XBRL-Graph Agent: Financial Numerical Question Answering over XBRL Knowledge Graphs

Anonymous ACL submission

Abstract

Financial numerical question answering requires retrieving precise numerical evidence from financial reports and performing domain-specific reasoning over it. Existing approaches primarily rely on report text and tables, while structured regulatory disclosures remain under-explored. EXtensible Business Reporting Language (XBRL) is a machine-readable standard that encodes financial disclosures with standardized tags, enabling structured access to 10-K and 10-Q filings. We propose XBRL-Graph Agent (XGA) for financial numerical question answering with two core modules: the XBRL Fact Tool, which retrieves question-relevant numerical evidence from XBRL filings, and the Formula Graph Tool, which supports reasoning over structured financial formulas and domain knowledge. We also introduce XBRL-FinNQA, a new dataset covering 12 key financial metrics for each company, designed to evaluate whether models can retrieve report-grounded numerical evidence from XBRL filings and reason over it. Experiments and ablation studies show that our method outperforms strong baselines across most evaluated LLMs. The results demonstrate that combining XBRL-based evidence retrieval with structured financial knowledge improves LLMs' numerical reasoning and highlights the value of XBRL-structured disclosures for financial numerical question answering.

1 Introduction

Large Language Models (LLMs) have shown strong performance in question answering (QA), a key benchmark for evaluating evidence identification and multi-step reasoning capabilities (Yue et al., 2025; Long et al., 2025). Financial QA is a particularly important application, where models must analyze financial reports, identify relevant evidence, and perform numerical reasoning for decision-making and risk assessment (Nie et al., 2024; Chen et al., 2021). Yet financial questions often require domain-specific knowledge and reasoning over heterogeneous inputs, including narrative disclosures and structured tables (Fatemi and Hu, 2024). This makes fine-grained numerical evidence retrieval a central bottleneck.

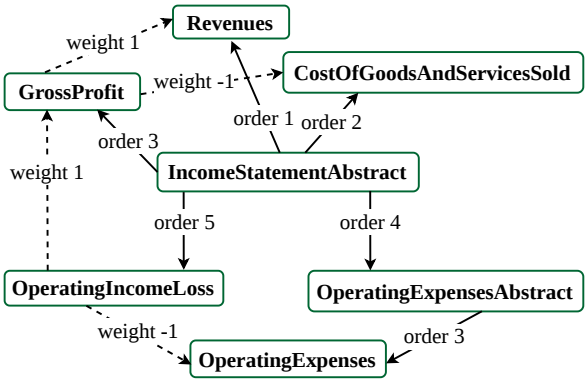

Figure 1: An example of structural relations encoded in XBRL. Solid arrows represent presentation (parent-child) relations among financial concepts. Sibling concepts are displayed according to their order attribute. Dashed arrows represent calculation (summation-item) relations, where weights of +1 and -1 denote positive and negative contributions, respectively. Additional details are discussed in Section 3.

Recent work has explored Retrieval-Augmented Generation (RAG) (Lewis et al., 2021) to ground LLMs on financial documents (Wang et al., 2025b; George et al., 2025; Sarmah et al., 2024). However, existing methods still struggle to retrieve precise numerical evidence and often lack the financial knowledge needed for reliable numerical reasoning.

This observation motivates our shift from plain text and tables to XBRL as a structured source for numerical evidence retrieval. XBRL is a global standard for business information exchange, where reported financial facts are tagged with taxonomy concepts and concept relations are organized through presentation, calculation, definition, and label linkbases, as illustrated in Figure 1. These

relations naturally support an XBRL-based knowledge graph (KG), which links facts to their tagged concepts and organizes concepts through structural and semantic relations. This representation enables more precise numerical evidence retrieval for financial QA. However, querying and reasoning over XBRL-based KGs in financial QA remains largely underexplored in the literature (Nararatwong et al., 2024).

To address these limitations, we propose XBRL-Graph Agent (XGA), an agentic RAG framework for financial numerical QA under the ReAct (Yao et al., 2023) paradigm. XGA integrates two tools: a *Formula Graph Tool*, which retrieves calculation rules for queried financial metrics from a graph of metrics, financial items, and arithmetic operators, and an *XBRL Fact Tool*, which retrieves question-relevant numerical facts from XBRL filings using explicit structural relations. The agent decomposes the original query, retrieves the required financial knowledge and numerical evidence through these tools, and synthesizes the final answer.

To evaluate numerical evidence retrieval and reasoning over XBRL filings, we construct XBRL-FinNQA, a benchmark dataset extending a subset of FinanceBench (Islam et al., 2023). XBRL-FinNQA contains 1,344 QA pairs covering 12 financial metrics commonly used in financial analysis, constructed from 10-K reports of 30 companies from 2020 to 2024. Each question is grounded in XBRL filings, making the dataset suitable for evaluating report-grounded numerical evidence retrieval and financial numerical reasoning. Through extensive experiments and ablations, XGA demonstrates substantial improvements over baseline approaches. Our contributions are summarized as follows:

- We propose XGA, a novel agentic RAG framework for financial numerical question answering that integrates financial knowledge retrieval with XBRL-based numerical evidence retrieval.
- We introduce XBRL-FinNQA, a new benchmark dataset comprising 1,344 QA pairs over 12 key financial metrics, constructed from 10-K reports of 30 companies from 2020 to 2024.
- We show that XGA substantially outperforms baseline methods across most evaluated LLMs, demonstrating the complementary benefits of financial knowledge grounding and XBRL-based numerical evidence retrieval.

2 Related Works

2.1 Financial Numerical QA

Financial numerical QA requires models to identify relevant items and perform numerical reasoning over heterogeneous data. Prior work has examined LLMs' ability to retrieve numerical information from financial tables for question answering (Balsiger et al., 2024). However, LLMs remain prone to numerical hallucinations and misinterpretations of financial terms (Balsiger et al., 2024). These limitations show that parametric knowledge alone is insufficient for faithful financial numerical QA.

RAG has therefore become a common strategy for grounding financial numerical QA in external evidence. Early systems retrieve relevant report chunks from vector databases to support answer generation (Iaroshev et al., 2024), while later work incorporates query expansion, sparse-dense hybrid retrieval, and metadata-aware search to distinguish highly similar financial items and values (Wang et al., 2025b). However, their static retrieval-generation pipelines limit their ability to handle complex questions requiring iterative evidence collection or multi-step numerical reasoning.

Recent studies have explored agentic RAG, where systems iteratively collect evidence for financial numerical QA (Singh et al., 2026). For example, George et al. (2025) combine an enhanced HyDE strategy with BM25 for multi-step retrieval, enabling better discrimination of highly similar numerical contexts and improving QA accuracy. Similarly, Wang et al. (2026) employ multiple collaborating agents to support numerical reasoning across tables and long documents.

The work most related to ours is XBRL-Agent (Han et al., 2024), which retrieves values from XBRL filings but treats them as generic documents rather than exploiting their inherent graph structure. This limits its ability to use structural relations among concepts for precise numerical evidence retrieval. Our work addresses this gap by leveraging XBRL structure for more accurate and reliable evidence grounding in financial numerical QA.

Financial Numerical QA Datasets Financial numerical QA datasets evaluate models' ability to identify numerical evidence and reason over financial reports. Early benchmarks such as FinQA (Chen et al., 2021) and TAT-QA (Zhu et al., 2021) focus on text-table numerical QA, requiring models to combine heterogeneous evidence to derive

answers. Subsequent benchmarks such as ConvFinQA (Chen et al., 2022) and MultiHiertt (Zhao et al., 2022) extend this setting to multi-turn context tracking, long-document evidence integration, and hierarchical numerical reasoning.

With the rise of LLMs, benchmarks have shifted toward more diverse tasks. FinanceBench (Islam et al., 2023) evaluates open-book QA over public-company filings, requiring models to locate supporting evidence and answer questions involving financial values and metrics. FinDER (Choi et al., 2025) extends this direction to RAG-based financial QA with query-evidence-answer triplets, enabling evaluation of both evidence retrieval and answer generation over large financial corpora. In the context of XBRL, FinLoRA (Wang et al., 2025a) introduces an XBRL QA dataset covering tag extraction, value extraction, formula construction, and formula calculation to evaluate models' understanding of structured financial information.

Despite these advances, existing datasets lack a benchmark for evaluating RAG-based numerical evidence retrieval and reasoning over XBRL filings. We fill this gap with XBRL-FinNQA, a benchmark in which the information required to answer each question can be grounded in XBRL filings.

2.2 Financial Knowledge Graph RAG

Knowledge graphs are commonly represented as structured graphs of entities and relations, often in the form of triples (Hogan et al., 2021). Recent work has combined KGs with RAG to improve retrieval and reasoning in financial analysis and QA (Li et al., 2025; Shukla et al., 2025; Shah et al., 2024; Chen et al., 2024b; Cheng et al., 2026). However, existing financial KGs primarily model companies, events, news, report entities, and industry relations (Elhammadi et al., 2020; Arun et al., 2025; Zhou et al., 2025), leaving graph-based representations of financial metric formulas underexplored. In particular, few studies represent financial metrics, items, and arithmetic operators in a unified graph to capture formula composition.

Our work incorporates two complementary forms of structured financial knowledge. First, XBRL provides explicit semantic relations among financial concepts and can be naturally represented as a KG. Second, we represent financial metric formulas as formula graphs, where metrics, items, and arithmetic operators are modeled as nodes. We investigate whether using these resources as external tools in RAG can provide LLMs with domain-specific financial knowledge and improve financial numerical QA.

3 Preliminaries

XBRL is an XML-based standard for tagging business reports, enabling financial statements to be parsed and analyzed by computers. An XBRL filing typically consists of a schema, an instance document, and a set of linkbases (Fourny, 2020). The schema defines the reporting elements, each of which is referred to as a *concept* (XBRL International, 2013). Concepts can be either *reportable* (i.e., non-abstract), meaning that they may carry reported values, such as Assets or Revenues, or *non-reportable* (i.e., abstract), meaning that they are used only to group related concepts, such as IncomeStatementAbstract or AssetsAbstract (European Securities and Markets Authority, 2025). The instance document contains reported facts, which refer to values tagged with XBRL concepts and associated contexts, while the linkbases specify relationships among concepts (Hoitash et al., 2021).

Among the linkbases, the presentation and calculation linkbases are particularly important for capturing the structural and arithmetic organization of financial statements. The presentation linkbase uses `parent-child` arcs to define hierarchical relations and the display order of concepts in a report. Parent nodes are often abstract concepts, whereas child nodes may be either lower-level abstract concepts or reportable concepts associated with actual values. The `order` attribute further specifies the sequence among sibling concepts. In contrast, the calculation linkbase encodes aggregation relations through `summation-item` arcs, where the `weight` attribute indicates whether each child concept contributes additively or subtractively to its parent, as illustrated in Figure 1. More generally, the relations defined in XBRL linkbases provide a structured representation of financial concepts and can therefore be used to construct a knowledge graph.

Notation We use a knowledge graph representation $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E})$ for XBRL structures, where $\mathcal{V}$ is the set of financial concepts, $\mathcal{R}$ is the set of relation types, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ is the set of triples (c_h, r, c_t), where $c_h, c_t \in \mathcal{V}$ are the head and tail concepts and $r \in \mathcal{R}$ is the relation type. For simplicity, we divide concepts into two groups: $\mathcal{V}^a$, the set of non-reportable concepts, and $\mathcal{V}^r$, the set of reportable concepts that may carry reported values. This grouping is used for ease of presentation.

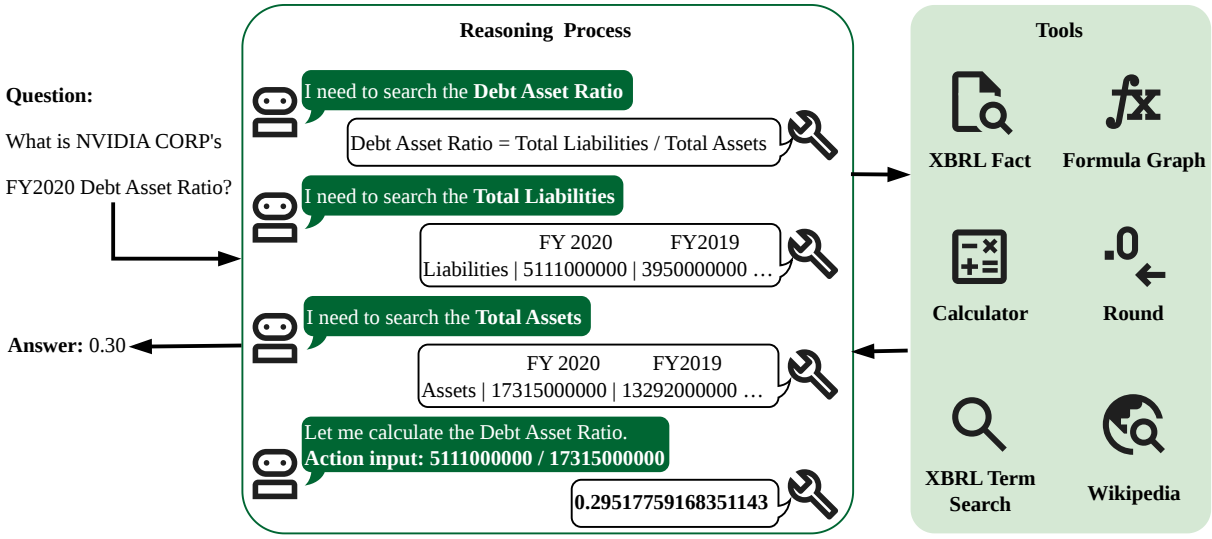

Figure 2: System overview of the XGA. Given a question, the LLM first uses the *Formula Graph Tool* to identify the financial metric formulas required for answering it, and then uses the *XBRL Fact Tool* to retrieve the necessary information based on the observations. After gathering the required facts, it uses a Calculator to compute the final answer. The Round Tool rounds numerical results to two decimal places, the XBRL Term Search Tool retrieves XBRL term definitions, and Wikipedia provides external knowledge.

4 XBRL-Graph Agent (XGA)

Figure 2 provides an overview of XGA. Building on XBRL-Agent (Han et al., 2024), XGA introduces two tools to address key limitations of LLMs in financial report analysis: inaccurate numerical evidence retrieval and insufficient financial domain knowledge. Specifically, the LLM can invoke: (1) the *XBRL Fact Tool*, which retrieves relevant financial values from XBRL filings; and (2) the *Formula Graph Tool*, which represents financial metric formulas as graphs and returns the associated financial items and calculation structure for the metric in the query. We describe these components below.

4.1 *XBRL Fact Tool*

XBRL Knowledge Graph Construction We use the presentation linkbase as the primary source for graph construction because it provides a comprehensive and user-facing organization of financial items. Its hierarchy reflects how concepts are arranged in financial statements, allowing the resulting graph to capture contextual relations that are directly relevant to financial QA. In contrast, the calculation linkbase mainly represents arithmetic aggregation relations. Although these relations are useful for numerical consistency, they are less aligned with our goal of modeling statement-level context and may vary across companies.

For each company-year, we extract financial concepts $\mathcal{V}$ and parent-child triples $\mathcal{E}_p \subseteq \mathcal{V} \times \mathcal{R}_p \times \mathcal{V}$ from the presentation linkbase, where each triple $(c_h, r_{\text{pc}}, c_t)$ uses $r_{\text{pc}} \in \mathcal{R}_p$ to denote the parent-child relation. We then use NetworkX[1] to construct the corresponding knowledge graph $\mathcal{G}_p = (\mathcal{V}, \mathcal{R}_p, \mathcal{E}_p)$ for downstream retrieval.

Two-Stage Retrieval In practice, the financial concepts used in XBRL filings often vary across companies. For example, some companies use `Revenues` in the income statement, while others use a more fine-grained concept such as `RevenueFromContractWithCustomerExcludingAssessedTax`. In addition, there may be a semantic gap between the user query and the target concept. For instance, a query for *Total Debt* may actually correspond to the *Total Liabilities*. These issues make simple lexical matching unreliable for XBRL concept retrieval.

We address this with a two-stage retrieval framework. First, we map an open-ended query to candidate XBRL concepts. Second, we use these candidates to retrieve corresponding facts, including years and values, from the XBRL instance file. We describe both stages below.

Stage 1: Concept Retrieval

Anchor Selection We first select query-relevant anchors from the reportable concept set $\mathcal{V}^r$ via semantic matching. We use a pretrained text encoder

[1] https://networkx.org/en/

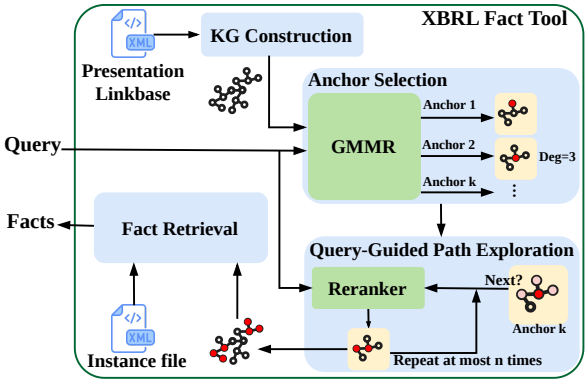

Figure 3: Overview Architecture of *XBRL Fact Tool*

[2] to encode the query q and each reportable concept name $c^r \in \mathcal{V}^r$, and compute their cosine similarity.

Selecting only the most similar concept may fail because $\mathcal{G}_p$ is not fully connected, and the target concept may lie in another subgraph. To improve coverage, we propose Graph-aware Maximal Marginal Relevance (GMMR), a graph-aware variant of Maximal Marginal Relevance (Carbonell and Goldstein, 1998), to select k anchors, with $k = 2$ in this work. GMMR initializes the anchor set with the most query-similar concept and then greedily selects the remaining $k-1$ anchors by balancing query relevance, graph connectivity, and diversity with respect to the selected anchors. Specifically, it incorporates each candidate anchor's in-degree, i.e., the number of non-reportable concepts pointing to it. Anchors with higher in-degree are connected to more non-reportable concepts, offering more expansion paths toward concepts associated with other sections during subsequent graph expansion.

Formally, GMMR initializes the anchor set $\mathcal{K}$ with the most query-similar concept:

$$c^r_{\text{init}} = \arg\max_{c^r_i \in \mathcal{V}^r} \text{sim}(q, c^r_i), \quad \mathcal{K} = \{c^r_{\text{init}}\}.$$

For each remaining candidate concept $c^r_i \in \mathcal{V}^r \setminus \mathcal{K}$, we define its GMMR score as:

$$\begin{aligned}\text{GMMR}(c^r_i) = \lambda\ &\text{sim}(q, c^r_i)\ \deg_{in}(c^r_i) \\ &- (1-\lambda) \max_{c^r_j \in \mathcal{K}} \text{sim}(c^r_i, c^r_j).\end{aligned}$$

where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity, and $\deg_{in}(\cdot)$ denotes the in-degree of a concept in $\mathcal{G}_p$. GMMR then adds the highest-scoring candidate to $\mathcal{K}$, recomputes the scores with the updated $\mathcal{K}$, and repeats this process until $|\mathcal{K}| = k$.

[2]https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Query-Guided Path Exploration After anchor selection, we perform query-guided path exploration on $\mathcal{G}_p$ from each anchor concept $c^r_k \in \mathcal{K}$.

For each anchor c^r_k, we add it to the path-expanded concept set $\mathcal{C}^{\text{path}}$, initialize the current node as $u_0 = c^r_k$ and iteratively explore the graph for at most n steps.

At step t, given the current node u_t, we collect neighboring non-reportable concepts as expansion nodes:

$$\begin{aligned}\mathcal{A}_t = \{c^a \in \mathcal{V}^a \mid &(c^a, r_{\text{pc}}, u_t) \in \mathcal{E}_p \\ &\text{or } (u_t, r_{\text{pc}}, c^a) \in \mathcal{E}_p\}.\end{aligned} \tag{1}$$

We consider predecessors and successors of the current node but keep only non-reportable concepts for expansion. If $\mathcal{A}_t$ is empty, the exploration stops.

For each expansion node $c^a_i \in \mathcal{A}_t$, we collect its one-hop children,

$$\text{Succ}(c^a_i) = \{c \in \mathcal{V} \mid (c^a_i, r_{\text{pc}}, c) \in \mathcal{E}_p\},$$

and convert c^a_i and its successors into a natural-language description using the template "the children of c^a_i are $\text{Succ}(c^a_i)$." This relation-to-text step represents each expansion node's structured neighborhood as text for query matching.

We then use a cross-encoder reranker[3] to guide the next step of exploration. Each description is paired with q and scored by the reranker. The reranker ranks expansion nodes by query relevance, and we select the top-ranked node as the next node:

$$u_{t+1} = \arg\max_{c^a_i \in \mathcal{A}_t} \text{Score}(q, \text{Text}(c^a_i, \text{Succ}(c^a_i))).$$

Here, $\text{Text}(\cdot,\cdot)$ and $\text{Score}(\cdot,\cdot)$ denote the relation-to-text and reranker scoring functions, respectively. After selecting u_{t+1}, we add its descendant concepts to $\mathcal{C}^{\text{path}}$ and continue the exploration from u_{t+1} using Eq. (1). This process runs for at most n steps, with $n = 3$ in this work.

Finally, we filter $\mathcal{C}^{\text{path}}$ by query-concept similarity, removing concepts below a predefined threshold to obtain the final candidate set $\mathcal{C}^{\text{cand}}$.

Stage 2: Fact Retrieval We retrieve facts for the candidate concepts from the XBRL instance file. For each candidate concept $c \in \mathcal{V}^r \cap \mathcal{C}^{\text{cand}}$, we use Arelle[4] to parse the instance file and extract associated facts, including the reporting period and reported value. The extracted facts are returned to the LLM as observations from the *XBRL Fact Tool*. Figure 3 shows the *XBRL Fact Tool* overview.

[3]https://huggingface.co/BAAI/bge-reranker-v2-m3
[4]https://github.com/Arelle/Arelle

4.2 *Formula Graph Tool*

We curate financial metric formulas, allowing operands that are either explicit financial items or values computed from them. We parse each formula with Python's Abstract Syntax Tree (AST)[5] and convert it into a directed acyclic graph. Financial metrics, financial items, and arithmetic operators are represented as nodes. Edges encode operand roles with three relation types: `left_operand`, `right_operand`, and `is_calculated_by`. We recursively traverse the AST to construct each formula graph. An example is shown in Figure A.4 in Appendix.

This representation allows the same financial item to appear in multiple formulas, connecting metrics through shared components. While this structure may offer limited benefits over dense-vector RAG for simple formula lookup, it supports dependency-aware analysis. For example, when a financial item changes, the system can identify dependent formulas and affected metrics. Such dependencies are difficult to capture with text-similarity-based retrieval alone but are explicit in our graph representation. Therefore, this representation supports the current retrieval task and provides a foundation for future financial reasoning applications.

In this work, we use the formula graph to retrieve and reconstruct financial metric formulas. We compute semantic similarity between the query and financial metric nodes, select the most relevant metric, and reconstruct its subgraph as a formula. The reconstructed formula is returned to the LLM as the observation of the *Formula Graph Tool*.

5 XBRL-FinNQA

XBRL-FinNQA is a sub-task extended from FinanceBench (Islam et al., 2023), designed to evaluate numerical reasoning over structured XBRL financial statements. It follows a similar column structure to FinanceBench while introducing an additional xbrl field for structured financial statement data. The dataset contains 1,344 question-answer pairs constructed from 139 10-K reports of 30 companies spanning fiscal years 2020–2024, collected using EDGAR-CRAWLER. [6] The dataset covers four major categories of financial analysis, namely liquidity, efficiency, solvency, and profitability. From these categories, we select 12 key financial metrics and construct questions using metric-specific template, such as "What is [company]'s [financial metric] in FY[year]?" For each question, we identify the relevant financial statements and target XBRL concepts, and compute the answer from the corresponding XBRL facts. Questions for which the ground-truth answer cannot be obtained from the filing are excluded. Each resulting entry includes a question, answer, company name, fiscal year, context, xbrl field, relevant financial statements, and target XBRL concepts.

A key feature of XBRL-FinNQA is its `xbrl` field, which contains structured financial statements extracted from XBRL filings. This field is introduced because original XBRL filings are often lengthy, while the hierarchical relations among financial concepts and their corresponding facts are distributed across different XBRL files. Instead of treating the full filings as input, we therefore extract question-relevant facts and hierarchical structures to form a compact, structured representation. Details of this extraction process are provided in Appendix A.1.

We use XBRL-FinNQA as the primary dataset for our experiments. The context field, derived from flattened financial statement content, contains an average of 800 tokens, while the reconstructed xbrl field averages 23,000 tokens. In contrast, the original full XBRL instance files average approximately 1,375,000 tokens. This substantial reduction highlights the importance of our transformation process for making XBRL data suitable for LLM-based numerical reasoning. A representative example is provided in Appendix A.2.

6 Experiments

6.1 Experimental Setup

Models We conduct a comprehensive evaluation with seven state-of-the-art LLMs: Llama-3.3-70B-Instruct (Grattafiori et al., 2024), GPT-OSS-120B (OpenAI et al., 2025), GPT-3.5-Turbo (OpenAI, 2023), GPT-4o (OpenAI et al., 2024), Claude-4.6-Sonnet (Anthropic, 2026), Qwen3-235B-A22B-Instruct-2507 (Yang et al., 2025), and DeepSeek-V3.1 (DeepSeek-AI, 2024). These models cover both open-weight and proprietary models, and include reasoning models as well as frontier LLMs. We run Llama-3.3-70B-Instruct locally using Ollama[7], access GPT-3.5-Turbo and

[5] https://docs.python.org/zh-tw/3/library/ast.html

[6] https://github.com/lefterisloukas/edgar-crawler

[7] https://ollama.com/

Model / Method	CoT + Flatten Text	*CoT + XBRL	XBRL-Agent	FinSage	XGA (Ours)
Llama-3.3-70B	0.71	0.69	0.03	0.21	**0.90**
GPT-OSS-120B	**0.79**	0.70	0.04	0.22	0.63
GPT-3.5-Turbo	0.59	0.53	0.03	0.11	**0.73**
GPT-4o	0.85	0.88	0.03	0.09	**0.93**
Claude-Sonnet-4.6	0.79	0.84	0.07	0.26	**0.99**
Qwen3	0.82	0.82	0.06	0.11	**0.97**
DeepSeek-V3.1	0.80	0.82	0.14	0.19	**0.97**

Table 1: Comparison of all methods across the evaluated models on the XBRL-FinNQA dataset. The best result for each model is shown in bold. *For the CoT + XBRL setting, the XBRL context is extremely long, making experiments with proprietary model APIs prohibitively expensive. We therefore conduct this setting on a randomly sampled 25% subset of the dataset.

GPT-4o through the OpenAI API, and access GPT-OSS-120B, DeepSeek-V3.1, Qwen3-235B-A22B-Instruct-2507, and Claude-4.6-Sonnet through OpenRouter[8]. To ensure a fair comparison, we use a unified temperature setting of 0.7 for all models. We use fixed models across all experiments: GPT-3.5-Turbo for the calculator tool, all-mpnet-base-v2 [9] for text encoding, and BAAI/bge-reranker-v2-m3 (Chen et al., 2024a) for cross-encoder reranking.

Evaluation We use GPT-4o-mini (OpenAI, 2024) to extract predicted answers and convert numeric strings into numerical values. Models are instructed to round answers to two decimal places during reasoning, and ground-truth values are rounded likewise. We compare predicted and ground-truth values using Python's `isclose` function, following Phogat et al. (2023).

Model performance is measured by overall accuracy, defined as the percentage of predictions that are the same as the ground-truth answers.

Baselines To evaluate the effectiveness of our proposed method, we compare it against four baselines. (1) **CoT + Flattened Text** provides the LLM with the question and the relevant financial table(s) serialized as flattened text, allowing the model to answer directly without retrieval. (2) **CoT + XBRL** provides the LLM with the question and the corresponding xbrl field as context for direct question answering. (3) **XBRL-Agent** (Han et al., 2024) is included as a representative XBRL-based financial agent that combines retrieval over chunked XBRL filings with a calculator to improve financial reasoning and numerical computation. (4) **FinSage** (Wang et al., 2025b) is a strong RAG-based baseline that improves retrieval quality through a multi-path retrieval pipeline combining sparse, dense, metadata, and HyDE retrievers.

[8] https://openrouter.ai/

[9] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

6.2 Results and Analysis

Table 1 reports the results of our method and baselines. Overall, our method performs best on most evaluated models. Claude-Sonnet-4.6, Qwen3, and DeepSeek-V3.1 achieve particularly strong results under our framework. Even for the less capable open-weight model Llama-3.3-70B, our method improves accuracy by 19% over CoT + Flattened Text. This demonstrates that our method provides a substantial benefit across different model families. The main exception is GPT-OSS-120B, where the CoT baseline performs better. One possible reason is GPT-OSS-120B's weaker instruction following, which can cause incorrect tool use and limit the benefit of XGA.

The two CoT baselines show that LLMs can perform reasonably well when given sufficient relevant context. Moreover, the comparable performance of CoT + Flattened Text and CoT + XBRL suggests that simply replacing flattened tables with XBRL representations as input context does not lead to a clear improvement. Thus, merely exposing LLMs to XBRL's XML-based structure is insufficient to exploit its value for financial QA. A similar limitation appears in XBRL-Agent, which retrieves evidence from chunked XBRL filings. Its results suggest that chunk-based retrieval over XBRL filings may not always collect all the evidence required to answer a question, leaving room for improvement in overall performance. In contrast, FinSage performs noticeably better with a stronger retrieval strategy, highlighting the importance of retrieval quality in financial numerical QA.

XGA goes beyond treating XBRL as textual

Model / Method	w/o *Formula Graph Tool*	w/o *XBRL Fact Tool*	XGA (Ours)
Llama-3.3-70B	0.79	0.02	**0.90**
GPT-OSS-120B	0.56	0.03	**0.63**
GPT-3.5-Turbo	0.54	0.02	**0.73**
GPT-4o	0.89	0.04	**0.93**
Claude-Sonnet-4.6	0.85	0.24	**0.99**
Qwen3	0.86	0.05	**0.97**
DeepSeek-V3.1	0.87	0.15	**0.97**

Table 2: Ablation results of XGA and its variants with individual components removed.

input or retrievable chunks. It uses the *XBRL Fact Tool* to exploit the structure of XBRL by retrieving evidence based on its structural properties. Through an agentic RAG framework, XGA further enables the LLM to assess evidence sufficiency and continue retrieving information when necessary. Meanwhile, the *Formula Graph Tool* supplements the model with financial domain knowledge. Together, these tools support more complete evidence collection and more effective financial numerical reasoning, resulting in the strongest overall performance across most evaluated models.

Ablations We conduct ablation studies, with results shown in Table 2. Removing the *Formula Graph Tool* degrades performance across all LLMs. Error analysis suggests that without formula guidance, models often rely on internal financial knowledge or infer computations from metric names, leading to incorrect variable selection. For example, in debt-asset ratio questions, most models infer the formula from the metric name and incorrectly use *Total Debt*, resulting in erroneous calculations. Removing the *XBRL Fact Tool* leads to a larger drop. Although relevant formulas are available, the model lacks a reliable mechanism for retrieving filing-specific evidence. As a result, models often hallucinate numerical values or rely on parametric knowledge rather than actual XBRL facts. This pattern is especially visible for Claude-Sonnet-4.6 and DeepSeek-V3.1, whose relatively stronger performance without this tool may reflect greater exposure to related financial information during training. Overall, the results show that accurate financial numerical reasoning requires both formula guidance and evidence retrieval, with the best performance achieved when both tools are used together.

7 Conclusion and Future Work

In this work, We propose XGA, an agentic framework for financial numerical QA with two key components: the *XBRL Fact Tool* and the *Formula Graph Tool*. We also introduce XBRL-FinNQA, a benchmark where the evidence required to answer each question can be grounded in XBRL filings.

Experiments with seven state-of-the-art LLMs show that XGA consistently outperforms strong baselines on most models. The results show that simply providing XBRL content as input context does not yield clear gains. Instead, XBRL is most useful when its structure is accessed through reliable evidence retrieval and combined with domain-specific financial knowledge. The *XBRL Fact Tool* retrieves numerical evidence from both financial tables and textual disclosures, as long as the evidence is defined in XBRL filings. This suggests that XBRL can serve as a unified interface for grounding financial QA over heterogeneous financial data.

There are several directions for future work. First, we plan to support more complex financial analysis by incorporating richer XBRL relation types. This work focuses mainly on the presentation linkbase, leaving calculation, definition, and label linkbases for future study. Second, we plan to extend the *Formula Graph Tool* with more financial formulas and derived metrics to support advanced reasoning. These directions can further improve accurate, grounded, and reliable financial reasoning by LLM-based agents.

8 Limitations

Our method achieves strong results but still has several limitations. First, firms vary in how they organize presentation and calculation linkbases and may introduce custom concepts. As a result, the same financial concept may appear at different depths or along different paths across filings, reducing the robustness of structure-based retrieval. This issue becomes more severe when the query and the target concept have low lexical overlap, since the retrieval anchor may fall in an irrelevant section and path

exploration may fail to reach the relevant evidence. Second, our framework relies on ReAct, requiring the LLM to follow instructions and use tools reliably. This assumption may fail for weaker models, whose unstable instruction following can cause incorrect tool use and degraded answer quality.

Although our system retrieves evidence from structured regulatory disclosures, retrieval or reasoning errors may still lead to inaccurate financial interpretations and affect downstream financial analysis or decision-making if the system is used without human verification. The framework may also inherit inconsistencies, omissions, or reporting biases present in corporate disclosures and XBRL taxonomies, resulting in uneven performance across firms and industries. More broadly, large-scale automated financial analysis systems may be misused to generate misleading financial information or support market manipulation. Since our framework uses LLMs as reasoning agents, it may also be affected by hallucination, overconfidence, and unreliable tool use. These risks highlight the importance of human oversight, careful validation, and transparent evidence tracing when applying the system in real-world financial settings.

References

Anthropic. 2026. Claude 4.6 sonnet. https://www.anthropic.com.

Abhinav Arun, Fabrizio Dimino, Tejas Prakash Agarwal, Bhaskarjit Sarmah, and Stefano Pasquali. 2025. Finreflectkg: Agentic construction and evaluation of financial knowledge graphs. In *Proceedings of the 6th ACM International Conference on AI in Finance*, ICAIF '25, page 283–290, New York, NY, USA. Association for Computing Machinery.

David Balsiger, Hans-Rudolf Dimmler, Samuel Egger-Horstmann, and Thomas Hanne. 2024. Assessing large language models used for extracting table information from annual financial reports. *Computers*, 13(10).

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 335–336, New York, NY, USA. Association for Computing Machinery.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Yuemin Chen, Feifan Wu, Jingwei Wang, Hao Qian, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, and Meng Wang. 2024b. Knowledge-augmented financial market analysis and report generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1207–1217, Miami, Florida, US. 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. 2021. 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. 2022. 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.

Haitao Cheng, Ke Wang, Qi Wang, Tao Liu, and Kai Sheng. 2026. A chinese financial event knowledge graph-based retrieval-augmented generation framework for financial question answering. *Engineering Applications of Artificial Intelligence*, 175:114670.

Chanyeol Choi, Jihoon Kwon, Jaeseon Ha, Hojun Choi, Chaewoon Kim, Yongjae Lee, Jy-yong Sohn, and Alejandro Lopez-Lira. 2025. Finder: Financial dataset for question answering and evaluating retrieval-augmented generation. In *Proceedings of the 6th ACM International Conference on AI in Finance*, ICAIF '25, page 638–646, New York, NY, USA. Association for Computing Machinery.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Sarah Elhammadi, Laks V.S. Lakshmanan, Raymond Ng, Michael Simpson, Baoxing Huai, Zhefeng Wang, and Lanjun Wang. 2020. A high precision pipeline for financial knowledge graph construction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 967–977, Barcelona, Spain (Online). International Committee on Computational Linguistics.

European Securities and Markets Authority. 2025. MiCA white paper XBRL taxonomy documentation. Version 1.0. Accessed 25 April 2026.

Sorouralsadat Fatemi and Yuheng Hu. 2024. Enhancing financial question answering with a multi-agent reflection framework. In *Proceedings of the 5th ACM International Conference on AI in Finance*, ICAIF '24, page 530–537, New York, NY, USA. Association for Computing Machinery.

Ghislain Fourny. 2020. *The XBRL Book: Simple, Precise, Technical*. Amazon Digital Services LLC - KDP Print US.

Ryan George, Akshay Govind Srinivasan, Jayden Koshy Joe, Harshith M R, Vijayavallabh J, Hrushikesh Kant, Rahul Vimalkanth, Sachin S, and Sudharshan Suresh. 2025. Enhancing financial RAG with agentic AI and multi-HyDE: A novel approach to knowledge retrieval and hallucination reduction. In *Proceedings of The 10th Workshop on Financial Technology and Natural Language Processing*, pages 19–32, Suzhou, China. 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, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Shijie Han, Haoqiang Kang, Bo Jin, Xiao-Yang Liu, and Steve Y Yang. 2024. Xbrl agent: Leveraging large language models for financial report analysis. In *Proceedings of the 5th ACM International Conference on AI in Finance*, ICAIF '24, page 856–864, New York, NY, USA. Association for Computing Machinery.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. Knowledge graphs. *ACM Computing Surveys*, 54(4):1–37.

Rani Hoitash, Udi Hoitash, and Landi Morris. 2021. extensible business reporting language (xbrl): A review and implications for future research. *AUDITING: A Journal of Practice & Theory*, 40(2):107–132.

Ivan Iaroshev, Ramalingam Pillai, Leandro Vaglietti, and Thomas Hanne. 2024. Evaluating retrieval-augmented generation models for financial report question and answering. *Applied Sciences*, 14(20).

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Xiang Li, Penglei Sun, Wanyun Zhou, Zikai Wei, Yongqi Zhang, and Xiaowen Chu. 2025. Finkario: Event-enhanced automated construction of financial knowledge graph. *Preprint*, arXiv:2508.00961.

Yitao Long, Tiansheng Hu, Yilun Zhao, Arman Cohan, and Chen Zhao. 2025. FinLFQA: Evaluating attributed text generation of LLMs in financial long-form question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 16730–16750, Suzhou, China. Association for Computational Linguistics.

Rungsiman Nararatwong, Chung-Chi Chen, Natthawut Kertkeidkachorn, Hiroya Takamura, and Ryutaro Ichise. 2024. DBQR-QA: A question answering dataset on a hybrid of database querying and reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15169–15182, Bangkok, Thailand. Association for Computational Linguistics.

Yuqi Nie, Yaxuan Kong, Xiaowen Dong, John M. Mulvey, H. Vincent Poor, Qingsong Wen, and Stefan Zohren. 2024. A survey of large language models for financial applications: Progress, prospects and challenges. *Preprint*, arXiv:2406.11903.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, and 108 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

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, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2023. Gpt-3.5 turbo. `https://developers.openai.com/api/docs/models/gpt-3.5-turbo`.

OpenAI. 2024. Gpt-4o mini. `https://developers.openai.com/api/docs/models/gpt-4o-mini`.

Karmvir Singh Phogat, Chetan Harsha, Sridhar Dasaratha, Shashishekar Ramakrishna, and Sai Akhil Puranam. 2023. Zero-shot question answering over financial documents using large language models. *Preprint*, arXiv:2311.14722.

Bhaskarjit Sarmah, Dhagash Mehta, Benika Hall, Rohan Rao, Sunil Patel, and Stefano Pasquali. 2024. Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. In *Proceedings of the 5th ACM International Conference on AI in Finance*, ICAIF '24, page 608–616, New York, NY, USA. Association for Computing Machinery.

Shalin Shah, Srikanth Ryali, and Ramasubbu Venkatesh. 2024. Multi-document financial question answering using llms. *Preprint*, arXiv:2411.07264.

Neelesh Kumar Shukla, Prabhat Prabhakar, Sakthivel Thangaraj, Sandeep Singh, Weiyi Sun, C Prasanna Venkatesan, and Viji Krishnamurthy. 2025. GraphRAG analysis for financial narrative summarization and a framework for optimizing domain adaptation. In *Proceedings of the Joint Workshop of the 9th Financial Technology and Natural Language Processing (FinNLP), the 6th Financial Narrative Processing (FNP), and the 1st Workshop on Large Language Models for Finance and Legal (LLMFinLegal)*, pages 23–34, Abu Dhabi, UAE. Association for Computational Linguistics.

Aditi Singh, Abul Ehtesham, Saket Kumar, Tala Talaei Khoei, and Athanasios V. Vasilakos. 2026. Agentic retrieval-augmented generation: A survey on agentic rag. *Preprint*, arXiv:2501.09136.

Dannong Wang, Jaisal Patel, Daochen Zha, Steve Y. Yang, and Xiao-Yang Liu. 2025a. Finlora: Benchmarking lora methods for fine-tuning llms on financial datasets. *Preprint*, arXiv:2505.19819.

Xinyu Wang, Jijun Chi, Zhenghan Tai, Tung Sum Thomas Kwok, Hailin He, Zhuhong Li, Yuchen Hua, Muzhi Li, Peng Lu, Suyucheng Wang, Yihong Wu, Huang Jerry, Jingrui Tian, Fengran Mo, Yufei Cui, and Ling Zhou. 2025b. Finsage: A multi-aspect rag system for financial filings question answering. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, page 6144–6152, New York, NY, USA. Association for Computing Machinery.

Yi-Cheng Wang, Wei-An Wang, and Chu-Song Chen. 2026. Document-level numerical reasoning across single and multiple tables in financial reports. *Preprint*, arXiv:2604.03664.

XBRL International. 2013. Extensible business reporting language (XBRL) 2.1. Recommendation 31 December 2003; corrected errata version 20 February 2013. Accessed 25 April 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Tan Yue, Rui Mao, Xuzhao Shi, Shuo Zhan, Zuhao Yang, and Dongyan Zhao. 2025. QAEval: Mixture of evaluators for question-answering task evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14717–14730, Vienna, Austria. Association for Computational Linguistics.

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.

Hao Zhou, Luis Felipe Costa Sperb, and Tiejun Ma. 2025. Zero-shot extraction of stock relationship graphs with LLMs. In *Proceedings of The 10th Workshop on Financial Technology and Natural Language Processing*, pages 12–18, Suzhou, China. Association for Computational Linguistics.

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.

A Appendix

A.1 XBRL-structured Statement Extraction

To construct the `xbrl` field, we integrate information from XBRL schemas, presentation linkbases, and instance files. We first parse the schema to identify major financial statements, including the balance sheet, income statement, and cash flow statement, by examining the `<link:roleType>` tags and their corresponding `<link:definition>` elements. We then use the associated `roleURI` values to locate the corresponding presentation networks in the presentation linkbase. For each network, we extract financial concepts and their `order` attributes to reconstruct the hierarchy of the original financial statement. Finally, we retrieve the corresponding facts from the XBRL instance file and insert them into the extracted hierarchy according to their concepts. The resulting XBRL-structured statements constitute the `xbrl` field of each dataset entry.

A.2 Example data from XBRL-FinNQA

Table 3 shows an example data entry from our proposed dataset. Each entry contains the following fields: CIK, company name, html_index, doc_period, question, answer, context, xbrl, statements_for_this_question, and target_concepts. The html_index field provides the homepage URL of the corresponding filing in SEC EDGAR, while doc_period indicates the reporting period. The context field is used in the CoT + Flattened Text experiment, whereas the xbrl field is used in the CoT + XBRL experiment. The statements_for_this_question field specifies the financial statement relevant to the question, and target_concepts lists the corresponding XBRL financial concepts required to answer it.

A.3 Twelve key financial metrics

Table 4 presents the twelve financial metrics used for question generation. These metrics are selected from four commonly used categories of financial analysis and are primarily computed from balance sheets, income statements, and cash flow statements.

A.4 Example of a Formula Graph

In a conventional AST, operand roles are implied by child positions in a binary tree. In our setting, however, the same financial item can appear in multiple financial metric formulas, forming a graph rather than independent trees. Therefore, operand roles cannot be inferred from connectivity alone. We use explicit `left_operand` and `right_operand` edges to preserve each item's role in a specific formula and enable formula reconstruction.

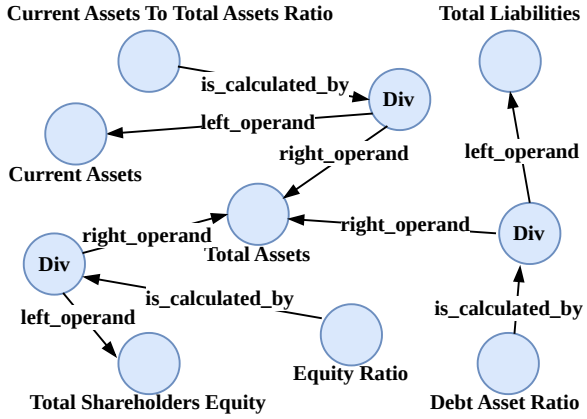

Figure 4: An example of a formula graph, where financial metrics, financial items, and arithmetic operators are modeled as nodes. Edges specify either calculation dependencies, using `is_calculated_by`, or operand roles, using `left_operand` and `right_operand`.

A.5 Implementation Details

We use LangChain[10] and LangGraph[11] to implement the ReAct-style agent workflow. The agent is allowed to iterate for at most 10 Thought/Action/Observation steps, and the LangGraph recursion limit is set to 50. For all LLM-based methods, we use a temperature of 0.7. For the CoT baselines, we set the maximum output length to 1,024 tokens. For the CoT + XBRL setting, we evaluate on a randomly sampled 25% subset of XBRL-FinNQA with random seed 42 because the XBRL context is substantially longer than the flattened-text context.

For XBRL fact retrieval, we use Arelle to parse XBRL instance files and NetworkX to construct presentation graphs. We use sentence-transformers/all-mpnet-base-v2 as the text encoder and compute cosine similarity between the query and XBRL concept names. In the XBRL Fact Tool, we select two anchor concepts for each query. For our proposed GMMR anchor selection method, we set $\lambda = 0.5$. Query-guided path exploration is run for at most three steps. During path exploration, expansion nodes are ranked using BAAI/bge-reranker-v2-m3 as the cross-encoder reranker, with a reranker score threshold of 0.0.

[10] https://github.com/langchain-ai/langchain
[11] https://github.com/langchain-ai/langgraph

After path expansion, candidate XBRL concepts whose query-concept cosine similarity is below 0.25 are filtered out. We exclude non-value-bearing XBRL concepts whose names end with Axis, Domain, Member, Abstract, LineItems, Table, RollForward, TextBlock, or List. We use the NLTK[12] words corpus to identify uppercase non-dictionary tokens as abbreviations during XBRL concept processing.

For the Formula Graph Tool, we use Python's built-in AST module to parse financial metric formulas and NetworkX to construct directed formula graphs. Financial metrics, financial items, and arithmetic operators are represented as nodes. Edges encode calculation dependencies and operand roles, including is_calculated_by, left_operand, and right_operand. Formula retrieval uses sentence-transformers/all-mpnet-base-v2 to encode the query and financial metric names, and selects the metric with the highest cosine similarity.

The agent also uses several auxiliary tools. The calculator tool is implemented with LangChain's LLMMathChain using GPT-3.5-Turbo. The rounding tool rounds numerical outputs to two decimal places. The XBRL terminology retriever uses RecursiveCharacterTextSplitter with chunk size 1,000 and chunk overlap 20, stores embeddings in Chroma[13], and retrieves the top 2 relevant chunks. The Wikipedia lookup tool uses the wikipedia-api[14] package to retrieve short summaries when external general knowledge is needed.

For evaluation, we use GPT-4o-mini to extract the final numerical answer from each model generation. We then parse the numeric value using a regular expression. Both predicted answers and ground-truth answers are rounded to two decimal places. Accuracy is computed by comparing the rounded predicted and ground-truth values using Python's `math.isclose` function with its default settings, i.e., `rel_tol=1e-09` and `abs_tol=0.0`.

[12] https://www.nltk.org/
[13] https://github.com/chroma-core/chroma
[14] https://pypi.org/project/Wikipedia-API/

CIK	1571996
company	Dell Technologies Inc.
html_index	https://www.sec.gov/Archives/edgar/data/1571996/0001571996-21-000007-index.html
doc_period	2021-01-29
question	What is Dell Technologies Inc.'s FY2021 Debt Asset Ratio? Respond to the question by assuming the perspective of an investment analyst who can only use the details shown. Please express your answer as a decimal and round your answer to second decimal places.
answer	0.934975489203095
context	DELL TECHNOLOGIES INC. CONSOLIDATED STATEMENTS OF FINANCIAL POSITION (in millions) January 29, 2021 January 31, 2020 ASSETS Current assets: Cash and cash equivalents 14, 201 9,302 ...
xbrl	<?xml version="1.0"?> <link:presentationLink xmlns="http://www.xbrl.org/2003/instance" xmlns:link="http://www.xbrl.org/2003/linkbase" xmlns:us-gaap="http://fasb.org/us-gaap/2020-01-31" xmlns:xbrldi="http://xbrl.org/2006/xbrldi" xmlns:xlink="http://www.w3.org/1999/xlink" xlink:role="http://www.dell.com/role/ consolidatedstatementsoffinancialposition" xlink:type="extended"> ... <us-gaap:CashAndCashEquivalentsAtCarryingValue contextRef= "iadcf37eef02248b1afbcceba92098c36_I20210129" unitRef="usd" decimals="-6"> 14201000000 </us-gaap:CashAndCashEquivalentsAtCarryingValue> ... </link:presentationLink>
statements for this question	http://www.dell.com/role/CONSOLIDATEDSTATEMENTSOFFINANCIALPOSITION
target_concepts	Assets, Liabilities

Table 3: Example data from XBRL-FinNQA

Metric Type	Financial Metric	Required Statements
Liquidity	Quick Ratio	Balance Sheet
Liquidity	Current Ratio	Balance Sheet
Liquidity	Operating Cash Flow Ratio	Balance Sheet and Cash Flow Statement
Efficiency	Inventory Turnover Rate	Balance Sheet and Income Statement
Efficiency	Accounts Receivable Turnover Rate	Balance Sheet and Income Statement
Efficiency	Current Assets to Total Assets Ratio	Balance Sheet
Solvency	Debt Asset Ratio	Balance Sheet
Solvency	Equity Ratio	Balance Sheet
Profitability	Basic EPS	Income Statement
Profitability	Gross Profit Margin	Income Statement
Profitability	Net Profit Margin	Income Statement
Profitability	Return on Assets	Balance Sheet and Income Statement

Table 4: Twelve financial metrics