Evaluating LLMs' Mathematical Reasoning in Financial Document Question Answering

Pragya Srivastava [#*], **Manuj Malik** [‡], **Vivek Gupta** [§†], **Tanuja Ganu** [#], **Dan Roth** [§]
[#]Microsoft Research, [‡]Singapore Management University, [§]University of Pennsylvania
{t-pragyasri, taganu}@microsoft.com, manujm@smu.edu.sg, {gvivek,danroth}@seas.upenn.edu

Abstract

Large Language Models (LLMs), excel in natural language understanding, but their capability for complex mathematical reasoning with a hybrid of structured tables and unstructured text remain uncertain. This study explores LLMs' mathematical reasoning on four financial tabular question-answering datasets: TATQA, FinQA, ConvFinQA, and Multihiertt. Through extensive experiments with various models and prompting techniques, we assess how LLMs adapt to complex tables and mathematical tasks. We focus on sensitivity to table complexity and performance variations with an increasing number of arithmetic reasoning steps. The results provide insights into LLMs' capabilities and limitations in handling complex mathematical scenarios for semi-structured tables. Ultimately, we introduce a novel prompting technique EEDP tailored to semi-structured documents, matching or outperforming baselines performance while providing a nuanced understanding of LLMs abilities.

1 Introduction

In the constantly evolving realm of artificial intelligence, Large Language Models (LLMs) have risen as cutting-edge tools for natural language understanding. They excel in a wide array of NLP tasks, including machine translation (MT), text summarization, question answering, and code generation. One specific area where LLMs' mathematical reasoning abilities come under scrutiny is the domain of numerical reasoning tasks. Past research has delved into the potential of language models for mathematical reasoning tasks, as seen in studies such as in Amini et al. (2019); Upadhyay and Chang (2017); Patel et al. (2021); Cobbe et al. (2021). These investigations provide a means to evaluate the performance of language models when it comes to solving mathematical problems, ranging from straightforward math word problems to more complex ones.

The Goldman Sachs Group Incorporation

Notes to Consolidated Financial Statements
The table below presents a summary of Level 3 financial assets.

Financial Asset	Dec. 2017
Cash Instruments	$15,395
Derivatives	$3,802
Other Financial Instruments	$4

Q: What was the total value of Level 3 financial assets for Goldman Sachs in December 2017?

A: $15,395 + $3,802 + $4 = $19,201

Figure 1: An example of a semi-structured financial document question answering.

However, the problem becomes significantly more challenging when we encounter a hybrid of structured such as semi-structured tables and unstructured text, as shown in example in figure 1. Such tables are common in documents such as invoices, health records, and financial reports in corporate settings. In the financial domain, these tables present numerical data in a structured format, including income statements, balance sheets, cash flow statements, shareholder equity data, and annual reports. The majority of NLP models are primarily trained to handle raw unstructured textual data, which limits their ability to reason over semi-structured data, such as tables, or more intricate hybrids of tables and text, as seen in Chen et al. (2020b); Aly et al. (2021); Chen et al. (2020a, 2021a). Tables, especially these with intricate relationships and dependencies, often necessitate multi-hop reasoning, connecting information across multiple steps, as shown in Gupta et al. (2020).

NLP models may encounter difficulties in performing such multi-step reasoning, particularly when dealing with complex mathematical opera-

[*]Work done during internship at Microsoft Research
[†] Primary Mentor and Corresponding Author

tions involving tables, as highlighted in Li et al. (2022). Previous research such as Chen (2023), exemplified these issues and demonstrated LLM capacity to process and reason with semi-structured tables. However, these studies are somewhat constrained and don't explicitly explore the models' mathematical reasoning abilities. This is particularly evident in data/tasks that involve a substantial number of arithmetic reasoning steps, operate on extreme orders of magnitude, or deal with intricate tables where extracting the relevant information for a query becomes challenging.

Moreover, when handling domain-specific documents, such as those in finance, a language model must not only have the necessary domain knowledge to craft the right approach for task-solving but also the capability to manipulate structured data, such as tables. Therefore, in this study, we aim to fill this gap by providing both qualitative and quantitative analyses of LLM's ability to reason on mathematical content on four finance datasets i.e. FinQA (Chen et al., 2021b), TATQA (Zhu et al., 2021), ConvFinQA (Chen et al., 2022b), and Multihiertt (Zhao et al., 2022). These datasets feature questions demanding intricate numerical reasoning, combining semi-structured tables and text. Each dataset provides pre-annotated executable programs for precise answer retrieval. Our goal is to illustrate how model performance varies as the numerical complexity of the underlying data and the intricacy of the mathematical reasoning steps required to solve a query increase. Building upon these observations, we propose a novel approach termed (**E**licit $\longrightarrow$ **E**xtract $\longrightarrow$ **D**ecompose $\longrightarrow$ **P**redict) *EEDP*, designed to deconstruct model responses into discrete components. This innovative method offers a deeper, more transparent insight into the numerical limitations of the model when tackling these tasks. Our contributions are as follows:

1. We conduct a comprehensive robust evaluation of state-of-the-art Large Language Models (LLMs) for tabular (hybrid) question answering, with a specific focus on mathematical reasoning tasks, using public financial tabular datasets to establish a thorough performance benchmark.

2. Our analysis is thorough and multifaceted, encompassing both qualitative and quantitative aspects across several dimensions. We aim to provide nuanced insights into the strengths and limitations of LLMs in tabular (hybrid) question answering, especially in scenarios involving mathematical reasoning.

3. Building upon qualitative analysis, we introduce a novel and improved prompting method called **EEDP**. Our novel approach not only enhances our understanding of model weaknesses but also substantially enhances model performance compared to existing prompting methods across multiple models types.

Our metadata dataset and source code are available at https://vgupta123.github.io/eedp.

2 Metadata Annotations

We annotated four tabular datasets: FinQA, TATQA, ConvFinQA, and Multihiertt with meta information related to a.) reasoning steps, b.) question category, c.) table length, d.) hierarchical complexity e.) missing information. [1] Below, we provide detailed information about these meta-data annotations:

1. Number of Reasoning Steps: Including the count of arithmetic operations in questions is crucial. More operations reflect increased complexity in reasoning, and their interdependence offers insights into the models' proficiency. This annotation, applied across all four datasets, reveals their ability in handling intricate arithmetic tasks. Refer to Figure 8 in Appendix A.4 for distribution of questions based on the number of reasoning steps involved.

2. Question Categorization: In numerical reasoning, grasping the evolution from fundamental arithmetic to advanced operations is crucial, marking a shift in cognitive complexity. As questions advance, they typically involve more intricate combinations of operations and linguistic nuances. Our research identify both the capabilities and limitations of LLMs in understanding these concepts.

We establish 12 mathematical concept categories (Table 1) with corresponding definitions, annotating each question. The dataset coverage across these categories is shown in Figure 9 in Appendix A.4. Notably, categories like DIVISION and RATIO share similarities but differ in focus: DIVISION involves the division operator, while RATIO encompasses ratios, fractions, and inverse problems. CHANGE IN RATIO questions add complexity with quantity changes requiring subtraction. Additionally, we omit NEED-IN-DOMAIN-INFO due

[1] One author annotated the data, and the other checked for accuracy; we took stringent measures to minimize errors.

Concepts	Definition
SUM	Questions that require only the knowledge of addition.
DIFFERENCE	Questions that require only the knowledge of subtraction.
PRODUCT	Questions that require only the knowledge of multiplication.
DIVISION	Questions that require only the knowledge of division.
RATIO	Questions that require knowing fractional forms, e.g., percentages, ratios.
CHANGE RATIO	Questions involving the difference between two fractional forms, e.g., percentage changes, difference in ratios.
RANGE	Questions requiring knowledge of the minimum and maximum of data observations.
COMPARE	Questions necessitating a comparison between mathematical quantities (e.g., greater than, less than).
AVERAGE	Questions needing knowledge of the average, used to calculate the central tendency of a group of data points.
IN-DOMAIN-INFO	Questions that require implicit knowledge to understand domain-specific mathematical formulations (e.g., return on investment (RoI), cost of goods sold (COGS), amortization rate, etc.).
TIME	Questions explicitly involving mathematical operators for time-spans not in the table or context.
COUNTING	Questions requiring the counting of elements in a set or group of data points.

Table 1: Mathematical concept categories and definitions for studying LLM concept comprehension abilities.

to domain-specific knowledge focus and TIME questions due to limited sample size.

3. Table Length: Evaluating performance with larger supporting tables is crucial. Larger tables complicate multi-hop reasoning tasks by increasing the amount of information, making it harder to identify relevant evidence. We prioritize these annotations for datasets like FinQA and Multihiertt, where questions mainly use tables as supporting evidence. Therefore, these annotations are confined to these datasets. In Multihiertt, when multiple tables support evidence, we consider the one with the highest row count i.e. maximum table length. The dataset distribution for Multihiertt and FinQA w.r.t table length (number of rows) is shown in Figure 6 in Appendix A.4.

4. Hierarchical Complexity: In hierarchical tables, such as those in Multihiertt, evaluating model performance concerning the growing hierarchical complexity in cells with critical information becomes paramount. To tackle this, we annotate each example in Multihiertt with the hierarchy depth of cells containing relevant information. For table with multiple relevant cells, we consider the cell with the highest hierarchical depth for our analysis. Our approach to estimating hierarchy depth is illustrated in Figure 10. Figure 7(a) in Appendix A.4 illustrate how we calculate hierarchical complexity for examples with multiple relevant rows at various hierarchical depths.

5. Missing Information: Interpreting a table becomes challenging as the number of empty cells increases. Empty cells indicate missing or undefined information, leading to potential gaps in understanding.

Assessing *empty cell proportions* is crucial to quantify data ambiguity. More empty cells suggest higher uncertainty, which can hinder models' ability to derive meaningful insights and impact reasoning accuracy. In Multihiertt, where tables are hierarchical in nature and empty cells occur quite frequently, we annotate examples with the empty cells percentage, contributing to our understanding of data ambiguity. For distribution of missing information (empty cells proportions) across datasets, refer to Figure 7 (b) in Appendix A.4.

Annotation Splits. We prioritized complex numerical questions in our selection criteria, balancing this with resource constraints such as the LLM context length limits. We also took into account tables with deeper hierarchies in Multihiertt and multi-turn conversations in ConvFinQA. For TATQA, we utilized 45% of the development set by filtering out examples involving simple span selection. In the case of Multihiertt, we included 68% of the test set by excluding examples where the table length exceeds 40. For FinQA and ConvFinQA, we employed the complete test and development sets, respectively.

3 Experimental Results

In this study, we choose to experiment with LLMs such as `GPT-3.5-Turbo`, `GPT-4`, `PaLM-540B`, `Mistral-7B-Instruct`[2], `Llama-2-13B`[3] and `MAmmoTH-13B`[4]. These LLMs are at the cutting edge for both open-source and closed models applications. Models like `MAmmoTH-13B` are specifically fine-tuned during pre-training to excel in mathematical reasoning tasks. For more detail about the the model choices refer to Appendix A.2.

[2]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1

[3]https://huggingface.co/meta-llama/Llama-2-13b-chat-hf

[4]https://huggingface.co/TIGER-Lab/MAmmoTH-13B

LLMs Prompting Methods: For an instruction-tuned LLM, it's assumed that we give the model a *task-specific* instruction $\mathcal{I}$ accompanied with a few (usually $k \in \{2, 4\}$) demonstrations $\mathcal{D}_\mathcal{T}$ of a task T. We experiment with the following prompting techniques:

(a.) **Direct:** In this setup, we explicitly instruct the models to abstain from providing explanations and just return the final answer to the posed question. For this scenario, $\mathcal{D}_\mathcal{T}$ contains $\{(p_i, q_i, a_i)\}_{i=1}^{k}$ where p is the premise (table-text), q is the question, and a is the ground-truth answer.

(b.) **CoT:** Moving forward, we experiment with the *chain-of-thoughts* prompting strategy where we instruct the model to output the explanation to the answer derived by it. Here, our $\mathcal{D}_\mathcal{T}$ contains $\{(p_i, q_i, e_i)\}_{i=1}^{k}$ where p is the premise which includes the table and the associated text, q is the question and e is the explanation of the answer.

(c.) **PoT:** In this case, the expected response is a code derivation of the answer. Here, $\mathcal{D}_\mathcal{T}$ contains $\{(p_i, q_i, c_i)\}_{i=1}^{k}$ where p is the premise which includes the table and the associated text, q is the question and c is the code-derivation of the answer.

(d.) **Decomposers:** (Ye et al., 2023) proposed to address the challenge of handling large tables by decomposing them into more manageable subtables. Similarly, complex questions are handled by breaking them down into simpler subquestions. Decomposition proves effective with SQL tables, facilitating the removal of distracting details while retaining all supporting evidence. Questions are first parsed to break them down into simpler, more manageable subquestions. The model then addresses each subquestion independently before composing the answers to arrive at the final solution. In this case, our demonstration set $\mathcal{D}_\mathcal{T}$ contains $\{(p_i^{'}, \langle q_1, q_2, ..., q_n \rangle, a_i)\}_{i=1}^{k}$ where $p^{'}$ is the premise obtained by the irrelevant information removal to the question from the original premise p and $\langle q_1, q_2, ..., q_n \rangle$ are the subquestions whose answers lead to the final answer.

EEDP Prompting Strategy: We propose a *novel prompting strategy*: **E**licit $\longrightarrow$ **E**xtract $\longrightarrow$ **D**ecompose $\longrightarrow$ **P**redict. Figure 4 show an illustration of our EEDP approach. Below are the detail of each EEDP step:

1. **Elicit:** We prompt the model explicitly to first *elicit* relevant domain knowledge for answering a given query.
2. **Extract:** Conditioned on the table, question and the elicited domain knowledge, the model extracts supporting evidences to answer a given question.
3. **Decompose:** We instruct the LLM to break a complex mathematical reasoning task into multiple atomic operations and compose the operations to arrive at the final answer.
4. **Predict:** The model finally returns the derived answer in the above steps.

Figure 11 shows a example for EEDP strategy with one shot.

Results and Analysis. Table 2 shows a comparison in performance between different prompting strategies. Despite being a single prompt, EEDP demonstrates comparable or superior performance compared to PoT. Notably, we outperform PoT significantly for `PaLM-2-540B` and `LLAMA-2-13B` across most datasets. Moreover, while PoT relies on external tools for executing mathematical programs/code to obtain answers, EEDP exclusively utilizes LLM for all tasks, including evidence extraction, operation identification, and execution, ensuring precision throughout the process.

As shown in Table 2, the Decomposers prompting strategy exhibits a much poorer performance compared to other strategies. The reason behind this was statistically found to be the inaccurate formation of subtables from the main table, leading to information loss as described in the previous paragraph. The performance of EEDP either surpasses or matches very closely with that of PoT. The number of shots was adjusted depending on the context length of the model.

We can see that `MAmmoTH-13B` model, which is fine-tuned on the `MathInstruct` dataset (Yue et al., 2024) containing Instruction-Response pairs where the responses are a hybrid of CoT and PoT rationales, fails to perform well with the EEDP methodology. We argue that this is due to two potential reasons: (a.) Reduction of the number of shots to adjust the context length as the EEDP response is longer than that of the other methods, and (b.) Finetuning may contribute to suboptimal performance due to its alignment with a particular style and format of responses, potentially limiting the model's adaptability and generalization to other diverse contexts.

Dataset	Model	Direct	CoT	PoT	EEDP	Decomposers
TATQA	GPT-4	55.81	86.91	**89.99**	88.67	47.46
	GPT-3.5-Turbo	31.38	77.57	**82.11**	79.73	28.53
	PaLM 2-540B	44.66	62.93	61.60	**81.51**	57.94
	Llama 2-13B	3.36	35.95	34.16	**40.95**	25.93
	MAmmoTH-13B	19.11	**56.25**	10.02	4.37	22.89
	Mistral-7B	10.92	**59.14**	16.53	56.06	7.24
FinQA	GPT-4	65.12	72.38	75.26	**76.05**	44.93
	GPT-3.5-Turbo	40.47	59.18	**68.97**	61.88	32.33
	PaLM 2-540B	30.33	34.79	30.41	**61.95**	46.38
	Llama 2-13B	1.80	25.34	12.97	**30.47**	11.91
	MAmmoTH-13B	22.83	**35.32**	15.86	35.05	17.65
	Mistral-7B	26.11	34.23	10.56	**34.86**	12.34
ConvFinQA	GPT-4	63.10	71.19	**78.81**	77.91	18.76
	GPT-3.5-Turbo	37.62	48.33	61.19	**61.75**	10.50
	PaLM 2-540B	20.19	38.00	40.14	**63.42**	22.32
	Llama 2-13B	3.80	29.45	29.92	**39.42**	10.35
	MAmmoTH-13B	21.61	**46.08**	8.78	32.77	7.83
	Mistral-7B	12.35	**48.45**	14.48	36.57	11.16
Multihiertt	GPT-4	41.35	55.13	67.23	**70.32**	36.86
	GPT-3.5-Turbo	25.88	42.33	**52.18**	49.65	20.61
	PaLM 2-540B	14.20	20.67	36.52	**37.97**	20.19
	Llama 2-13B	1.54	**30.66**	18.12	24.15	16.86
	MAmmoTH-13B	10.12	**18.56**	6.57	18.36	11.87
	Mistral-7B	14.909	**22.92**	14.94	10.97	11.63

Table 2: Comparison of performance of different models tested against a variety of prompting strategies

EEDP's Computational Efficiency EEDP functions as a unified single-prompt method, minimizing computational complexity. Unlike methods like PoT, which rely on external tools, EEDP operates independently. When assessing computational cost, we consider API calls and token generation. Since EEDP uses a single-step prompting approach, only one API call is needed per query, making its computational cost comparable to methods like CoT. For inference with open-source models, we used hardware with an A40 40GB GPU. Processing one dataset per model using the vLLM library took approximately 10 hours.

4 Where do LLMs fail?

Through manual inspection, we rigorously evaluate the EEDP responses against the meta-annotations from section 2 as ground-truth benchmarks for extraction and model reasoning accuracy. The reasoning programs represent sequences of arithmetic operations necessary to derive the final answer, utilizing values extracted from supporting evidence as operands. To assess calculation accuracy, we manually identify the model's instantiation and precision errors. Our EEDP prompt ensures that the model predominantly outputs responses in the expected format, with exceptions being rare. However, since we manually analyze all outputs, we do not penalize the model for format deviations but rather for incorrect outputs. Penalties are applied only when the model makes errors in extraction, reasoning, and/or calculation. Below, we categorize the EEDP response errors in detail based on their origins:

Dataset	Error	Type	Per.(%)
FinQA	Extraction	E1	10.38
		E2	25
	Reasoning	R1	25
		R2	15.57
	Calculation	C1/C2	24.06
ConvFinQA	Extraction	E1	8.45
		E2	14.08
	Reasoning	R1	19.72
		R2	36.62
	Calculation	C1/C2	21.13
TATQA	Extraction	E1	13.79
		E2	31.03
	Reasoning	R1	22.41
		R2	5.17
	Calculation	C1/C2	27.59
Multihiertt	Extraction	E1	20.5
		E2	31.5
	Reasoning	R1	15.5
		R2	12
	Calculation	C1/C2	20.5

Table 3: Error Analysis on Various Datasets. In this table, Extraction.E1: Missing Evidences, Extraction.E2: Wrong Evidences, Reasoning.R1: Insufficient Domain Knowledge, Reasoning.R2: Question Misinterpretation, Calculation: Instantiation (C1) and Precision errors (C2)

1. Incorrect Extraction: This category encompasses errors where the model faces difficulties

in accurately identifying and extracting the pertinent information necessary for effective problem-solving. These errors point to challenges in retrieving precise information. These errors can further be subdivided into two categories

- **Missing/Incomplete Evidences (E1):** The model fails to extract all the necessary evidences which serve as ingredients to derive the final answer.
- **Wrong Evidences (E2):** The model extracts wrong values for variables as supporting evidences from the premise.

2. Incorrect Reasoning: Errors in reasoning occur when the model struggles to formulate an appropriate and contextually relevant approach to tackle a given problem. Possible reasons include a lack of domain knowledge or an inaccurate interpretation of the posed question. Consequently, this error type can arise from two sources.

- **Deficit in Domain Knowledge (R1):** These errors occur when the model attempts to derive an answer to the posed question using a wrong formula for *domain-specific* measures, for eg. COGS, ROI etc.
- **Question Misinterpretation (R2):** These errors occur when the model interprets the question differently and provides responses that are not aligned with the intended query. Overall, the model's outputs lack grounding in the original question posed to it.

3. Incorrect Calculation: This variety of errors include those where the model commits mistakes due to calculation mistakes. This can be of two types as described below.

- **Incorrect Instantiation (C1):** These include cases if the model extracts the right evidences, uses the right derivation formula but instantiates the variables incorrectly with the values resulting in an incorrect answer.
- **Precision Error (C2):** Language models employ mathematical algorithms for arithmetic operations, but their results may not always be perfectly accurate due to insufficient data pattern coverage or introduced biases during training. Consequently, they can sometimes generate outputs with slight inaccuracies or deviations from correct results. We show a detailed analysis in A.1.

Analysis: The above categorization provides a nuanced understanding of the diverse challenges and shortcomings exhibited in different facets of mathematical reasoning. We observe that in a lot of cases, the error propagates because of a deficiency in domain knowledge. It is critical for both evidence extraction and reasoning. Despite possessing general domain knowledge owing to the massive amount of data these models have been pre-trained upon, these models may require explicit prompts to elicit the specific domain knowledge needed for a particular question. Furthermore, errors can arise due to the models' limited proficiency in multi-step reasoning, especially in tackling questions involving multiple arithmetic operations in a sequence.

We give a quantitative measure of each type of errors for each of the 4 datasets we consider for our study in Table 3. We also provide examples corresponding to each error category in figures 12, 13, 14, 15, 16 and 17. Statistically, we find that reasoning errors contribute a significant chunk to the total number of errors. In case of complex hierarchical tables like that in Multihiertt, the model is found to struggle with extracting the right supporting evidences from the premise for a given question. Calculation errors can be taken care of if a third-party calculation tool (an external agent) is chained to the language model.

5 Analysis on Reasoning Annotations

We analyse model performance on the basis of fine-grained annotations as described in the section 4.

1. Performance vs Number of Reasoning Steps. We investigate model performance with increasing mathematical reasoning steps, as shown in Figure 2. This analysis provides insights into models' ability to handle varying task complexities. As expected, performance decreases with more reasoning steps, indicating LLMs' challenges in retrieving information and reasoning as complexity grows.

Anomalies are observed in ConvFinQA, where accuracy improves after greater than or equal to two reasoning steps, potentially due to questions referring to answers of prior conversation turns. Anomalies like these warrant further investigation beyond this study's scope.

2. Performance across Question Types. We analyze the performance trends across different question categories, as defined in Table 1, to assess the models' understanding of various mathematical

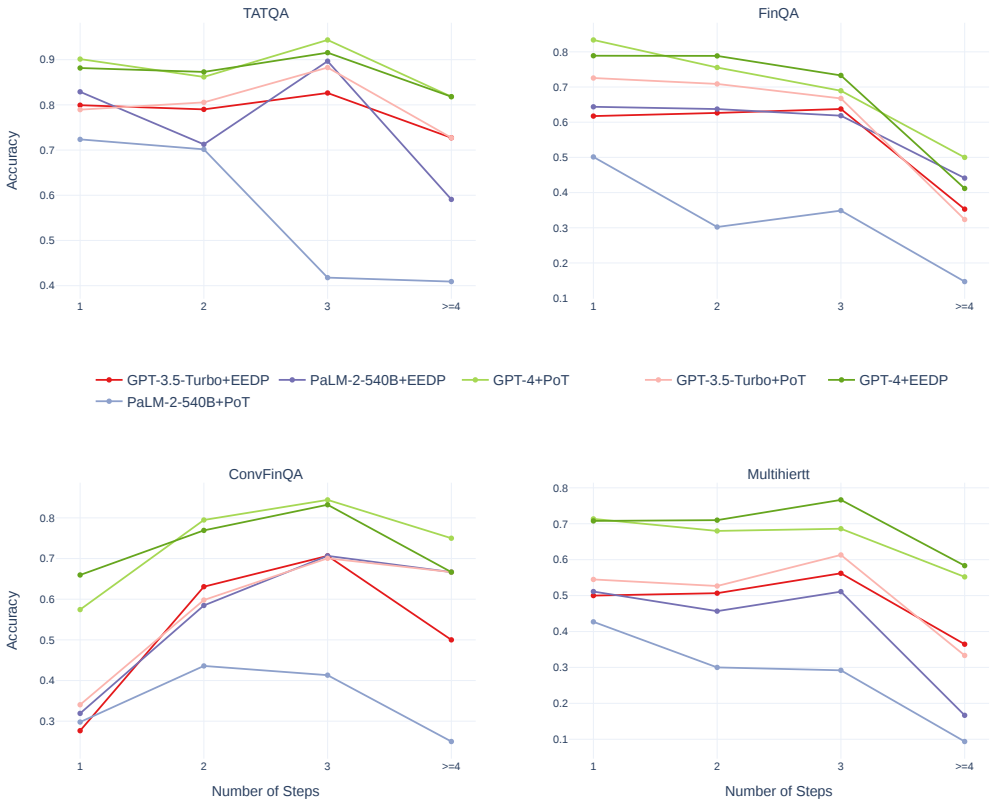


Figure 2: A comparison showcasing the performance trends across various datasets with the increasing number of reasoning steps. The analysis contrasts the effectiveness of EEDP (our method) against PoT in addressing complex reasoning.

and financial concepts. Figure 3 shows that EEDP consistently performs better than or as well as PoT across all datasets. The improvement is particularly pronounced for `PaLM-2-540B` in all question categories.

3. Performance across Arithmetic Operations. Figure 5 in Appendix A.1 shows that for relatively simpler arithmetic operations like addition and subtraction, the effect of order of magnitude of the operands is less profound as compared to harder arithmetic operations such as multiplication and division. We observe the trend in the performance accuracy with the growing and diminishing orders of magnitude. We also observe bigger and more capable models such as `GPT-4`, `GPT-3.5-TURBO` and `PaLM 2-540B` perform much better on simpler addition, subtraction task in comparison to multiplication, division task. For more details on refer to the Appendix A.1.

6 Other Related Works

6.1 LLMs on Mathematical Reasoning

Pre-trained Language Models (PLMs) excel in NLP tasks (Devlin et al., 2019; Zhuang et al., 2021) by leveraging extensive textual corpora to acquire world knowledge (Guu et al., 2020). Expanding PLMs for math-related tasks has been challenging due to their non-specific training. Recent attempts include MWP-BERT and Minerva (Liang et al., 2022; Lewkowycz et al., 2022), but curating high-quality math data remains difficult. To bridge the gap, researchers fine-tune PLMs for specific math tasks. Notable works, like Bhaskara, Self-sampler, Aristo, FinQANet, TAGOP, MT2Net, and others (Mishra et al., 2022; Ni et al., 2022; Clark et al., 2021; Chen et al., 2021b; Zhu et al., 2021; Zhao et al., 2022; Cao and Xiao, 2022; Welleck et al., 2022), employ PLMs such as GPT-Neo and RoBERTa for math problem-solving.

6.2 Tabular Question Answering

Handling diverse input formats in question answering, including structured tables and visual data, poses challenges for language models. HybridQA (Chen et al., 2020b) introduces questions requiring reasoning over tables and text. MultimodalQA (Talmor et al., 2021) adds visual inference. Our focus is on multi-hop question answering over tables and text. TAPAS (Herzig et al., 2020) tackles table-

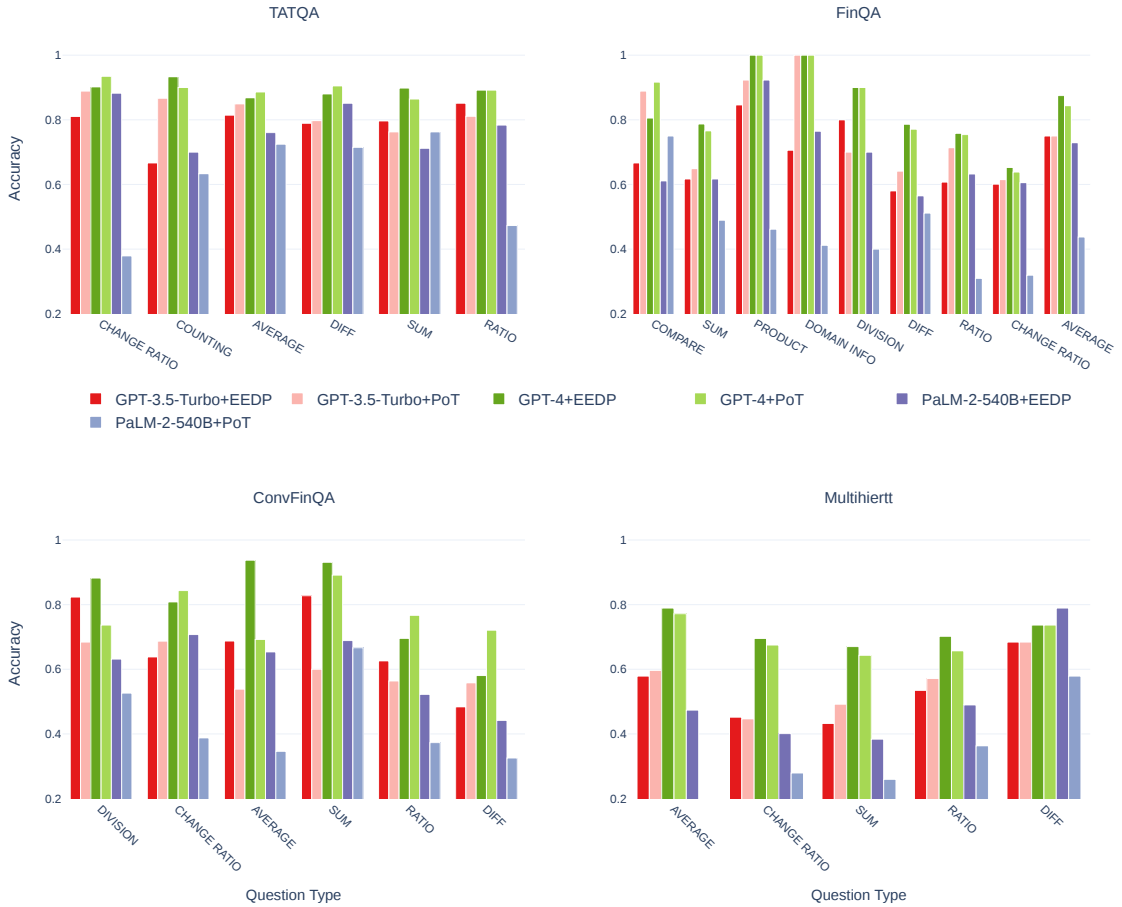


Figure 3: A comparison showcasing the performance trends observed in various datasets across different question types. The analysis contrasts the effectiveness of EEDP (our method) against Few-Shot PoT (PoT). Best viewed in color.

based questions without logical forms, while Tapex (Liu et al., 2022) empowers generative models with table reasoning.

Models like FinQANet, TagOP, and MT2Net (Chen et al., 2021b; Zhu et al., 2021; Zhao et al., 2022) employ a retriever module to extract supporting facts from input financial reports, followed by a reasoning module to derive the final answer. Retrieving relevant evidence and conducting reasoning both demand domain-specific knowledge, such as understanding financial transactions, identifying revenue trends, and interpreting complex investment statements. Thus, reliance on an external knowledge base becomes crucial for addressing the challenges of domain-specific multi-hop question answering.

6.3 Prompt Engineering

In-context Learning (ICL) equips Language Models (LLMs) with task examples and queries, enabling them to perform target tasks without updating model parameters (Brown et al., 2020; OpenAI, 2023). They excel in mathematical reasoning with few-shot prompts but struggle with more complex tasks. Methods like chain-of-thoughts (CoT) (Wei et al., 2022) have improved LLM performance by guiding them through intermediate reasoning steps. Enhancing multi-step reasoning involves two main approaches: improving in-context examples and obtaining better reasoning steps. Some focus on stable in-context example selection (Fu et al., 2023; Rubin et al., 2022; Lu et al., 2023). Others adopt a modular approach, using off-the-shelf tools (Schick et al., 2023), *program of thoughts* (PoT) (Chen et al., 2022a), or decomposition strategies (Zhou et al., 2023; Dua et al., 2022; Khot et al., 2023).

Our task requires complex multi-step reasoning across diverse information sources. LLMs, as demonstrated by (Chen, 2023), can reason over structured tables without explicit encoding. They also serve as versatile decomposers, breaking down extensive evidence and complex questions (Ye et al., 2023).

7 Key Takeaways

Our Contributions. Our study aimed to delve into the mathematical reasoning abilities of LLMs within the context of financial documents where models are tasked with complex hybrid (table-text) question answering. This presents a significant challenge, requiring models not only to provide accurate numerical analysis but also to retrieve right supporting evidence tailored to specific question requirement. Moreover, it necessitates the extraction of necessary knowledge from the model's pretrained parameters to address queries.

Firstly, we meticulously annotate popular financial datasets, such as FinQA, ConvFinQA, TATQA, and Multihiertt, with detailed meta-information. This includes specifying mathematical operations, types of reasoning involved, table dimensions, question types, and the depth of table hierarchy. Furthermore, we conduct a manual error analysis to quantify error types across multiple LLMs. These detailed annotations are invaluable for analyzing various dimensions where LLM models encounter challenges. This, in turn, aids in the development of better prompting techniques such as EEDP, aimed at enhancing LLMs' mathematical reasoning abilities. The resulting improvement in performance with EEDP across multiples datasets serves as compelling evidence of the effectiveness of this approach.

What did we learn? Our analysis revealed that LLMs can accurately handle addition and subtraction tasks e.g. modle perform fairly when calculating total expenses or profits, but struggle with multiplication and division e.g. model performs poorly with questions requiring reasoning operations involving proportions, ratios, percentages, and division. Moreover, as the complexity of the data increases either through a higher absolute order scale or more decimal numbers, model performance degrades. Model performance also degrades with increasing number of reasoning steps and lengthy complex hierarchical table structures. e.g. in complex datasets with hierarchical structures such as Multiheirtt, TATQA, incorrect extraction leads to modeling errors. Similarly, on queries involving multiple conversational turns, such as those in ConvFinQA, model perform poorly due to reasoning failures, like misinterpreting multiple queries longer context. Across all models, incorrect reasoning and incorrect extraction consistently emerge as common sources of errors. For smaller models, even straightforward calculations, result in errors due to imprecise calculations.

EEDP vs other methods

(a.) EEDP vs PoT: PoT enhances LLM inference with the use of variable names for the supporting values extracted from the premise and prompts the LLM to express their thought process in the form of programs. The model output is a program which is executed externally to derive the final answer. EEDP proposes to decompose a complex reasoning task into simple atomic steps whose solutions can be composed to give the final answer. In PoT, they don't make the language model do the computation while in our case the language model not only outputs the reason but also computes the final answer. This distinction implies that PoT may have an inherent advantage over EEDP.

(b.) EEDP vs Decomposers: The prompting strategy proposed by (Ye et al., 2023) was originally designed for querying SQL tables, they use LLMs to break down evidence and questions for SQL interpreters. In contrast, our approach addresses more complex scenarios involving both tables and text, requiring advanced reasoning skills beyond the capability of standard SQL interpreters. Pruning a non-SQL table using this method can lead to significant information loss from the premise which can be a potential ingredient required to derive the final answer. Additionally, this is an expensive method as it requires 3X API calls as opposed to other prompting methods. Moreover, EEDP is a unified prompting strategy which integrates multiple solver elements into a single unified prompt for elicitation, extraction, decomposition and prediction.

8 Conclusion

In conclusion, our study delved into LLMs' mathematical reasoning in complex financial scenarios, assessing their adaptability to mixed structured tables and unstructured text. Through rigorous experimentation, we uncovered insights into their performance and limitations, presenting a tailored prompting technique that outperformed other baseline methods. Our findings advance understanding of LLMs' abilities in tackling intricate mathematical tasks within semi-structured documents, suggesting directions for future research. Please refer to appendix section A.3 for future work details.

Limitations

The scope of this work is limited by the following challenges:

Dataset Scarcity. There are not many datasets dealing with numerical reasoning over semi-structured data apart from the ones from financial domain. As a future work, it would be interesting to similar analysis across various domains such as e-commerce, healthcare, sports and scientific tables from research papers, uncovering new challenges and insights. This expansion will enhance the applicability and impact of our research within the NLP community. However, creating tailored datasets for these domains presents a significant challenge.

For now to ensure a comprehensive evaluation of LLMs, we have integrated financial datasets that offer diverse challenges. For instance, Multihiertt evaluates model performance with intricate premise structures, providing insights into handling complex data hierarchies. ConvFinQA delves into the intricate chains of numerical reasoning within conversational question answering contexts, offering a unique perspective on dynamic data interpretation. Moreover, FinQA and TAT-QA encompass a wide array of reasoning types, with a significant portion requiring domain-specific knowledge, thereby broadening the evaluation spectrum.

Factors Isolation. It is essential to acknowledge that there may be multiple factors influencing the performance of large language models while dealing with numerical reasoning over semi-structured data. In our analysis, we have focused on specific factors and trends, but it is important to recognize that the overall performance is affected by a multitude of variables. Marginalizing i.e. observing the trend along one while keeping the rest as constants or isolating a single factor is challenging and cannot be done with real-world data. Additionally, future investigations may benefit from simulating controlled scenarios on synthetic and counterfactual datasets to gain deeper insights into the impact of individual factors on model performance.

Modeling Improvement. We emphasize our analysis on prominent models such as `GPT-4`, `GPT-3.5-TURBO`, and `PaLM 2-540B` due to their substantial size and capabilities. Notably, other open-sourced large language models like `LLaMA 2-13B`, `MAmmoTH-13B` and `Mistral-7B-Instruct` did not exhibit satisfactory performance in numerical reasoning over semi-structured data. For more detail about the the model choices refer to Appendix A.2. This accentuates the need for exploring computationally feasible and cheaper models that can deliver remarkable performance in tasks involving numerical reasoning over heterogeneous sources of information. Future experiments with ample computational resources may involve exploring larger open-source models like OLMo, Mixtral, and DBRX, which have been recently released.

Ethics Statement

We, the authors of this work, affirm that our work complies with the highest ethical standards in research and publication. In conducting this research, we have considered and addressed various ethical considerations to ensure the responsible and fair use of computational linguistics methodologies. We provide detailed information to facilitate the reproducibility of our results. This includes sharing code, datasets (in our case, we deal with publicly available datasets and comply to the ethical standards mentioned by the authors of the respective works.), and other relevant resources to enable the research community to validate and build upon our work. The claims in the paper match the experimentation results, however, with *black-box* large language models, a certain degree of stochasticity is expected which we attempt to minimize by keeping a fixed temperature. We describe in the fullest details the annotations, dataset splits, models used and prompting methods tried, ensuring reproducibility of our work.

Acknowledgements

Research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-20-1-0080. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. This work was partially funded by ONR Contract N00014-19-1-2620. Lastly, we extend our appreciation to the reviewing team for their insightful comments.

References

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Jie Cao and Jing Xiao. 2022. An augmented benchmark dataset for geometric question answering through dual parallel text encoding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1511–1520, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Wenhu Chen. 2023. Large language models are few(1)-shot table reasoners. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1120–1130, Dubrovnik, Croatia. Association for Computational Linguistics.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2022a. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020a. Tabfact : A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia.

Wenhu Chen, Ming wei Chang, Eva Schlinger, William Wang, and William Cohen. 2021a. Open question answering over tables and text. *Proceedings of ICLR 2021*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020b. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021b. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022b. ConvFinQA: Exploring the chain of numerical reasoning in conversational finance question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6292, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Peter Clark, Oren Etzioni, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, Sumithra Bhakthavatsalam, Dirk Groeneveld, Michal Guerquin, and Michael Schmitz. 2021. From 'f' to 'a' on the n.y. regents science exams: An overview of the aristo project.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dheeru Dua, Shivanshu Gupta, Sameer Singh, and Matt Gardner. 2022. Successive prompting for decomposing complex questions. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1251–1265, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. INFOTABS: Inference on tables as semi-structured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324, Online. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 3843–3857. Curran Associates, Inc.

Chenying Li, Wenbo Ye, and Yilun Zhao. 2022. FinMath: Injecting a tree-structured solver for question answering over financial reports. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6147–6152, Marseille, France. European Language Resources Association.

Zhenwen Liang, Jipeng Zhang, Lei Wang, Wei Qin, Yunshi Lan, Jie Shao, and Xiangliang Zhang. 2022. MWP-BERT: Numeracy-augmented pre-training for math word problem solving. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 997–1009, Seattle, United States. Association for Computational Linguistics.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022. TAPEX: Table pre-training via learning a neural SQL executor. In *International Conference on Learning Representations*.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2023. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *International Conference on Learning Representations (ICLR)*.

Swaroop Mishra, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Welleck, Chitta Baral, Tanmay Rajpurohit, Oyvind Tafjord, Ashish Sabharwal, Peter Clark, and Ashwin Kalyan. 2022. LILA: A unified benchmark for mathematical reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5807–5832, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ansong Ni, Jeevana Priya Inala, Chenglong Wang, Alex Polozov, Christopher Meek, Dragomir Radev, and Jianfeng Gao. 2022. Learning math reasoning from self-sampled correct and partially-correct solutions. In *The Eleventh International Conference on Learning Representations*.

OpenAI. 2023. Gpt-4 technical report.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. 2021. Multimodalqa: Complex question answering over text, tables and images.

Shyam Upadhyay and Ming-Wei Chang. 2017. Annotating derivations: A new evaluation strategy and dataset for algebra word problems. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 494–504, Valencia, Spain. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*,

volume 35, pages 24824–24837. Curran Associates, Inc.

Sean Welleck, Jiacheng Liu, Ximing Lu, Hannaneh Hajishirzi, and Yejin Choi. 2022. Naturalprover: Grounded mathematical proof generation with language models. In *Advances in Neural Information Processing Systems*.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 174–184, New York, NY, USA. Association for Computing Machinery.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2024. MAmmoTH: Building math generalist models through hybrid instruction tuning. In *The Twelfth International Conference on Learning Representations*.

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)*, pages 6588–6600, Dublin, Ireland. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Appendix

A.1 How proficient are LLMs in performing simple arithmetic operations?

To assess the effectiveness of Large Language Models in handling fundamental arithmetic tasks (addition (+), subtraction (-), multiplication (*), and division (/)) across operands of varying magnitudes, we generate a set of 2600 synthetic arithmetic expressions using `GPT-4`. This set includes 650 problems for each arithmetic operation. Within each operation category, we categorize tasks into groups based on a parameter denoted as τ:

$$\tau = \text{OOM}(\ \arg\max_{\text{op}} \|\text{OOM}(\text{op})\|\ )$$

where, $\arg\max$ selects the operand op with the greater absolute value of the order of magnitude, and OOM represents the order of magnitude.

This approach is motivated by cognitive challenges commonly faced by humans, as they often encounter difficulties with both high and low orders of magnitude. Essentially, captures the order of magnitude of the operand with the larger absolute value among the two. For each arithmetic operation, we establish groups with τ, ranging from -6 to 6. Within each group, there are 50 arithmetic expressions. This systematic grouping provides a comprehensive assessment across various difficulty levels based on operand magnitudes.

Analysis. Figure 5 illustrates that for simpler arithmetic operations like addition and subtraction, the impact of the order of magnitude of the operands is less significant compared to harder operations like multiplication and division. We observe a trend in performance accuracy with increasing and decreasing orders of magnitude. Larger models such as `GPT-4`, `GPT-3.5-TURBO`, and `PaLM 2-540B` perform significantly better on addition and subtraction tasks as compared to the multiplication and division tasks.

A.2 Model Selection Criteria

Our model selection process was guided primarily by resource constraints and the timeframe of our research endeavor. We aimed to identify models that represented the state-of-the-art (SOTA), such as GPT-4, or those with a specific focus on mathematical reasoning, such as `MAmmoTH`, aligning with the parameters of our project. Here's a breakdown:

1. **Resource and Budget Constraints:** Given our limited computing resources and budget, we meticulously selected models that could provide valuable insights within the confines of our project. Incorporating additional models would have been impractical given our resource limitations.

   The number of shots, indicated by shots = k, was tailored to the context length of the model.

Actual Prompt: <Instructions>

Please ***carefully analyze*** the provided information, perform ***necessary numerical calculations***, and ***accurate answers*** to the given question using the provided data.

Take into ***account the pre-text, table, and post-text*** when formulating your response. When performing numerical calculations, ensure you *gather the* ***required information and follow a step-by-step approach.***

elicit → ***Elicit the most relevant domain fact or knowledge*** that might be useful for you to extract the right operands and **forming the right approach to the problem.**

extract → ***Translate table data into sentences and identify gold evidence*** for answering the question.

decompose → Define a **sequence of *atomic operations*** *(Add, Subtract, Divide, Multiply, Greater, Min, Max, Exp etc.)* which take into **account only one or two operands** at a time.

predict → Divide a ***complex task into a sequence of atomic operations*** defined above. Finally, conclude as : ***The final answer is <answer>.***

Here are a few demonstrations of the task: ***<Demonstration>***

Actual Prompt: <Demonstration>

Effective Income Tax Rate: A reconciliation of the United States federal statutory income tax rate to our effective income tax rate is as follows:

Description	Dec 31, 2018 (%)	Dec 31, 2019 (%)
US Federal Statutory Rate	21.00	21.00
Effective Income Tax Rate	1.99	-0.01

In 2019 and 2018 we had pre-tax losses of $19,573 and $25,403, respectively, which are available for carry forward to offset future taxable income.

Question: What was the 2019 percentage change in pre-tax losses?
Output:

elicit → **Domain Knowledge:** Pre-tax losses, or operating losses, refer to financial losses that a company incurs before considering the effects of income taxes.
To find the 2019 percentage change in pre-tax losses, we need to find the difference between the new and the old value of the pre-tax losses, then divide the obtained difference by the old value and multiply this value by 100.

extract → **Gold Evidences:** The **pre-tax losses** in **2019 is $19,573.** The pre-tax losses in **2018 is $25,403.**
Steps:

Step 1: Subtract 25403 from 19573. → *Response 1:* **19573 - 25403 = -5830**
Step 2: Divide #1 by 25403. → *Response 2:* **\frac{#1}{25403} = -0.2295**
Step 3: Multiply #2 by 100. → *Response 3:* **#2 * 100 = - 22.95**

decompose, **predict**

The **final answer** is **-22.95**

Figure 4: Our EEDP Approach (a.) Instructions, and (b.) Demonstration.

Figure 5: Accuracy of different arithmetic operations across different orders of magnitude.

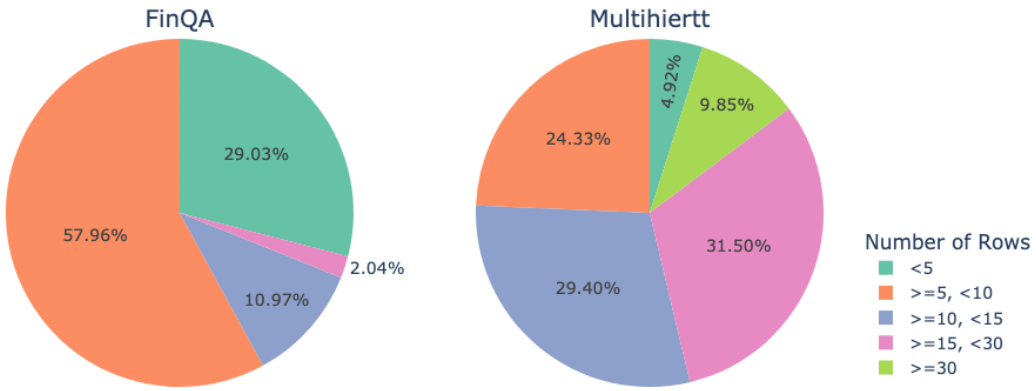
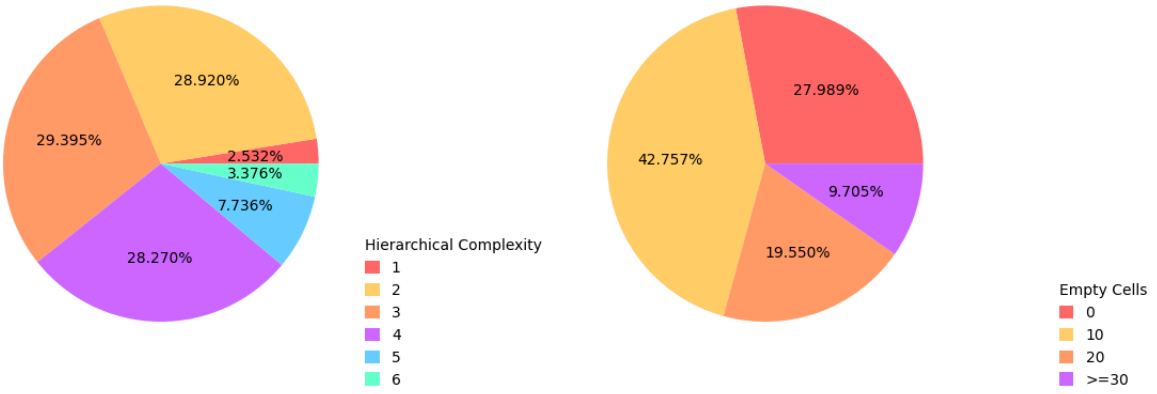


Figure 6: Sample distribution of Multihiertt & FinQA datasets partitioned by number of rows in the supporting table.



(a) Hierarchical Complexity

(b) Empty Cells

Figure 7: Sample distribution of Multihiertt Dataset partitioned by (a) hierarchical complexity of the gold evidence. (b) the percentage of empty cells in the supporting table.

Specifically, for models with a context length exceeding the input length, we standardized k to 4. For instance, we allocated 2 shots for models like LLaMA and MammoTH due to their constrained context length. However, for other models capable of accommodating larger contexts, we increased the number of shots to 4. Additionally, we used a temperature of 0 and top_p = 1 for our experiments.

2. **Models with Mathematical Capabilities:** We prioritized models renowned for their advanced mathematical prowess, such as `MAmmoTH`, alongside state-of-the-art Large Language Models (LLMs) like GPT-4. Our goal was to gain deeper insights into the mathematical reasoning capabilities of cutting-edge models within the context of financial documents.

3. **Better Prompting Approaches:** Rather than focusing solely on model diversity, we concentrated on exploring a variety of prompting methods, particularly those aimed at enhancing mathematical reasoning. We believed this approach would yield more valuable insights into the performance of both LLMs and their associated prompting techniques in real-world financial tasks.

4. **Excluding Underperforming Models:** While we experimented with various models, such as Falcon-7B-Instruct and MPT-7B-Instruct, we found them to underperform significantly compared to models like LLaMA and Mistral. Consequently, we excluded them from detailed analysis. Future experiments with ample computational resources may involve exploring additional open-source models like OLMo, Mixtral, and DBRX, which have been recently released.

A.3 Other Modeling Techniques

Based on our research and the results obtained from our proposed method 'EEDP', we do have several insights that could guide future model development:

1. **Domain-Specific Pre-training:** Our method "EEDP" suggests that LLMs could benefit from pre-training that focuses on extracting domain-specific knowledge. In the context of financial documents, for instance, this could involve training models on a corpus of financial texts, thereby enabling them to better understand and reason about financial concepts and terminology.

2. **Knowledge Elicitation:** The elicitation step in "EEDP" indicates the potential for designing LLMs that can elicit or extract relevant information from a given context more effectively. This could involve developing models that are better at identifying and focusing on key pieces of information in a document, which is crucial for accurately answering questions about the document.

3. **Modular Modeling:** Our research introduce a novel approach to the reasoning process, wherein it's broken down into modular steps. In this methodology, Large Language Models

(LLMs) handle different aspects of a task in distinct stages. This division potentially enhances the overall accuracy and efficiency of the model.

For instance, the model might begin by eliciting domain-specific knowledge, then proceed to extract relevant information from the premise. Following this, it engages in reasoning about this information to answer a question and finally derives the answer, using the output from the preceding reasoning steps as a reference point.

By potentially training individual expert models to handle each specific stage, we could optimize performance for each distinct task. This modular approach allows for specialized processing of each step, thereby improving the overall performance and interpretability of the final output.

4. **Hierarchical Structure Understanding:** Representing the input structure of the table in a better format to the LLM could be beneficial. One can also explore introducing special positional encodings, similar to those used in TAPAS, to serve as row and column IDs for each cell. This approach would differ from traditional positional encodings, which are designed to capture the inherently linear structure of textual data. TThis integration would facilitate the extraction of relvant information from the table correctly, considering its structure more effectively, avoiding information extraction errors. Another idea could be converting the premise containing the complex table and text into a common representation such as a knowledge graph. Furthermore, models specifically tuned to answer to human queries over complex documents in multiple conversational turns (like that in ConvFinQA) should also be considered, as it's a challenge for language model's to backtrack to their decisions that were made previously in the conversation.

A.4 Metadata Annotations Dataset Coverage

Figure 6 displays the dataset distribution for Multihiertt and FinQA based on table length (number of rows). Figure 7 (a) shows how we calculate hierarchical complexity for examples with multiple relevant rows at various hierarchical depths. For the distribution of missing information (empty cell proportions) across datasets, refer to Figure 7 (b). Figure 8 for the distribution of questions by reasoning steps. We define 12 mathematical concept categories, see Table 1) and annotate each question accordingly. The dataset coverage for these categories is shown in Figure 9. Our method for estimating hierarchy depth is shown in Figure 10. Figure 11 shows a example for EEDP strategy with one shot. Figure 11 shows a example for EEDP strategy with one shot.

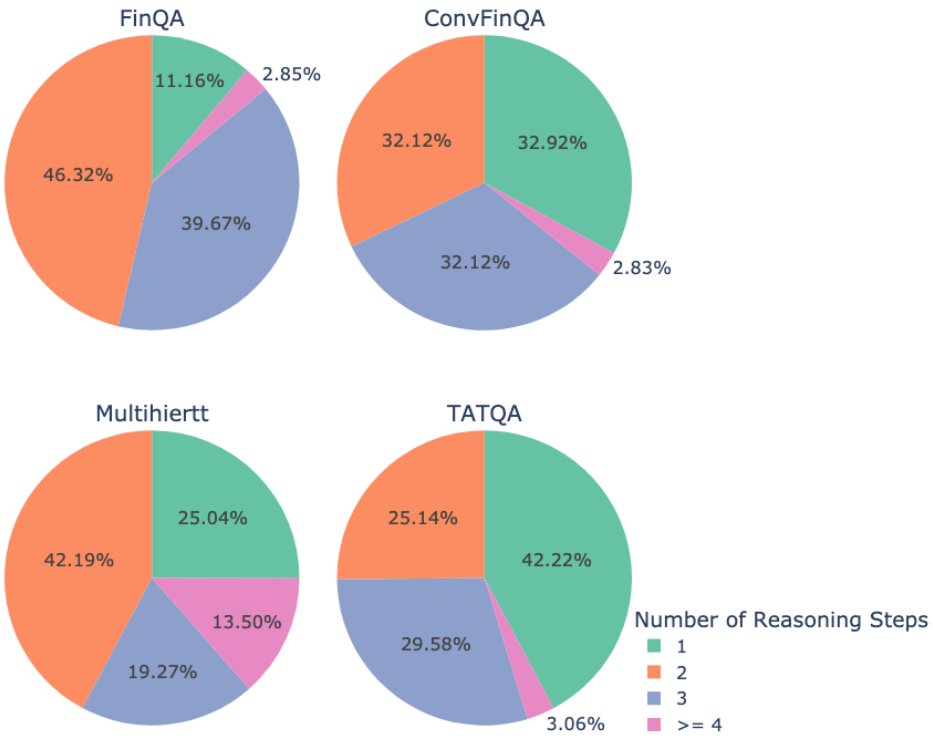


Figure 8: Sample distribution of examples in numerical reasoning on tabular datasets partitioned by the number of reasoning steps involved. Clockwise (from top-left) : FinQA, ConvFinQA, TATQA, Multihiertt.

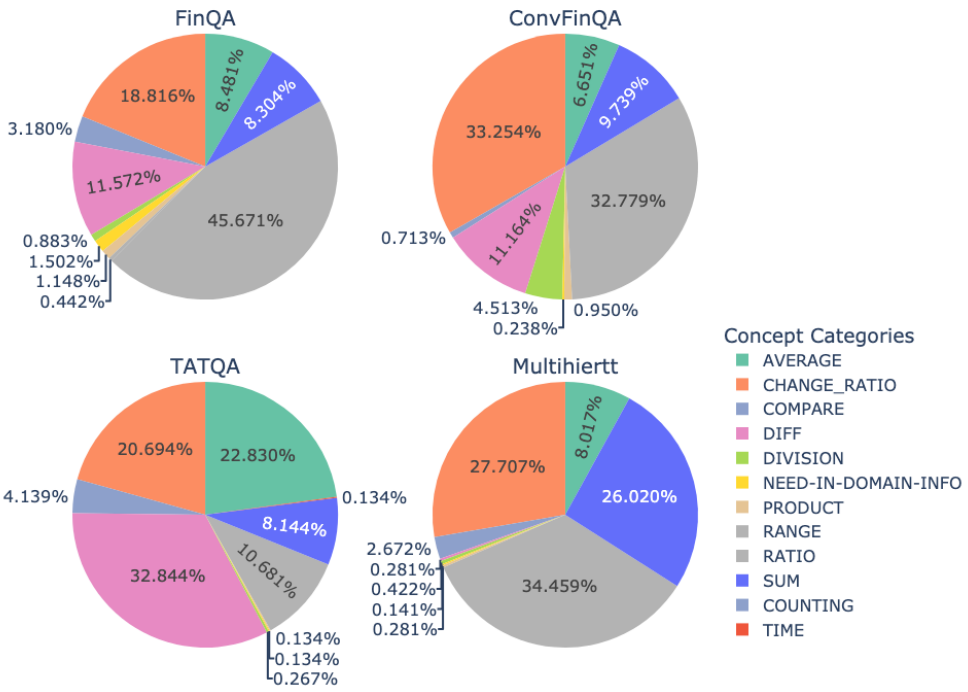


Figure 9: Sample distribution of Numerical & Tabular Reasoning Datasets partitioned by Question Concept Category types. Clockwise (from top-left): FinQA, ConvFinQA, Multihiertt, TATQA.

Years Ended December 31,	2009	2008
(in millions, except percentages)		
Revenues		
Management and financial advice fees	$1,234	$1,339
Distribution fees	$1,733	$1,912
Net investment income	$297	$-43
Other revenues	$85	$80
Total revenues	$3,349	$3,288
Banking and deposit interest expense	$133	$178
Total net revenues	$3,216	$3,110
Expenses		
Distribution expenses	$1,968	$2,121
General and administrative expense	$1,282	$1,138
Total expenses	$3,250	$3,259
Pretax loss	$-34	$-149

Question: What will Distribution fees reach in 2010 if it continues to grow at its current rate? (in millions)?

Gold Evidences:

- Table shows Distribution fees of Years Ended December 31, 2009 (in millions, except percentages) is $1,733 .
- Table shows Distribution fees of Years Ended December 31, 2008 (in millions, except percentages) is $1,912 .

Hierarchical Complexity: max(4, 4) = 4

Figure 10: This illustration demonstrates how the "hierarchical complexity" is determined for each supporting piece of evidence. The hierarchical complexity corresponds to the number of levels or tiers of information structure within the table. The top level (1) encompasses the table itself, which contains information about the results of operations in a firm's Advice & Wealth Management segment. The second level (2) includes columns specifying the years (2009, 2008), creating a substructure within the table. The third level (3) involves the column "(in millions, except percentages)," and the fourth level (4) encompasses the rows under categories like "Revenues," "Distribution fees," "Net investment income," and so on. To locate specific values like "$1733" and "$1912", a retriever module needs to navigate through these four levels.

Instruction: Please carefully analyze the provided information, perform necessary numerical calculations, and provide accurate answers to the given question using the provided data. Take into account the pre-text, table, and post-text when formulating your response.
When performing numerical calculations, ensure you gather the required information and follow a step-by-step approach.

1. Elicit the most relevant domain fact or knowledge that might be useful for you to extract the right operands and forming the right approach to the problem.
2. Translate table data into sentences and identify gold evidence for answering the question.
3. Define a sequence of atomic operations (Add, Subtract, Divide, Multiply, Greater, Min, Max, Exp) which take into account only two operands at a time. Divide a complex task into a sequence of atomic operations defined above.
4. Finally, conclude as: The final answer is <answer>.

Demonstration: "Effective Income Tax Rate": A reconciliation of the United States federal statutory income tax rate to our effective income tax rate is as follows: In 2019 and 2018 we had pre-tax losses of $19,573 and $25,403, respectively, which are available for carry forward to offset future taxable income.

	Year Ended	**Year Ended**
	December 31, 2018	**December 31, 2019**
United States federal statutory rate	21.00%	21.00%
Effective income tax rate	1.99%	-0.01%

Question: What was the 2019 percentage change in pre-tax losses?

Response #:

Domain Knowledge: Pre-tax losses, or operating losses, refer to financial losses that a company incurs before considering the effects of income taxes. To find the 2019 percentage change in pre-tax losses, we need to find the difference between the new and the old value of the pre-tax losses, then divide the obtained difference by the old value and multiply this value by 100.

Gold Evidences:

- The pre-tax losses in 2019 are $19,573.
- The pre-tax losses in 2018 are $25,403.

Steps:

1. Subtract 25403 from 19573.
   - **Response 1:** 19573 - 25403 = -5830
2. Divide #1 by 25403.
   - **Response 2:** $\frac{-5830}{25403} = -0.2295$
3. Multiply #2 by 100.
   - **Response 3:** $-0.2295 \times 100 = -22.95$

The final answer is **-22.95%**.

Figure 11: A Template for our proposed prompting strategy, EEDP with 1-shot demonstration.

Regulatory capital, assets, and risk-based capital ratios for JPMorgan Chase and its significant IDI subsidiaries under Basel III Standardized Transitional and Basel III Advanced Transitional at December 31, 2017, and 2016.

	Basel III Standardized Transitional		**Basel III Advanced Transitional**	
(in millions)	**Dec 31, 2017**	**Dec 31, 2016**	**Dec 31, 2017**	**Dec 31, 2016**
Regulatory capital				
CET1 capital	$184,375	$179,319	$184,375	$179,319
Tier 1 capital(a)	$184,375	$179,341	$184,375	$179,341
Total capital	$195,839	$191,662	$189,419	$184,637
Assets				
Capital ratios(c)				
CET1	13.8%	13.7%	(e)	15.0%
Tier 1(a)	13.8%	13.7%	(e)	15.0%
Total	14.7%	14.6%	(e)	15.4%
Tier 1 leverage(d)	8.7%	8.6%		8.7%

Question: What is the sum of CET1 capital, Tier 1 capital, and Total capital in 2017? (in million)

Response #:

Domain Knowledge: In banking, CET1 capital, Tier 1 capital, and Total capital are used to measure a bank's financial health and stability. CET1 (Common Equity Tier 1) capital consists mainly of common shares and retained earnings. Tier 1 capital is the core capital mainly comprising shareholders' equity and disclosed reserves. Total capital includes both Tier 1 capital and Tier 2 capital, which consists of hybrid capital instruments and subordinated debt.

Gold Evidences:

- Table shows JPMorgan Chase Bank, N.A. Basel III Standardized Transitional CET1 capital on Dec 31, 2017 as $184,375 million.
- Table shows JPMorgan Chase Bank, N.A. Basel III Standardized Transitional Tier 1 capital on Dec 31, 2017 as $184,375 million.
- Table shows JPMorgan Chase Bank, N.A. Basel III Standardized Transitional Total capital on Dec 31, 2017 as $195,839 million.

Steps:

1. Add CET1 capital and Tier 1 capital.
    - **Response 1:** $184,375 + $184,375 = $368,750
2. Add #1 and Total capital.
    - **Response 2:** #1 + $195,839 = $564,589

The final answer is $564,589 million.

Ground-Truth Steps: $184,375 + $184,375 + $195,839 + $184,375 + $184,375 + $189,419 = $1,122,758

Reason: The model missed adding JPMorgan Chase Bank, N.A. Basel III Advanced Transitional CET1 capital ($184,375), Tier 1 ($184,375), and Total Capital ($189,419) in 2017.

Figure 12: Error due to missing evidences

Year	Life/Fin. Ins.	Gen. Ins.	Rtd. Services	Fin. Services	Asset Mngmt	Total
2006						
Fixed Maturities:						
Bonds available for sale	$67,994	$288,018	$1,357	$29,500	—	$386,869
Bonds held to maturity	$21,437	—	—	—	—	$21,437
Bond trading securities	$1	$10,835	—	—	—	$10,836
Equity Securities:						
Common stocks	$4,245	$8,705	—	$226	$80	$13,256
Preferred stocks	$1,884	$650	$5	—	—	$2,539
Other Assets:						
Mortgage and other loans	$17	$21,043	$2,398	$4,884	$76	$28,418
Flight equipment	—	—	$39,875	—	—	$39,875
Total Invested Assets	$116,059	$428,825	$185,404	$68,280	$3,373	$801,941

Question: What is the proportion of Asset Management to the total Mortgage and other loans receivable, net of allowance in 2006?

Response #:

Domain Knowledge: Proportion refers to the relationship between two quantities, where one quantity is considered in relation to the other. To find the proportion of Asset Management to the total Mortgage and other loans receivable, net of allowance in 2006, we need to divide the value of Asset Management by the total value of Mortgage and other loans receivable, net of allowance in 2006.

Gold Evidences:

- Table shows Asset Management of Total invested assets in 2006 is $68,280.
- Table shows Mortgage and other loans receivable, net of allowance (in millions) in 2006 is $28,418.

Steps:

1. Divide Asset Management by the total Mortgage and other loans receivable, net of allowance in 2006.
    - **Response 1:** $\frac{68280}{28418} = 2.4024$

The final answer is 2.4024.

Ground-truth steps: $\frac{4884.0}{28418.0} = 0.172$

Reason: Asset Management in 2006 is taken as $68,280.0 by the model instead of $4,884.0 due to error in the extraction of the ground-truth.

Figure 13: Error due to wrong evidences

	2013	2014	2015	2016	2017
Masco	$138.48	$155.26	$200.79	$227.08	$318.46
S&P 500 Index	$132.04	$149.89	$151.94	$169.82	$206.49
S&P Industrials Index	$140.18	$153.73	$149.83	$177.65	$214.55
S&P Consumer Durables & Apparel Index	$135.84	$148.31	$147.23	$138.82	$164.39

Question: What was the difference in percentage cumulative total shareholder return on Masco common stock versus the S&P 500 Index for the five-year period ended 2017?

Response #:

Domain Knowledge: Cumulative total shareholder return is a measure of the overall performance of a stock over a given period of time. It includes the change in the stock's price as well as any dividends paid to shareholders during that time.

Gold Evidences:

- The cumulative total shareholder return for Masco in 2017 is $318.46.
- The cumulative total shareholder return for the S&P 500 Index in 2017 is $206.49.

Steps:

1. Divide the cumulative total shareholder return for Masco in 2017 (318.46) by the cumulative total shareholder return for the S&P 500 index in 2017 (206.49)
    - **Response 1:** 318.46 / 206.49 = 1.5422
2. Subtract 1 from the quotient obtained in #1
    - **Response 2:** 1.5422 - 1 = 0.5422
3. Multiply the value obtained in #2 with 100 to find the percentage difference
    - **Response 3:** 0.5422 * 100 = 54.22

The final answer is 54.22%.

Ground-truth steps: [((318.46 - 100.0)/100.0)*100.0 - ((206.49 - 100.0)/100.0)*100.0] = 111.97

Reason: Wrong formula for return on investment

Figure 14: Example of an error caused due to insufficient domain knowledge

	Accr. Yield (Beginning)	Carrying Amt. (Beginning)	Allowance (Beginning)
Beginning Balance	$27	$920	$95
Purchases	$1	$130	$0
Disposals/Payments Received	$-11	$-594	$0
Accretion	$-44	$44	$0
Builds (Reductions) to the Allowance	$0	$0	$-18
Increase to Expected Cash Flows	$-2	$19	$0
FX/Other	$17	$-50	$0
Balance at December 31, 2010	$116	$469	$77

Question: What percentage did this total represent in relation to the carrying amount?

Response #:

Domain Knowledge: To find the percentage of a value (A) in relation to another value (B), divide the first value (A) by the second value (B) and multiply it by 100. To find the combined total, add the two values.

Gold Evidences:

- The carrying amount of the company's purchased distressed loan portfolio at December 31, 2010, was $392 million.
- The net allowance was $77 million as of December 31, 2010.

Steps:

1. Find the combined total of net allowance and the carrying amount.
    - **Response 1:** 392 + 77 = 469
2. Calculate the percentage of net allowance in relation to the carrying amount.
    - **Response 2:** $\frac{77}{392} \times 100 = 19.64$
3. Calculate the percentage of the combined total in relation to the carrying amount.
    - **Response 3:** $\left(\frac{469}{392}\right) \times 100 = 119.64$

The final answer is 119.64%.

Ground-truth steps: [(77/(392 + 77))*100] = 16.42%

Reason: By the phrase in relation to the question expected a ratio of $quantity_1/quantity_2$ but the model returned $quantity_2/quantity_1$

Figure 15: Example of an error caused due to question misinterpretation

	2001	2000
Gain on Sales of Depreciable Properties	$45428	$52067
Gain on Land Sales	$5080	$9165
Impairment Adjustment	$-4800 (4800)	$-540 (540)
Total	$45708	$60692

Question: What is the percent change in general and administrative expense from 2000 to 2001?

Response #:

Domain Knowledge: To find the percent change between two values, we need to find the difference between the new and old values, divide the difference by the old value, and multiply the result by 100.

Gold Evidences:

- General and administrative expense decreased from $21.1 million in 2000 to $15.6 million for the year ended December 31, 2001.

Steps:

1. Calculate the change in general and administrative expense by subtracting the 2001 value ($15.6 million) from the 2000 value ($21.1 million).
    - **Response 1:** 21.1 - 15.6 = 5.5
2. Divide the difference obtained in Step 1 by the general and administrative expense in 2000 ($21.1 million).
    - **Response 2:** $\frac{5.5}{21.1} = 0.2607$
3. Multiply the result obtained in Step 2 by 100 to get the percent change.
    - **Response 3:** $0.2607 \times 100 = 26.07$

The final answer is 26.07%.

Ground-truth steps: [((15.6 - 21.1)/21.31)*100] = -26.07%

Reason: Incorrect Instantiation of variables resulting in a value with a flipped sign

Figure 16: Example of an error caused due to incorrect instantiation of variables

	2013	2014	2015	2016	2017
CME Group Inc.	$164.01	$194.06	$208.95	$279.85	$370.32
S&P 500	$132.39	$150.51	$152.59	$170.84	$208.14
Peer Group	$176.61	$187.48	$219.99	$249.31	$323.23

Question: What is the annualized return for s&p 500 from 2012 to 2017?

Response #:

Domain Knowledge: The annualized return refers to the average rate of return over a given period for an investment, which has been adjusted for compounding. To find the annualized return for the S&P 500 from 2012 to 2017, we need to use the formula:

$$\text{Annualized return} = \left(\left(\frac{\text{Ending value}}{\text{Starting value}}\right)^{\frac{1}{\text{Number of years}}} - 1\right)$$

Gold Evidences:

- Investment was $100 on December 31, 2012;Value of S&P 500 investment on December 31, 2017 is $208.14.

Steps:

1. Divide the ending value (208.14) by the starting value (100)
   - **Response 1:** $\frac{208.14}{100} = 2.0814$
2. Calculate the number of years elapsed (2017 - 2012)
   - **Response 2:** 2017 - 2012 = 5
3. Raise the value obtained in #1 to the power of (1/number of years)
   - **Response 3:** $(2.0814)^{\frac{1}{5}} = 1.1606$
4. Subtract 1 from the value obtained in #3
   - **Response 4:** 1.1606 - 1 = 0.1606
5. Multiply the value obtained in #4 by 100 to get the annualized return in percentage
   - **Response 5:** 0.1606 * 100 = 16.06

The final answer is 16.06%.

Ground-truth steps: [$(208.14/100)^(1/(2017-2012))$ - 1] = 0.1577 * 100 = 15.77

Reason: Inaccurate calculation

Figure 17: Example of an error caused due to imprecise calculations