demonstrate knowledge usage, accompanied by appropriate citations.

2.2 Attributed Text Generation

Attribution has gained significant attention for enhancing the interpretability, verifiability, and safety of LLMs (Li et al., 2023; Cohen-Wang et al., 2024). Two main approaches have emerged in this field. The first approach utilizes prompt-based or in-context learning, where LLMs are instructed to generate responses with citations given the question and retrieved paragraphs (Kamalloo et al., 2023; Gao et al., 2023b). The second approach employs post-hoc methods, which attribute existing responses using an auxiliary language model to identify relevant sources (Huang et al., 2024c; Gao et al., 2023a). Ye et al. (2024) demonstrated that fine-tuning an LLM to generate citations and iteratively refine responses yields more accurate results than both prompting-based and post-hoc methods. However, existing work on attribution for language models has primarily focused on citation generation. In the financial domain, citation attribution alone is insufficient to mitigate hallucination and generate robust results, as financial tasks often require numerical reasoning and domain-specific knowledge. Therefore, FINLFQA includes not only citation attribution but also numerical reasoning processes and financial domain expertise.

3 FINLFQA Benchmark

In this section, we first formulate the task, then describe our data annotation process and present the statistical analysis of the dataset. Table 7 in the Appendix presents the profiles of the eleven annotators involved.

3.1 Task Formulation

We define the task of FINLFQA as follows: Given portions of financial documents from two companies comprising multiple text paragraphs in set $\mathcal{D}$, where each document contains both textual and tabular data, and a query q, the model should generate a response $\mathcal{R}$ based on the provided context. The response consists of n statements $s_1, s_2, \ldots, s_n$, where together these statements form the answer to the query. Each statement s_i consists of: (a) A clause t_i that contributes to answering the query (b) Three types of attributions: (1) A list of paragraph indices $\mathcal{C}_i = i_1, i_2, \ldots, i_k$ where $i_j \in 1, 2, \ldots, |\mathcal{D}|$, indicating the relevant source paragraphs supporting the statement. (2) An intermediate reasoning process $\mathcal{P}_i$ expressed as a Python program, if the statement involves numerical reasoning. (3) A list of professional knowledge indices $\mathcal{K}_i = k_1, k_2, \ldots, k_m$ where $k_j \in 1, 2, \ldots, |\mathcal{K}|$, referencing entries in the knowledge base $\mathcal{K}$. Thus, each statement can be represented as $s_i = (t_i, \mathcal{C}_i, \mathcal{P}_i, \mathcal{K}_i)$, and the optimal response can be formulated as:

$$\mathcal{R}^* = \underset{\mathcal{R}}{\operatorname{argmax}}\, P(\mathcal{R}|q, \mathcal{D}, \mathcal{K}) \quad (1)$$

where $P(\mathcal{R}|q, \mathcal{D}, \mathcal{K})$ represents the probability of generating response $\mathcal{R}$ given the query q, document set $\mathcal{D}$, and knowledge base $\mathcal{K}$.

3.2 Data Annotation

Report Selection Following previous work (Zhao et al., 2024c,b), we use quarterly reports (i.e., Form 10-Q) of companies as our source documents, which are publicly available in the U.S. Securities and Exchange Commission's open-source database[2]. We selected two companies based on their Standard Industrial Classification (SIC) Code[3], randomly choosing companies within the same code to ensure industry comparability. For these two companies, we chose financial reports

[2] https://www.sec.gov/edgar/search/

[3] https://www.sec.gov/search-filings/standard-industrial-classification-sic-code-list

Property (Median/Avg)	**Value**
Question Length	16.0 / 16.3
Answer Length	51.0 / 52.4
# Clauses	3.0 / 3.4
Report Length	2144.0 / 2221.6
# Paragraphs	40.0 / 40.5
# Companies	89
# Evidence	2.0 / 2.9
# Code	1.0 / 1.1
# Professional Knowledge	1.0 / 1.2
Development Set Size	302
Test Set Size	706

Table 1: Data statistics in FINLFQA dataset.

from the same fiscal quarter to maintain temporal consistency.

Data Annotation Given financial reports from two companies, annotators first read the reports to identify sections containing numerical values and common content between the two reports. Based on these sections, annotators design questions that require calculations, with evidence preferably spanning multiple paragraphs and tables. We provide bonus compensation for questions requiring complex mathematical calculations beyond simple addition or ratio computations. After query annotation, annotators must provide answers structured in atomically numbered clauses, where each clause includes either supporting evidence paragraph indices or inferences from previous clauses. For mathematical calculations, annotators must document the computational steps alongside the clauses.

Attribution Annotation Finance-expert annotators verify the initial annotations, ensuring that answers are clear, well-supported, and entirely derivable from the provided text. Upon passing quality checks, annotators proceed with evidence attribution annotation. For the professional knowledge component, we randomly sample one thousand financial concepts from our Wikipedia-based knowledge base with the size of twenty for each question. Annotators then indicate whether any of these concepts inform the reasoning in each clause. For the calculation process annotation, financial experts first verify the mathematical equations, which are then passed to finance-expert annotators who convert the verified calculations into Python functions. These functions are structured to include variable assignments, calculation steps, and return values.

3.3 Data Statistics

Table 1 summarizes the statistics of FINLFQA, which comprises 1008 examples. The dataset is randomly split into a development set (302 examples, 30%) and a test set (706 examples, 70%). To avoid data contamination, test set answers remain private. Instead, an online evaluation platform enables researchers to assess models and join a leaderboard. Evaluation is performed in a zero-shot setting.

4 Methods

In FINLFQA, we require LLMs to not only generate answers to queries but also provide three distinct attributions for each response. To achieve this, we propose two approaches: post-hoc generation (§4.1) and end-to-end generation (§4.2), which is also illustrated at Figure 2. Additionally, we evaluate our method against two established baselines from prior work (§4.3).

4.1 Post-hoc Generation

The post-hoc generation process consists of two stages. In the first stage, given the financial reports and question as input, the model generates an answer. In the second stage, we generate attributions post-hoc: given the reports, question, and the previously generated answer, the model produces three attributions for each clause in the response. The prompts for answer generation and attribution generation are shown in Figure 3 and Figure 4.

4.2 End-to-end Generation

In end-to-end generation, LLMs generate both clauses and their attributions simultaneously. We explore this approach under two distinct settings. The prompts for generation and feedback are shown in Figure 5 and Figure 6.

Single-pass Generation The model generates both the answer and attributions in a single pass, taking financial reports and the question as input.

Iterative Generation Iterative generation introduces a multi-step refinement process to enhance response accuracy and reliability. The LLM first generates an initial response. The same LLM then extracts and executes a Python function, incorporating the computed values into the generated response. Given the financial reports, question, initial response, and execution results, the LLM provides feedback based on four criteria: (1) completeness—whether the response fully addresses

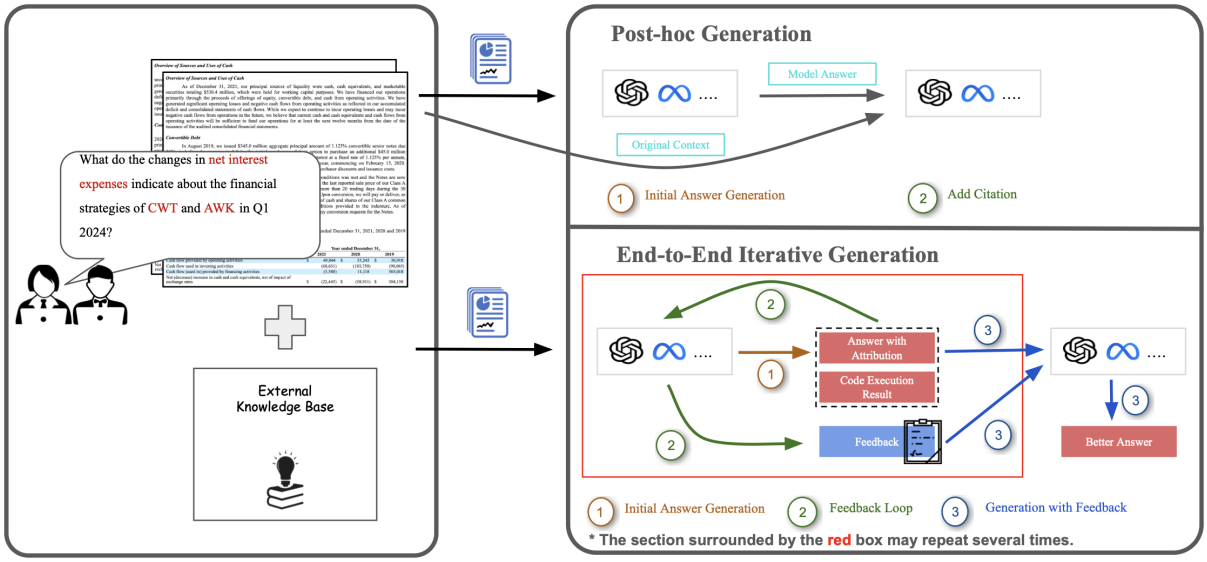

Figure 2: Overciew of post-hoc generation and end-to-end generation.

the question; (2) evidential support—ensuring every claim is backed by the financial reports to mitigate hallucination and enhance robustness; (3) numerical consistency—verifying that the extracted program output aligns with the values stated in the response, and if execution fails, providing debugging guidance; and (4) professional knowledge integration—assessing whether the identified domain knowledge contributes to reasoning in the clause. Based on this evaluation, the model generates structured feedback to guide the next iteration. This iterative refinement continues until either no further improvements are identified or the process reaches a predefined maximum iteration threshold.

4.3 Others

We evaluated our benchmark on two baselines, namely **Vanilla** (Gao et al., 2023b) and **Coarse to Fine** (CoF) (Zhang et al., 2024). The evaluation focuses on answer generation and the generation of one of the three attributions: evidence.

Vanilla This method implements end-to-end generation with citation and in-context learning. In FINLFQA, we skip the BM25 retrieval step because the context of financial reports are not long. Moreover, the original prompt designed for the **ELI5** dataset (Fan et al., 2019)—focused on how/why/what questions with lengthy answers—is well suited for our domain. Thus, we use GPT-4o to generate citations on FINLFQA, and the prompt we use is at Appendix Figure 7.

CoF This method is a post-hoc generation approach that adopts a three-stage generation process: (1) Generating an initial answer; (2) Extracting chunk-level citations by splitting the answer into 128-token chunks; (3) Refining sentence-level citations by further segmenting and improving citation accuracy. In FINLFQA, we modify the CoF to better handle financial reports containing information from multiple sources. Instead of directly segmenting content into 128-token chunks, we first label each chunk with its source (e.g., "Company A," "Company B," or "Professional Knowledge") to maintain contextual clarity. For tabular data, we preserve entire tables as single units to prevent information fragmentation. We retain the retrieve-and-answer process using GPT-4o. The prompts we are applying are shown in Appendix Figure 8, Figure 9, and Figure 10.

5 Evaluations

FINLFQA provides automatic evaluations on both answer and attributions, as it is shown at Figure 1.

5.1 Answer Evaluation

Token-level We use **ROUGE** (Lin, 2004) to evaluate the lexical overlap between generated answers and ground truth responses. Specifically, we adopt ROUGE-L as it captures the longest common subsequence, which is better for long-form QA tasks where exact phrasing may vary while preserving key information.

Semantic-level We use **BERTScore** (Zhang et al., 2020) to measure the contextual similarity between generated answers and ground truth responses. BERTScore computes token embeddings using a pre-trained language model and evaluates similarity based on cosine similarity, which enables a more nuanced assessment of meaning beyond surface-level overlap.

LLM-as-a-judge This approach (Zheng et al., 2023) leverages large language models as automated evaluators, offering a scalable alternative to costly human evaluation while maintaining assessment quality. Unlike ROUGE and BERTScore, which rely solely on surface-level or embedding-based similarity measurements, LLM-based evaluation can assess nuanced aspects of answer quality. We evaluate responses by providing GPT-4o (OpenAI et al., 2024) with the financial reports, question, generated answer, and ground truth. The model assesses three key criteria: (1) Accuracy: whether the answer correctly addresses the question and aligns with the ground truth; (2) Numerical Correctness: whether all numerical calculations and values are precise and accurate; and (3) Supporting Evidence: whether all claims are properly substantiated by information from the financial reports. Each criterion is scored on a scale of 1 to 5, with the final score being the sum of these three components, ranging from 3 to 15.

Numerical Values In financial analysis, precise numerical accuracy is crucial even when the overall semantic meaning is preserved. Small numerical errors can significantly impact financial decisions, making traditional semantic similarity metrics insufficient. Therefore, we specifically evaluate numerical accuracy by extracting all numerical values from both the ground truth and generated answers. We calculate **precision**, **recall**, and **F1 score**. To account for real-world variations in numerical representation, we implement flexible matching criteria by normalizing numbers to account for different decimal precisions (e.g., "3.965" is considered correct if the ground truth is "3.97") and standardizing values across different scale representations (e.g., "3 million" is considered equivalent to "3,000 thousands").

5.2 Attribution Evaluation

Evidence We evaluate the quality of evidence attribution by measuring how well the model identifies and cites relevant supporting evidence from the financial reports. For each generated answer, we compute **precision**, **recall**, and **F1 score**.

Code We evaluate the model's ability to generate executable code for numerical calculations by measuring the **execution success rate**. This metric calculates the percentage of generated code snippets that successfully execute. This evaluation ensures that the model not only provides correct answers but also generates valid, executable code to support its calculations.

Professional Knowledge We assess the model's use of professional financial knowledge through a **recall**-based evaluation. This measures how well the model identify knowledge that contributing reasoning of statements.

6 Experiments

This section describes the experimental setup, models evaluated, main results, and error analysis.

6.1 Models

We evaluate eight LLMs on FINLFQA, including GPT-4o, Qwen2.5-72B (Qwen et al., 2025), Llama-3.3-70B (Grattafiori et al., 2024), Llama-3.2-3B, Llama-3.2-1B, Mistral-Small-24B (Jiang et al., 2023), Mistral-8x22B (Jiang et al., 2024), and Phi-4 (Abdin et al., 2024). The exact versions and specifications of these models are provided in Table 3 in the Appendix.

6.2 Experimental Setup

We conducted experiments on open-source LLMs using the vLLM framework (Kwon et al., 2023). All experiments were performed under a zero-shot setting, with a temperature of 1.0 and a maximum output length of 2048 tokens. Following previous work (Zhao et al., 2024c,b), we serialize tabular data by using a vertical bar (|) to separate columns and a newline to separate rows.

6.3 Main Results

Tables 2 and 4 present the performance of LLMs on the development and test sets, respectively. Additionally, Tables 5 and 6 provide a detailed breakdown of the LLM-as-a-judge evaluation, including accuracy, numerical correctness, and the quality of supporting evidence. We summarize the main takeaways from the experiments below.

Model	Attribution Performance					Answer Performance					
	Evidence			% Exec.	Knowl.	R-L	BS	LLM-	Numerical		
	Prec.	Recall	F1	Success	Recall			as-a-judge	Prec.	Recall	F1
Post-hoc											
GPT-4o	49.0	**78.1**	55.5	9.3	19.3	23.1	87.5	13.6	**37.9**	**58.0**	**42.3**
Qwen2.5-72B	39.5	71.1	45.7	10.3	19.1	22.6	87.9	13.0	33.7	52.8	37.8
Llama-3.3-70B	44.9	71.3	50.4	16.6	16.1	19.7	87.5	13.1	35.9	51.8	39.1
Llama-3.2-1B	1.7	3.0	1.9	1.5	0.5	15.3	84.6	5.6	12.3	14.5	11.7
Llama-3.2-3B	10.5	19.2	12.2	14.3	8.2	19.1	86.6	8.9	24.2	32.6	25.3
Mistral-Small-24B	49.1	69.9	53.0	4.5	22.2	25.0	88.2	12.6	36.3	48.7	38.5
Mistral-8x22B	36.2	58.5	40.9	11.1	16.7	21.7	87.3	12.1	33.7	42.6	34.1
phi-4	42.1	66.8	47.3	8.1	18.3	20.1	87.0	13.0	32.5	48.8	35.6
End-to-end (single-pass)											
GPT-4o	46.9	75.0	53.3	26.3	17.3	19.3	87.0	**13.7**	35.2	54.9	39.4
Qwen2.5-72B	48.7	68.6	50.4	15.4	17.1	21.7	86.9	12.1	33.4	45.3	36.1
Llama-3.3-70B	**53.7**	68.3	56.1	20.1	17.1	21.1	87.0	12.0	35.8	45.0	36.5
Llama-3.2-1B	0.6	0.7	0.6	8.5	0.0	13.9	81.1	4.5	6.8	8.5	6.6
Llama-3.2-3B	13.3	22.8	15.1	22.5	6.2	18.5	85.8	7.5	19.3	27.0	19.9
Mistral-Small-24B	52.9	69.3	**55.8**	12.9	**22.4**	23.3	87.4	12.4	33.8	47.5	36.5
Mistral-8x22B	40.1	57.8	41.2	18.4	16.5	21.4	86.6	12.0	33.6	43.7	37.1
phi-4	40.7	59.2	44.5	14.1	16.7	21.2	86.6	12.5	37.4	45.9	37.3
End-to-end (Iterative)											
GPT-4o	46.7	75.0	53.1	**29.8**	17.1	20.0	86.8	13.5	34.2	54.7	38.9
Qwen2.5-72B	46.6	68.0	52.0	22.7	17.2	22.2	86.6	12.0	35.8	46.0	36.5
Llama-3.3-70B	50.8	67.9	54.1	27.3	17.3	22.0	86.7	12.1	35.4	46.8	36.6
Llama-3.2-1B	0.5	0.7	0.5	8.9	0.0	13.0	80.3	3.8	2.8	3.0	2.3
Llama-3.2-3B	9.1	13.3	10.0	22.5	5.0	20.6	86.8	6.7	19.1	22.9	18.5
Mistral-Small-24B	53.0	68.5	55.7	18.1	18.2	25.0	88.0	12.2	33.8	46.7	36.5
Mistral-8x22B	43.2	57.3	46.1	18.7	16.3	25.4	88.0	12.1	33.9	47.1	37.1
phi-4	43.8	60.9	47.2	16.6	16.0	25.4	87.8	12.5	37.0	46.1	37.4
Others											
Vanilla	43.9	63.3	48.9	–	–	25.0	88.2	13.3	1.9	2.6	1.9
CoF	40.1	49.0	41.7	–	–	**29.9**	**89.0**	12.0	0.5	0.8	0.6

Table 2: Results of *development* set. R-L denotes ROUGE-L, BS denotes BERTScore. The LLM-as-a-judge evaluation uses a 15-point scale, consisting of three main criteria: accuracy, numerical correctness, and supporting evidence. Each criterion is scored from 1 to 5 points. The detailed breakdown of results for each criterion is shown in the Appendix Table 5.

Fine-grained evaluation metrics provide better comparison All models receive low ROUGE scores because it relies on exact match for evaluation, which is less effective for open-ended tasks where multiple valid responses exist. BERTScore, despite capturing semantic similarity, also fails to differentiate model performance meaningfully. In financial applications, numerical values and factual accuracy are crucial, and even minor differences can alter the factual correctness of a response. Since all models achieve similar BERTScores (approximately 88), this highlights its limitations in detecting factual inconsistencies. Therefore, fine-grained evaluation metrics are essential for financial long-form question answering, and we provide results across multiple dimensions.

GPT-4o performs best Our results show that GPT-4o achieves the highest score in LLM-as-a-judge evaluation (13.7). It also excels in numerical accuracy, a critical aspect in financial applications, producing responses with more correct numerical values compared to other models. Its precision, recall, and F1 scores in numerical matching are 37.9, 58.0, and 42.3, respectively. In code generation, GPT-4o also leads, with a code execution success rate of 29.8% in the end-to-end iterative setting. However, open-source models demonstrate strong potential. Notably, Mistral-small-24B performs best in evidence attribution and professional knowledge recall. In several other metrics, the best open-source models are approaching proprietary models, making them increasingly viable alternatives given their accessibility, lower cost, and faster inference speed.

Post-hoc and end-to-end generation show no significant difference Our results indicate that post-hoc and end-to-end generation perform similarly, suggesting that current frontier models can handle complex tasks effectively. In the post-hoc approach, models first generate an answer and then attribute supporting evidence, whereas in the end-to-end approach, both are generated simultaneously. The latter does not degrade performance and offers two advantages: (1) generating everything at once reduces computational cost and latency, and (2) it improves code generation consistency. In post-hoc generation, numerical reasoning steps are inferred from a pre-generated answer, leading to inconsistencies and lower execution success rates. In contrast, end-to-end generation ensures that answers and reasoning steps are aligned, making outputs more robust and verifiable.

Iterative generation via self-feedback does not improve performance We find no significant difference between single-pass and iterative generation. Since this task primarily relies on reasoning abilities, our results align with prior work (Huang et al., 2024a) showing that self-feedback without external signals does not enhance performance. The only observed improvement is in code execution success rate, which can be attributed to feedback from execution results. When errors occur, they provide corrective signals to the model, explaining why execution success improves in the iterative setting.

6.4 Error Analysis

We randomly select 50 generated outputs from our development set for each of the following settings evaluated on GPT-4o: post-hoc, end-to-end single-pass, and end-to-end iterative generation, totally 150 samples. We identify the following five main types of errors.

Evidence attribution errors [25%] This category includes cases where the model provides redundant or invalid evidence or omits essential supporting details. Challenges in accurately identifying and extracting relevant evidence from paragraphs lead to low precision and recall, hindering the reliability of responses.

Execution errors [22%] These errors occur when the model generates non-executable code due to syntax or logical issues. These errors occur because the model generates code that contains syntax or logical mistakes, making it non-executable.

Numerical information extraction and calculation errors [20%] These errors involve the incorrect extraction of key financial terms, the use of mismatched units (e.g., when units are provided in table headers), and calculation mistakes such as rounding errors.

Knowledge validation errors [15%] Errors arise when the model leverages external knowledge without proper citation. This leads to challenges in maintaining accuracy and consistency in validating facts.

Fluency, factual consistency and reasoning Errors [12%] Errors in this category include generating incorrect timestamps or dates, confusing monetary units, mixing data from different companies, offering faulty reasoning that lacks logical support, and outright hallucinations.

Others [6%] Other errors include overly lengthy or empty answers and instances where responses are produced in languages other than English.

7 Conclusion

This paper introduces FINLFQA, a comprehensive benchmark designed to evaluate the ability of LLMs to generate factually grounded and well-attributed answers in long-form question answering. We design fine-grained automatic evaluation methods that measure answer quality at both token and semantic levels, assess numerical correctness, and evaluate attribution quality across evidence, code, and professional knowledge. Our extensive experiments compare different generation strategies, including post-hoc and end-to-end generation, with single-pass and iterative refinement settings. Experiment results show that GPT-4o achieves the highest overall performance, while open-source models, demonstrate strong potential. Additionally, FINLFQA underscores the importance of fine-grained metrics in financial long-form question answering task. FINLFQA indicate that post-hoc and end-to-end generation perform similarly, suggesting that frontier LLMs can handle complex attribution tasks effectively. Furthermore, iterative self-feedback does not significantly improve overall performance, except in execution-based tasks where explicit feedback from generated outputs aids refinement.

Limitations

FINLFQA evaluates only one proprietary model, limiting insights into broader closed-source LLM performance. The benchmark also focuses on two companies, incorporating more companies will improve generalizability. Lastly, the lack of human evaluation means our assessment relies solely on automatic metrics, which may not fully capture financial reasoning accuracy.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Bernd Bohnet, Vinh Q. Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, Kai Hui, Tom Kwiatkowski, Ji Ma, Jianmo Ni, Lierni Sestorain Saralegui, Tal Schuster, William W. Cohen, Michael Collins, Dipanjan Das, Donald Metzler, Slav Petrov, and Kellie Webster. 2023. Attributed question answering: Evaluation and modeling for attributed large language models. *Preprint*, arXiv:2212.08037.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Madry. 2024. Contextcite: Attributing model generation to context. *Preprint*, arXiv:2409.00729.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Ziliang Gan, Yu Lu, Dong Zhang, Haohan Li, Che Liu, Jian Liu, Ji Liu, Haipang Wu, Chaoyou Fu, Zenglin Xu, Rongjunchen Zhang, and Yong Dai. 2024. Mme-finance: A multimodal finance benchmark for expert-level understanding and reasoning. *Preprint*, arXiv:2411.03314.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023a. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada. Association for Computational Linguistics.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023b. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar

Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia,

Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Nan Hu, Jiaoyan Chen, Yike Wu, Guilin Qi, Sheng Bi, Tongtong Wu, and Jeff Z. Pan. 2024. Benchmarking large language models in complex question answering attribution using knowledge graphs. *Preprint*, arXiv:2401.14640.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024a. Large language models cannot self-correct reasoning yet. *Preprint*, arXiv:2310.01798.

Lei Huang, Xiaocheng Feng, Weitao Ma, Yuxuan Gu, Weihong Zhong, Xiachong Feng, Weijiang Yu, Weihua Peng, Duyu Tang, Dandan Tu, and Bing Qin. 2024b. Learning fine-grained grounded citations for attributed large language models. *Preprint*, arXiv:2408.04568.

Lei Huang, Xiaocheng Feng, Weitao Ma, Liang Zhao, Yuchun Fan, Weihong Zhong, Dongliang Xu, Qing Yang, Hongtao Liu, and Bing Qin. 2024c. Advancing large language model attribution through self-improving. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3822–3836, Miami, Florida, USA. Association for Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Ehsan Kamalloo, Aref Jafari, Xinyu Zhang, Nandan Thakur, and Jimmy Lin. 2023. Hagrid: A human-llm collaborative dataset for generative information-seeking with attribution. *Preprint*, arXiv:2307.16883.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Dongfang Li, Zetian Sun, Xinshuo Hu, Zhenyu Liu, Ziyang Chen, Baotian Hu, Aiguo Wu, and Min Zhang. 2023. A survey of large language models attribution. *Preprint*, arXiv:2311.03731.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ariana Martino, Michael Iannelli, and Coleen Truong. 2023. Knowledge injection to counter large language model (llm) hallucination. In *European Semantic Web Conference*, pages 182–185. Springer.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy,

David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Theodora Worledge, Judy Hanwen Shen, Nicole Meister, Caleb Winston, and Carlos Guestrin. 2024. Unifying corroborative and contributive attributions in large language models. In *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 665–683.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages

2734–2744, Online. Association for Computational Linguistics.

Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. 2023. A critical evaluation of evaluations for long-form question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3225–3245, Toronto, Canada. Association for Computational Linguistics.

Yilong Xu, Jinhua Gao, Xiaoming Yu, Baolong Bi, Huawei Shen, and Xueqi Cheng. 2024. Aliice: Evaluating positional fine-grained citation generation. *Preprint*, arXiv:2406.13375.

Xi Ye, Ruoxi Sun, Sercan Arik, and Tomas Pfister. 2024. Effective large language model adaptation for improved grounding and citation generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6237–6251, Mexico City, Mexico. Association for Computational Linguistics.

Jiajie Zhang, Yushi Bai, Xin Lv, Wanjun Gu, Danqing Liu, Minhao Zou, Shulin Cao, Lei Hou, Yuxiao Dong, Ling Feng, and Juanzi Li. 2024. Longcite: Enabling llms to generate fine-grained citations in long-context qa. *Preprint*, arXiv:2409.02897.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Yilun Zhao, Lyuhao Chen, Arman Cohan, and Chen Zhao. 2024a. TaPERA: Enhancing faithfulness and interpretability in long-form table QA by content planning and execution-based reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12824–12840, Bangkok, Thailand. 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. 2024b. FinDVer: 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, Miami, Florida, USA. Association for Computational Linguistics.

Yilun Zhao, Yitao Long, Hongjun Liu, Ryo Kamoi, Linyong Nan, Lyuhao Chen, Yixin Liu, Xiangru Tang, Rui Zhang, and Arman Cohan. 2024c. DocMath-eval: Evaluating math reasoning capabilities of LLMs in understanding long and specialized documents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16103–16120, Bangkok, Thailand. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

A Appendix

Organization	Model	Size	Source
OpenAI	GPT-4o	–	`gpt-4o-2024-08-06`
Alibaba	Qwen2.5	72B	`Qwen/Qwen2.5-72B-Instruct`
Meta	Llama-3.3	70B	`meta-llama/Llama-3.3-70B-Instruct`
	Llama-3.2	1 & 3B	`meta-llama/Llama-3.2-*B-Instruct`
Mistral AI	Mistral-Small	24B	`mistralai/Mistral-Small-24B-Instruct-2501`
	Mistral	8x22B	`mistralai/Mixtral-8x22B-Instruct-v0.1`
Microsoft	phi-4	14B	`microsoft/phi-4`

Table 3: Details of the LLMs evaluated in this study.

Model	**Attribution Performance**					**Answer Performance**					
	Evidence			**% Exec.**	**Knowl.**	**R-L**	**BS**	**LLM-**	**Numerical**		
	Prec.	**Recall**	**F1**	**Success**	**Recall**			**as-a-judge**	**Prec.**	**Recall**	**F1**
Post-hoc											
GPT-4o	51.0	**75.8**	56.5	9.4	18.1	23.9	87.7	1**3.5**	38.5	**59.8**	**43.4**
Qwen2.5-72B	40.9	69.0	46.6	8.8	18.4	22.9	88.0	12.9	35.4	57.4	40.1
Llama-3.3-70B	47.2	72.5	52.9	19.0	16.9	20.3	87.7	12.9	36.5	56.8	41.1
Llama-3.2-1B	1.4	2.3	1.5	1.6	1.0	16.9	85.1	5.5	13.7	15.9	13.4
Llama-3.2-3B	10.5	18.1	11.8	12.9	8.4	19.6	86.6	8.8	25.5	35.4	26.9
Mistral-Small-24B	52.1	69.2	55.3	5.2	18.9	25.8	88.4	12.5	38.4	56.0	41.9
Mistral-8x22B	38.5	60.4	42.9	9.5	19.3	22.3	87.5	12.0	36.1	47.3	37.3
phi-4	41.6	63.7	46.1	10.0	17.8	20.9	87.2	12.8	34.8	54.1	39.3
End-to-end (single-pass)											
GPT-4o	49.6	74.6	55.2	26.9	16.4	20.2	86.4	13.3	37.6	57.8	41.8
Qwen2.5-72B	48.6	68.8	50.1	15.4	17.2	21.9	87.0	11.9	33.3	45.5	36.1
Llama-3.3-70B	55.0	69.7	57.9	20.5	15.5	21.4	87.2	11.9	37.7	49.3	39.6
Llama-3.2-1B	1.3	1.5	1.3	10.4	0.6	14.2	81.5	4.3	5.0	7.7	5.3
Llama-3.2-3B	11.7	19.6	13.2	21.3	8.5	18.9	85.3	7.4	19.3	27.8	20.8
Mistral-Small-24B	57.1	70.0	**59.2**	13.9	**21.1**	24.0	87.9	12.2	39.0	52.9	40.8
Mistral-8x22B	52.6	69.1	55.8	12.9	20.4	23.5	87.5	12.4	33.6	47.5	36.5
phi-4	44.1	60.7	47.4	18.2	17.4	22.0	86.8	12.4	38.1	48.7	39.4
End-to-end (Iterative)											
GPT-4o	50.2	74.9	55.7	**28.9**	16.0	21.0	85.7	13.2	37.2	57.7	41.7
Qwen2.5-72B	46.8	67.9	52.1	22.5	17.4	22.4	86.6	12.1	36.1	46.2	36.6
Llama-3.3-70B	52.9	69.3	56.3	28.3	15.8	22.3	86.7	12.1	36.6	49.7	39.0
Llama-3.2-1B	0.6	0.6	0.5	9.6	0.0	13.5	81.2	4.1	3.7	4.6	3.6
Llama-3.2-3B	10.1	14.0	10.7	20.3	6.5	20.6	85.7	6.5	16.7	20.0	16.6
Mistral-Small-24B	**56.4**	68.4	58.2	20.0	19.0	25.8	87.7	12.0	**40.1**	51.2	41.2
Mistral-8x22B	43.2	57.6	46.4	17.9	16.2	25.3	87.9	12.4	34.1	46.9	37.2
phi-4	45.4	60.3	48.4	18.2	17.7	25.4	87.5	12.3	39.1	49.8	40.4
Others											
Vanilla	44.3	62.7	49.0	–	–	25.6	88.0	13.4	2.1	2.5	2.0
CoF	40.6	48.6	41.8	–	–	**30.0**	**88.7**	12.2	0.7	0.6	0.6

Table 4: Results of *test* set. R-L denotes ROUGE-L, BS denotes BERTScore. The LLM-as-a-judge evaluation uses a 15-point scale, consisting of three main criteria: accuracy, numerical correctness, and supporting evidence. Each criterion is scored from 1 to 5 points. The detailed breakdown of results for each criterion is shown in the Appendix Table 6.

Model	LLM-as-a-judge		
	Accuracy	Numerical Correctness	Supporting Evidence
Post-hoc			
GPT-4o	4.6	4.4	4.6
Qwen2.5-72B	4.4	4.2	4.4
Llama-3.3-70B	4.4	4.3	4.4
Llama-3.2-1B	1.9	1.8	1.9
Llama-3.2-3B	2.9	3.0	3.0
Mistral-Small-24B	4.2	4.1	4.2
Mistral-8x22B	4.0	4.1	4.0
phi-4	4.4	4.2	4.4
End-to-end (single-pass)			
GPT-4o	4.7	4.4	4.6
Qwen2.5-72B	4.2	4.1	3.8
Llama-3.3-70B	4.1	4.0	3.9
Llama-3.2-1B	1.5	1.5	1.5
Llama-3.2-3B	2.4	2.6	2.5
Mistral-Small-24B	4.2	4.1	4.2
Mistral-8x22B	4.0	3.9	4.1
phi-4	4.2	4.1	4.2
End-to-end (Iterative)			
GPT-4o	4.6	4.3	4.5
Qwen2.5-72B	4.1	3.9	4.1
Llama-3.3-70B	4.1	4.0	4.0
Llama-3.2-1B	1.3	1.3	1.3
Llama-3.2-3B	2.2	2.3	2.3
Mistral-Small-24B	4.1	4.0	4.1
Mistral-8x22B	4.0	4.1	4.0
phi-4	4.2	4.0	4.2
Others			
Vanilla	4.6	4.3	4.4
CoF	4.1	4.0	3.9

Table 5: Breakdown results of LLM-as-a-judge at *development* set.

Model	LLM-as-a-judge		
	Accuracy	Numerical Correctness	Supporting Evidence
Post-hoc			
GPT-4o	4.6	4.3	4.5
Qwen2.5-72B	4.4	4.1	4.4
Llama-3.3-70B	4.4	4.2	4.4
Llama-3.2-1B	1.9	1.7	1.9
Llama-3.2-3B	2.9	2.9	3.0
Mistral-Small-24B	4.2	4.1	4.2
Mistral-8x22B	4.0	4.0	4.0
phi-4	4.3	4.1	4.4
End-to-end (single-pass)			
GPT-4o	4.5	4.3	4.5
Qwen2.5-72B	3.9	3.9	4.1
Llama-3.3-70B	4.0	3.9	3.9
Llama-3.2-1B	1.5	1.4	1.4
Llama-3.2-3B	2.4	2.4	2.5
Mistral-Small-24B	4.1	4.0	4.1
Mistral-8x22B	4.2	4.0	4.2
phi-4	4.2	4.0	4.2
End-to-end (Iterative)			
GPT-4o	4.5	4.2	4.5
Qwen2.5-72B	4.1	4.0	4.1
Llama-3.3-70B	4.1	4.0	4.0
Llama-3.2-1B	1.4	1.4	1.3
Llama-3.2-3B	2.2	2.2	2.2
Mistral-Small-24B	4.0	3.9	4.0
Mistral-8x22B	4.3	4.1	4.0
phi-4	4.2	4.0	4.2
Others			
Vanilla	4.6	4.2	4.4
CoF	4.0	3.9	3.9

Table 6: Breakdown results of LLM-as-a-judge at *test* set.

Answer Generation

You are a professional financial analyst. Your task is to analyze financial reports from two companies and answer a specific question based on the provided information. Your response must be well-structured, concise, and fact-based.

When presenting your analysis, structure your response into numbered clauses.

Each clause should be a single concise sentence presenting a fact-based financial insight.

Avoid sub-bullets, section headers, or formatting symbols (such as dashes, asterisks, or bold text).

Write in full sentences. Do not introduce bullet points or separate explanations within the same clause.

Here is an example of response: *example*

The following are the financial reports of two companies.

Financial Report for Company **<name1>**:
<report1>

Financial Report for Company **<name2>**:
<report2>

Based on the information provided, answer the following question: Question: **<question>**

Figure 3: Prompt of post-hoc answer generation.

Attribution Generation

Given the financial reports of two companies, a question, an answer to the question structured into numbered clauses, and a list of professional knowledge, your task is to identify three attributes for each clause.

The three attributes are:
1. The evidence supporting the clause. It can be a list of paragraph indices from the company's financial report or inferred from previous clauses.
2. Code: If the clause involves numerical values that being calculated from the context, provide a Python function that takes necessary input values, performs the calculation, and returns the result.
3. Professional Knowledge: Identify the professional knowledge items that are relevant to the clause from the professional knowledge list if any.

Here is an example of response: *example*

The following are the financial reports of two companies, a question, and the answer to the question in the numbered clauses format, and a list of professional knowledge. Based on the information provided, identify three attributes for each clause.

Financial Report for Company **<name1>**:
<report1>

Financial Report for Company **<name2>**:
<report2>

Question: **<question>**

Answer to the Question:
<answer>

Professional Knowledge List: **<knowledge list>**

Attributes for each clause:

Figure 4: Prompt of post-hoc attribution generation.

End-to-end Generation

You are a professional financial analyst. Your task is to analyze financial reports from two companies to answer a specific question based on the provided information. For each clause in your response, provide the following three attributes:
1. Evidence: Specify supporting evidence as paragraph indices from the financial reports or indicate if it's inferred from previous clauses.
2. Code: If numerical values are calculated, provide a Python function that performs the calculation and returns the result.
3. Professional Knowledge: Identify relevant professional knowledge items from the provided list, if applicable.

Structure your response into numbered clauses, each with its corresponding attributes.
Here is an example of response: *example*

The following are the financial reports of two companies, a question, and a list of professional knowledge. Based on the information provided, please answer the question in the format of numbered clauses, each with three attributes: Evidence, Code, and Professional Knowledge.

Financial Report for Company **<name1>**:
<report1>

Financial Report for Company **<name2>**:
<report2>

Question: **<question>**

Professional Knowledge List: **<knowledge list>**

Your response:

Figure 5: Prompt of end-to-end generation.

Annotator ID	Finance Industry Experience	Annotation Work
1	1 working and 2 internship at US	Data Annotation
2	2 working and >= 2 internship at US	Data Annotation
3	1 working at UK and 2 internship at US	Data Annotation
4	1 working and >= 3 internship at US	Data Annotation
5	2 internship at US, 1 internship at Canada	Data Annotation
6	Graduate student majored in finance	Data Annotation
7	1 working at Singapore	Annotation validation
8	1 internship at US, 3 internship at China	Annotation validation
9	1 working at Canada	Annotation validation
10	Graduate student majored in data science	Code annotation
11	Graduate student majored in computer science	Code annotation

Table 7: Details of annotators involved in dataset construction.

End-to-end Feedback

You are a professional financial analyst responsible for reviewing and improving a financial analysis. You will be given:

1. Financial reports from two companies.
2. A question related to their financial performance.
3. A current analysis, structured as numbered clauses. Each clause consists of:
- Clause Content: A clear and concise financial insight.
- Attributes:
1) Evidence: References to supporting paragraph indices from the financial reports or logical inference from previous clauses.
2) Code: A Python function performing necessary calculations, including an execution result for verification.
3) Professional Knowledge: Relevant financial concepts from the provided knowledge list.

Provide constructive feedback by evaluating each clause based on the following criteria:
1. Conciseness Relevance to the Question
- Does the analysis answer the question directly and efficiently?
- Does it include unnecessary details or over-explained information? If so, suggest more concise phrasing.
2. Numerical Accuracy
- Verify whether numerical values are correctly calculated based on the financial reports.
- Ensure that all provided calculations are necessary and relevant to answering the question.
3. Evidence Support Justification
- Does each clause rely on valid evidence from the reports? If inference is used, is it logically sound?
- Are financial concepts correctly applied according to the provided knowledge list?
- Does the analysis remain within the provided information, avoiding speculation or unsupported conclusions?
4. Code Accuracy:
- Is the provided code correct, and does its execution result align with the content?
- Professional Knowledge Validity: Are the cited financial concepts appropriate and correctly applied?
Clearly state specific issues and provide actionable suggestions for improvement.
If no corrections are needed, output "Done" without any additional text.

The following are the financial reports of two companies, a question, a list of professional knowledge, and a financial analysis. Based on the information provided, please provide a feedback. If no additional feedback is necessary, output "Done" without any other text.

Financial Report for Company **<name1>**:
<report1>

Financial Report for Company **<name2>**:
<report2>

Question: **<question>**

Professional Knowledge List: **<knowledge list>**

Analysis: **<analysis>**

Your response:

Figure 6: Prompt of end-to-end feedback.

VANILLA Modified Prompt

Instruction: As a finance analyst, write an accurate, engaging, and concise answer for the given question using only the provided search results (some of which might be irrelevant) and cite them properly.

Use an unbiased and journalistic tone. Always cite for any factual claim and every clause must have a citation.

When citing several search results of evidence, use evidence:[0,1,...]. Cite at least one document and at most five documents in each clause (separated by either ',', or '.').

For professional knowledge (calculation formula inside the docs), if they are used, include them in the citation of evidence.

If multiple documents support the sentence, cite some of the documents you think most relevant and needed (do not exclude the professional knowledge because of this).

If this sentence doesn't use any evidence or professional knowledge, then put inference:[] at the end with correct indices. The indices correspond to the source sentences in the answer before this sentence.

Example:

Question: How do changes in depreciation and amortization mean for CWT and AWK's investment strategies in Q1 2024?

Document [1](Title: 0): Depreciation and amortization expense increased $2.9 million, or 9.8%, to $32.8 million for the three months ended March 31, 2024, as compared to $29.9 million for the three months ended March 31, 2023, primarily due to utility plant placed in service in 2023.

Document [2](Title: 1): Income tax expense increased $21.1 million to $15.5 million for the three months ended March 31, 2024, as compared to income tax benefit of $5.6 million for the three months ended March 31, 2023. The increase in income tax expense was primarily due to an increase in the pre-tax operating income for the three months ended March 31, 2024 as compared to the same period of 2023 due to higher pre-tax operating income.

Document [3](Title: 2): Presented in the table below are the company's consolidated results of operations. (Tabular data omitted for brevity.)

Document [4](Title: 3): Table 29: For the Three Months Ended March 31, 2024 and 2023 (In millions)
...
Document [5](Title: 4): Capital Structure = Capital Expenditure Ratio = Capital Expenditures / Total Operating Revenues.

Answer: AWK's depreciation and amortization increase of 9.30% code: [0] and CWT's 9.70% code:[1] increase indicate ongoing capital investments. evidence: [0,3,4] Specifically, AWK's investments imply scaling across multiple projects state-wide, while CWT's more modest absolute amount reflects proportional growth through specific regional infrastructure advances. inference: [0]

Figure 7: Prompt of Vanilla generation.

CoF Stage 1 Answer Generation Prompt

You are a finance analyst give the answer to the question based on the context in a concise and professional tone. Directly give the answer without any extra explanation. Also use comma and punctuation for the answer instead of using "**" .
<*context*>
<*query*>

Figure 8: Prompt of CoF Answer Generation.

CoF Stage 2 Chunk Level Citation

Your task is to add citations to the existing answer for a financial analysis task. In this task, you are provided with an answer that was generated based on specific financial data and contextual information (e.g., interest rate sensitivity tables, financial reports, analytical outputs, or market indicators). Your goal is to enhance the answer by adding reference citations that support any factual claims made.
Don't give "Here's the updated answer with accurate citations based on the snippets provided:" at the beginning of you answer. Directly output your answer without any explanation.
Below are the detailed instructions:
1. Citation Requirement: For each factual statement S in the answer that relies on specific numerical data, analytical results, or direct excerpts from the provided financial context snippets, append the snippet number(s) that support S using the following citation format:
<statement>S<cite>[l1][l2]...[ln]</cite></statement>
- Replace 'S' with the factual statement. - Replace '[l1]', '[l2]', etc. with the respective snippet numbers from the context.
2. No Direct Supporting Evidence: For sentences that function as introductions, summaries, general commentary, reasoning, or inference—where no specific numeric or factual evidence from the provided contexts is used—simply add an empty citation:
<statement>S<cite></cite></statement>
3. Preservation of Original Content: Do not change any part of the original answer's text aside from adding the citation tags. Maintain the original wording, punctuation, and overall structure.
4. Content of Contexts: The provided contexts may include tables, numerical data, and descriptive statements that illustrate financial metrics (such as net interest income sensitivity, funding cost variations, interest rate risk analysis results, etc.). Carefully examine these contexts to identify which snippet(s) support each factual assertion in the answer.
5. Requirements regard the professional knowledge: Some snippet offer some professional knowledges, if these professional knowledge are used, include them. Important: Try to include them if the statement contains some calculations as these professional knowledge is useful for improving the calculation correctness.
6. Example with Extended Snippets and an Additional Sentence:
Suppose the provided contexts and answer are as follows:
[Contexts Start] Snippet [1] The interest sensitivity table for EBC as of March 31, 2024 shows that when interest rates increase by +200 basis points...
Snippet [2] ...
Snippet [3] ...
[Contexts End]
[Question]
How does EBC's net interest income sensitivity change between December 31, 2023, and March 31, 2024 when subjected to a +200 basis points rate change?
[Existing Answer Start]
Based on the simulation, EBC's net interest income sensitivity slightly worsened from December 31, 2023, to March 31, 2024. The increase in funding costs, as shown in the report, contributed to this sensitivity change.
[Existing Answer End]
[Answer with Citations]
<statement>Based on the simulation, EBC's net interest income sensitivity slightly worsened from December 31, 2023, to March 31, 2024.<cite>[1]</cite></statement> <statement>The increase in funding costs, as shown in the report, contributed to this sensitivity change.<cite>[2]</cite></statement>
Now get ready to add citations for the following test case.
[Contexts Start] <context> **[Context End]**
[Question] <question>
[Existing Answer Start] <answer> **[Existing Answer End]**
[Answer with Citations] <statement>Answer sentence 1<cite>[...citation numbers if applicable...]</cite></statement> <statement>Answer sentence 2<cite>[...citation numbers if applicable...]</cite></statement> ...

Figure 9: Prompt of CoF Chunk Level Citation.

CoF Stage 3 Sentence Level Citation

You will receive a passage and a factual statement. Your task is to identify the parts in the passage (i.e., chunks <Cs_1>-<Ce_1>, <Cs_2>-<Ce_2>, . . . , <Cs_n>-<Ce_n>) that support some key points of the statement, and output the chunk numbers in the format:

$$[s\_1 - e\_1]$$

$$[s\_2 - e\_2]$$

$$\ldots$$

$$[s\_n - e\_n]$$

Make sure that you don't include any delimiters (such as triple quotes) in your answer.
Also [1-2] means skipping the first sentence and only includes the second and third sentences.
Directly give your answer without any explanation. Try as hard as possible to include some citation. Some statements may include some calculated number; you need to infer the source of the calculation. If there is no relevant citation, return "None".
If the passage contains no key information relevant to the statement, you must output "No relevant information". Try your best to cite, and if you think some number may result from a calculation, cite the source number for the calculation.
Below are some examples tailored for finance:
Example 1:
[Passage Start]
<C0>EBC reported its net interest income figures for different rate scenarios as part of its interest sensitivity analysis. <C1>At December 31, 2023, EBC's simulated net interest income under a +200 basis points increase showed a change of -2.9%. <C2>At March 31, 2024, the simulation indicated a change of -3.1% under the same scenario. <C3>The table data reflects minor variations between the two reporting dates.
[Passage End]
[Statement]
EBC's net interest income sensitivity increased when the interest rate change is +200 basis points from December 31, 2023 to March 31, 2024.
[Output]
[1-2]
Example 2:
...
Example 3:
...
Now get ready to process the following test case:
[Passage Start] <context> **[Passage end]**
[Statement] <statement>
[Output]

Figure 10: Prompts of CoF Sentence Level Citation