

Enhancing Financial Numerical Reasoning: A Neuro-Symbolic Retrieval-Augmented Generation Framework for Mitigating Hallucinations in Large Language Models

Student Name: Ahmed Naji Lutf

Course: Scientific Research Methodologies in Computer Science

Date: December 18, 2025

Table of Contents

Abstract	1
Introduction	1
Background	2
Problem Statement	2
Research Objectives	3
Research Scope	4
Literature Review	5
Research Methodology	7
System Architecture: The Multi-Agent Workflow	7
Experimental Design (The FAITH-Mask Protocol)	8
Limitations	8
Expected Outcomes	9
Gantt Chart	10
References	11

Abstract

The integration of Large Language Models (LLMs) into the high-stakes domain of financial services presents a critical paradox: while these models demonstrate unprecedented fluency in processing unstructured narrative text, they exhibit a structural inability to perform reliable arithmetic and reasoning over heterogeneous data sources. This phenomenon, often termed "hallucination," manifests in finance as the generation of plausible but factually incorrect numerical figures—a failure mode that is unacceptable in regulatory reporting, investment analysis, and compliance auditing. This research report addresses the **"Empirical Validation Gap"** in Financial LLMs by proposing a novel, specialized Retrieval-Augmented Generation (RAG) system designed to decouple linguistic generation from numerical calculation.

Drawing on the emerging hypothesis of **"Liar Circuits"** within transformer architectures (Mirajkar et al., 2025) and the limitations of stochastic token prediction, we articulate a Neuro-Symbolic framework—**FinCalc-RAG**. This system integrates a **"Program of Thoughts" (PoT)** reasoning engine that compels the model to generate executable Python code for arithmetic derivation rather than predicting numerical tokens directly. Furthermore, we address the challenge of heterogeneous data by implementing a **Table-Aware retrieval mechanism** inspired by the TableRAG and FAITH frameworks. This report provides an exhaustive analysis of the theoretical underpinnings of arithmetic hallucinations, details the architectural specifications of the proposed system, and outlines a rigorous experimental methodology using the FinQA and TAT-QA benchmarks to measure the reduction in intrinsic numerical errors and scale discrepancies.

Introduction

The digital transformation of the financial sector has reached an inflection point with the advent of Generative Artificial Intelligence (GenAI). Large Language Models (LLMs) such as GPT-4, Claude 3, and Llama 3 have demonstrated a remarkable capacity to parse vast repositories of unstructured data, offering the potential to automate labor-intensive tasks such as summarizing earnings call transcripts, extracting risk factors from 10-K filings, and synthesizing market sentiment analysis. The ability to query complex financial documents in natural language promises to democratize financial analysis, reducing the time required for due diligence from days to minutes.

However, the transition from experimental pilots to production-grade financial systems faces a formidable barrier: the non-negotiable requirement for absolute precision. In domains such as creative writing or casual dialogue, the stochastic nature of LLMs—their tendency to "hallucinate" or invent plausible details—can be seen as a feature of creativity. In finance, it is a catastrophic vulnerability. The financial domain operates under a strict definition of truth,

distinct from the clinical facts of medicine or the legal precedents of law. In finance, truth is often numerical, tabular, and time-dependent. A statement such as "Revenue increased by 5%" is either mathematically true based on the underlying data, or it is false. There is no ambiguity. Yet, standard LLMs, trained to predict the next token based on statistical likelihood rather than symbolic logic, frequently fail this binary test. They function as "stochastic parrots," prioritizing linguistic fluency over factual accuracy.

Background

In the domain of software engineering, it is axiomatic that there is "**no silver bullet**"—only trade-offs. This principle, famously articulated by Fred Brooks, applies equally to the current state of Artificial Intelligence. While "Scaling Laws" have improved the general capabilities of LLMs by increasing parameter counts and training data, they have not resolved the fundamental issue of reliability in high-stakes domains like medicine and finance (Xu et al., 2024).

Financial data is uniquely structured; it is not merely text but a complex interweaving of narrative disclosures (unstructured) and rigorous accounting tables (structured). A single answer often requires retrieving a value from a table (e.g., "Total Assets") and a value from a footnote (e.g., "Pending Litigation Costs") and combining them. Studies show that the most dangerous errors in financial AI are **numerical and temporal**, such as getting a stock price wrong for a specific day or misinterpreting a fiscal year versus a calendar year. A model might read a table correctly but fail to notice the header says "(in millions)," leading to a calculation that is off by six orders of magnitude—a "scale error."

This reality forces us to abandon the search for a "magical single model" that can do everything end-to-end. Instead, we must focus on building **Composite AI Systems** or **Neuro-Symbolic Architectures**. These systems leverage the LLM for its strength (understanding intent and language) while offloading the weakness (calculation and logic) to deterministic tools. This research is situated at this intersection of Neural AI (LLMs) and Symbolic AI (Code Execution).

Problem Statement

The core problem addressed in this research is the **Empirical Validation Gap** in Financial LLMs. This gap exists between the *perceived* capability of these models—their ability to speak the language of finance fluently—and their *actual* capability to perform rigorous, multi-step numerical reasoning without hallucination. Specifically, this research addresses three critical failure modes:

1. **Structural "Liar Circuits":** Research into the internal mechanics of transformer models has revealed that arithmetic errors are not merely random noise but are often the result of specific internal failures. The "Liar Circuit" hypothesis suggests that distinct

components within the model's architecture—specifically aggregation layers like Layer 46 in GPT-2 XL—act as bottlenecks where the model attempts to "guess" a number based on its aggregate probability mass rather than computing it symbolically (Mirajkar et al., 2025). When a model is asked to compute "Revenue minus Cost," it does not perform subtraction; it retrieves the token that is most statistically likely to follow that sequence of words. In high-stakes finance, relying on probabilistic guessing is untenable.

2. **The Heterogeneous Data Challenge (Context Schism):** Financial documents are complex mixtures of text and tables. Standard Retrieval-Augmented Generation (RAG) systems struggle with this because they typically "flatten" tables into text strings during the indexing process. This linearization destroys the spatial relationships—rows and columns—that define the data's meaning. This leads to "**Context Schism**," where the model can no longer distinguish between "2022 Revenue" and "2023 Revenue" because the column headers were lost or disassociated from the cell values during the chunking process (Yu et al., 2025).
3. **The Verification Void:** Perhaps the most critical aspect of the problem is the lack of verifiability. When a standard LLM outputs a number, it provides no audit trail. There is no proof of calculation. In financial auditing and compliance, a result without a derivation path is useless. The system must not only provide the correct answer but also demonstrate the *method* used to achieve it, allowing human experts to verify the logic.

Research Objectives

We aim to design a new, proven method to stop numerical mistakes in financial AI.

- **Primary Goal:** To design, test, and measure a specialized **Neuro-Symbolic RAG system** that focuses on numbers to stop multi-step numerical hallucinations in financial AI models, aiming to lower critical error rates.
- **Objective 1 (Architectural Design):** To design "FinCalc-RAG," a system that strictly separates numerical calculations from natural language generation. The architecture will enforce a "Program of Thoughts" (PoT) workflow, where the LLM is restricted to writing code, and a Python interpreter is responsible for executing that code (Khatuya et al., 2025).
- **Objective 2 (Dataset Creation):** To create a specialized test set, the "**FAITH-Mask**" benchmark, derived from existing complex benchmarks (FinQA, TAT-QA). This dataset will specifically isolate and test "scale errors" (e.g., confusion between millions and billions) and "derivation accuracy" in 10-K filings (Bang et al., 2025).
- **Objective 3 (Empirical Evaluation):** To evaluate the proposed FinCalc-RAG system against current state-of-the-art baselines (Naive RAG, Chain-of-Thought RAG). We will measure the reduction in numerical error rates (Calculation Accuracy) and the improvement in evidence retrieval (Table Recall).

- **Objective 4 (Component Analysis):** To analyze which specific modules of the new system—specifically the **Table-Aware Retriever** versus the **Symbolic Executor**—contribute most significantly to hallucination mitigation. This will provide clear, actionable steps for building trustworthy financial AI.

Research Scope

Defining the boundaries of this study is crucial for its feasibility and scientific rigor. This research is scoped as follows:

1. Domain Scope: Corporate Financial Reporting

We focus exclusively on Corporate Finance, specifically the analysis of Annual Reports (10-K filings) and Quarterly Reports (10-Q filings) from S&P 500 companies. These documents represent the "gold standard" of financial truth. They are dense, highly structured, and legally binding, making them the ideal testbed for hallucination research. We exclude informal financial data such as news articles, social media sentiment, or stock price tick data, as these introduce noise that distracts from the core problem of document reasoning.

2. Task Scope: Numerical Question Answering (QA)

The primary task is Numerical QA requiring multi-hop reasoning.

- *In Scope:* Operations such as addition, subtraction, multiplication, division, ratio calculation (e.g., "debt-to-equity"), trend analysis (e.g., "CAGR"), and unit conversion.
- *In Scope:* Questions that require synthesizing data from a table row and a corresponding text footnote (heterogeneous reasoning).
- *Out of Scope:* Generative tasks like "Write an investment thesis" or "Predict next year's stock price." We are interested in the accuracy of *retrieved* and *calculated* facts, not speculative generation.

3. Data Scope and Timeframe

The study will utilize financial filings from the fiscal years 2020 to 2024. This five-year window covers diverse economic conditions (the COVID-19 pandemic, inflation, recovery), ensuring the models are tested on varied numerical distributions. We will construct our test set by sampling 500 document segments from the FinQA and TAT-QA datasets, augmented with the FAITH masking strategy.

Literature Review

In Large Language Models (LLMs), hallucination refers to the production of fluent and grammatically correct outputs that are nevertheless factually incorrect or unsupported by external evidence. Although this issue affects many application areas, it poses a particularly severe risk in financial contexts, where numerical accuracy is binary and even minor errors can lead to significant consequences. Xu et al. (2024) characterize hallucination as an inherent limitation of probabilistic language models, arguing that conventional training paradigms are insufficient to completely eliminate it. Within financial applications, hallucinations are commonly classified as intrinsic—where generated outputs contradict the source material—or extrinsic—where the model fabricates information not present in the data. This study focuses primarily on intrinsic hallucinations, as they indicate a failure of Retrieval-Augmented Generation (RAG) systems to faithfully reproduce retrieved financial information.

Recent progress in mechanistic interpretability has provided insights into the architectural origins of hallucinations, particularly in arithmetic reasoning. The “Liar Circuit” hypothesis proposed by Mirajkar et al. (2025) suggests that such errors arise from systematic failures within specific transformer components rather than from random noise. Their analysis of the GPT-2 XL architecture identifies a two-stage arithmetic process: an intermediate distributed “scratchpad” spanning the middle layers, where information is accumulated, followed by a late-stage aggregation circuit that produces the final answer. This aggregation mechanism, localized around Layer 46, functions as a bottleneck. When the preceding information is incomplete or ambiguous—a common scenario when processing complex financial tables—the model defaults to a high-probability guess instead of a computed result. These findings reinforce the argument that mitigating hallucinations in financial reasoning requires architectural or procedural interventions that bypass the model’s internal numerical computations.

Retrieval-Augmented Generation, first introduced by Lewis et al. (2020), has become the dominant framework for grounding LLM outputs in external knowledge. However, standard RAG approaches perform poorly on financial documents due to a phenomenon often described as “structure loss.” Early implementations relied on naive text chunking, which fragmented tabular data by separating headers from their corresponding values, thereby stripping numerical entries of their semantic context. To address this limitation, Yu et al. (2025) proposed TableRAG, a structure-aware retrieval framework that preserves tabular integrity by independently retrieving table schemas

and individual cells. By leveraging SQL-style interactions and treating tables as databases, TableRAG resolves what is known as the “context schism” problem and enables more reliable numerical grounding.

Parallel efforts to improve reasoning accuracy have explored alternatives to natural-language-based computation. Wei et al. (2022) introduced Chain-of-Thought prompting, encouraging models to generate step-by-step reasoning processes. While this approach improves logical coherence, the underlying calculations remain embedded within the model’s neural weights and thus remain vulnerable to arithmetic errors. Building on this limitation, Khatuya et al. (2025) proposed the FINDER framework, which adopts a Program-of-Thoughts paradigm tailored to financial reasoning. Instead of performing calculations implicitly, the model generates executable Python code to carry out numerical operations. Evaluations on the FinQA benchmark demonstrate a 5.98% improvement in execution accuracy over standard Chain-of-Thought methods, providing empirical support for the use of external, code-based reasoning engines in financial analysis.

More recent literature has shifted toward agentic and validation-driven RAG systems to handle the complexity of financial disclosures such as 10-K filings. Srinivasan et al. (2025) introduced an agentic RAG framework employing a Multi-HyDE strategy, in which autonomous agents generate multiple hypothetical answers to expand the retrieval space before querying documents. Their approach achieved a 15% reduction in hallucination rates, supporting the use of analyst-style agents capable of iterative querying when initial retrieval attempts are insufficient. Complementing this direction, Lee and Yu (2025) proposed FRED, a fine-tuned small language model designed specifically to detect and correct hallucinations in financial text. Their categorization of errors into numerical, temporal, and entity-based types provides a granular evaluation framework well-suited for financial applications. Additionally, Strich et al. (2025) released T²-RAGBench, a benchmark focused on text-and-table retrieval, demonstrating that hybrid retrieval approaches combining sparse BM25 and dense methods outperform purely dense retrieval when locating exact numerical values. This finding directly informs the adoption of hybrid retrieval strategies in financial RAG systems. Finally, Zhao et al. (2025) expanded the evaluation landscape with FinRAGBench-V, a multimodal benchmark incorporating charts and visual elements from financial reports. Although the present study is limited to text and tables, their emphasis on visual citation and precise data localization underscores the growing importance of traceability, a requirement addressed in this work through code-level execution and validation mechanisms

Research Methodology

We propose a **Design Science Research (DSR)** approach, focused on the creation and evaluation of an IT artifact: the **FinCalc-RAG** system. The methodology follows a "Glass Box" philosophy, prioritizing transparency over black-box performance.

System Architecture: The Multi-Agent Workflow

The FinCalc-RAG system is designed as a pipeline of four specialized agents, implemented using the **LangGraph** framework:

1. The Ingestion Agent (The Librarian):

- *Task*: Parses raw PDF 10-Ks.
- *Innovation*: Uses **Microsoft Table Transformer** (DETR) to detect table bounding boxes. Instead of converting tables to text, it extracts them as **HTML/Markdown** structures and linked **Pandas DataFrames**.
- *Metadata*: Tags every chunk with {Company, Year, Quarter, Unit, Section}. This metadata is critical for preventing "Context Schism."

2. The Retrieval Agent (The Researcher):

- *Task*: Finds relevant data for the user query.
- *Strategy*: **Hybrid Search**.
 - *Dense Search*: Uses bge-m3 embeddings to find semantic matches (e.g., "profitability").
 - *Sparse Search*: Uses **BM25** to find exact matches (e.g., "Item 7", "\$14.5 million").
 - *Reranking*: A Cross-Encoder model reranks the top 20 results to select the top 5 most relevant "Evidence Chunks."

3. The Reasoning Agent (The Analyst - PoT Engine):

- *Task*: Formulates the calculation logic.
- *Constraint*: This agent is **forbidden** from calculating. It outputs **Python Code** only.
- *Prompting*: Uses "Program of Thoughts" (PoT) few-shot examples. It defines variables from the retrieved table cells (e.g., revenue_2023 = 10500) and writes the formula (e.g., growth = (revenue_2023 - revenue_2022)/revenue_2022).

4. The Execution Agent (The Verifier):

- *Task*: Executes the code.
- *Environment*: A secure, sandboxed Python environment.
- *Reflexion Loop*: If the code fails (e.g., syntax error), the error is sent back to the Analyst agent to self-correct and retry up to 3 times. The final output is the result of the code execution.

Experimental Design (The FAITH-Mask Protocol)

- **Dataset:** We will curate a **FAITH-Mask** test set containing 500 samples from S&P 500 10-K filings (2020-2024).
- **Procedure:**
 1. Select a "Gold" answer from the filing (e.g., "Net Income: \$50M").
 2. Mask this value in the source text.
 3. Feed the surrounding context and the question to the model.
 4. Verify if the model can derive the unmasked value.
- **Baselines:**
 - *Baseline 1:* Naive RAG (GPT-4o) - Direct text generation.
 - *Baseline 2:* CoT RAG (GPT-4o) - Text-based step-by-step reasoning.
 - *Experiment:* FinCalc-RAG (PoT) - Code-based reasoning.

Limitations

While the proposed FinCalc-RAG framework aims to address critical gaps, several limitations inherent to the methodology and scope must be acknowledged:

1. **Latency and Computational Cost:** The multi-agent workflow, specifically the iterative "Reflexion Loop" and the "Program of Thoughts" generation, inherently introduces higher latency (estimated 5-10 seconds per query) compared to standard end-to-end RAG (sub-second). This trade-off between *speed* and *accuracy* makes the system unsuitable for high-frequency trading or real-time arbitrage scenarios, limiting its application to back-office analysis and reporting.
2. **Dependency on Retrieval Quality:** The "Program of Thoughts" engine is deterministic only in its *calculation*; it is still probabilistic in its *variable selection*. If the "Researcher" agent fails to retrieve the correct table or extracts a row with a similar name (e.g., retrieving "Gross Revenue" instead of "Net Revenue"), the Python code will execute perfectly but yield the wrong answer. The system is thus limited by the precision of the underlying retrieval module.
3. **Data Parsing Constraints:** The success of Table-Aware retrieval relies heavily on the quality of PDF parsing. Older financial filings (pre-2010) or scanned image-based PDFs may resist accurate table extraction by the Microsoft Table Transformer, potentially leading to "garbage in, garbage out" scenarios where the structured DataFrame is corrupted.
4. **Scope of Reasoning:** The system is designed for *explicit* numerical reasoning (arithmetic). It may struggle with *implicit* or "soft" reasoning tasks, such as interpreting the "tone" of a risk disclosure or inferring causality between a geopolitical event and a financial loss, which requires broader world knowledge not encoded in the 10-K tables.

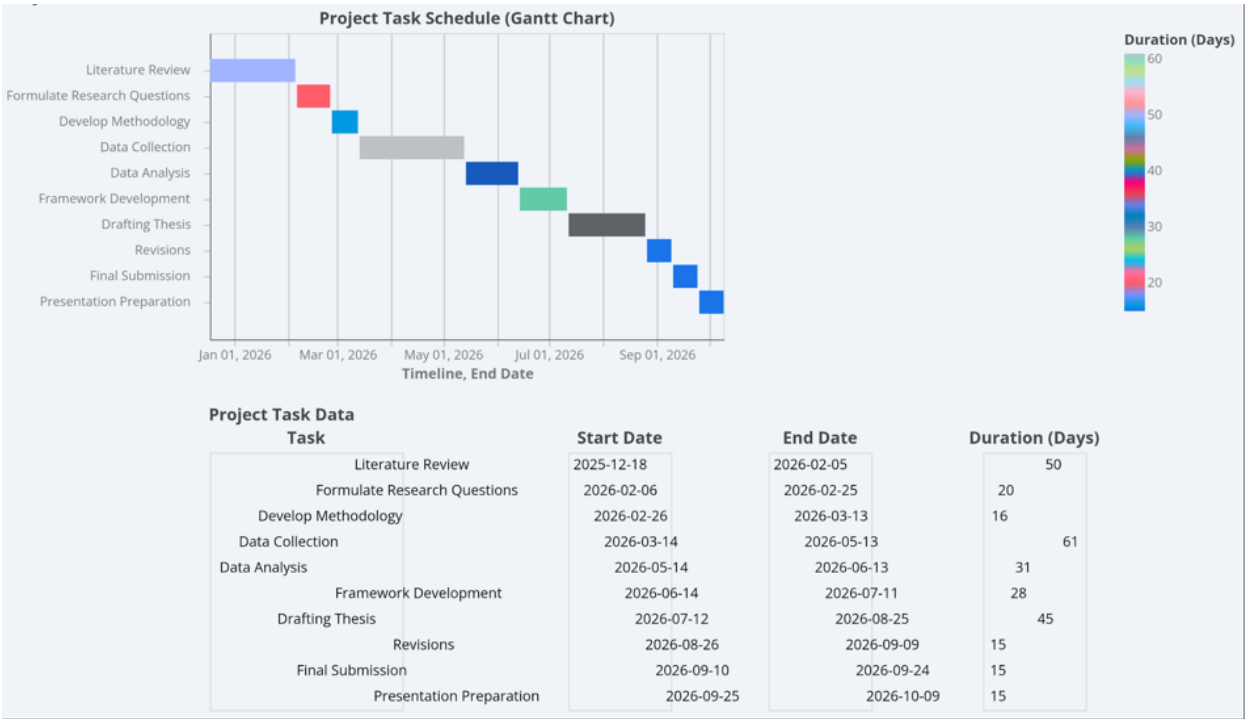
Expected Outcomes

By implementing this Neuro-Symbolic architecture, we anticipate the following specific, measurable outcomes:

1. **Reduction in Arithmetic Hallucinations:** We expect a statistically significant reduction (>15% improvement over Baseline 2) in numerical calculation errors. By shifting the "load" of calculation from the neural network to the Python interpreter, we eliminate the stochastic noise inherent in token prediction.
2. **Elimination of Scale Errors:** By treating financial data as structured variables (e.g., revenue = 1.5e9) in code rather than text tokens, we expect to virtually **eliminate scale errors** (confusing 1 million with 1 billion). The Python environment enforces strict type and unit consistency.
3. **Verifiable Audit Trails:** The system will produce a "**Code Trace**" for every answer. Unlike a standard LLM which says "The answer is 5%," FinCalc-RAG will say "The answer is 5%, and here is the Python script I used to calculate it from the raw data." This outcome is crucial for regulatory compliance.
4. **Robustness to Heterogeneous Data:** We expect the Table-Aware Retrieval to achieve higher **Recall@5** for questions that span both text and tables compared to Naive RAG, proving that structural preservation is key to retrieval accuracy.

Gantt Chart

The following schedule outlines the 8-month timeline for the Master's Thesis, from initial literature review to final presentation.



References

- Bang, Y., Fu, J., Warrier, T., Wang, Y., Tan, T., & Huang, K. (2025). FAITH: A framework for assessing intrinsic tabular hallucinations in finance. *arXiv preprint arXiv:2508.05201*. <https://arxiv.org/abs/2508.05201>
- Chen, Z., Li, S., Smiley, C., Ma, Z., Shah, S., & Wang, W. Y. (2022). ConvFinQA: Exploring the chain of numerical reasoning in conversational finance question answering. *arXiv preprint arXiv:2210.03849*. <https://arxiv.org/abs/2210.03849>
- Chen, Z., Wen, H., Smiley, C., & Wang, W. Y. (2021). FinQA: A dataset of numerical reasoning over financial data. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3697–3711. <https://aclanthology.org/2021.emnlp-main.294/>
- Khatuya, S., Naidu, S., Goyal, P., & Ganguly, N. (2025). Program of thoughts for financial reasoning: Leveraging dynamic in-context examples and generative retrieval. *Proceedings of EMNLP*, 30994–31006. <https://aclanthology.org/2025.emnlp-main.1577/>
- Lee, L., & Yu, S. (2025). REFIN: Quantifying model reliance in financial RAG. *arXiv preprint arXiv:2507.20930*. <https://arxiv.org/abs/2507.20930>
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N.,... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459–9474. <https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html>
- Mirajkar, S., et al. (2025). Dissecting the ledger: Locating and suppressing 'liar circuits' in financial large language models. *arXiv preprint arXiv:2511.21756*. <https://arxiv.org/abs/2511.21756>
- Srinivasan, A. G., George, R. J., Joe, J. K., Kant, H., M, R. H., Sundar, S., Suresh, S., Vimalkanth, R., & Vijayavallabh. (2025). Enhancing financial RAG with agentic AI and Multi-HyDE: A novel approach to knowledge retrieval and hallucination reduction. *arXiv preprint arXiv:2509.16369*. <https://arxiv.org/abs/2509.16369>
- Strich, J., Isgorur, E. K., Trescher, M., Biemann, C., & Semmann, M. (2025). T²-RAGBench: Text-and-table benchmark for evaluating retrieval-augmented generation. *arXiv preprint arXiv:2506.12071*. <https://arxiv.org/abs/2506.12071>

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35, 24824–24837. <https://arxiv.org/abs/2201.11903>

Xu, Z., Jain, S., & Kankanhalli, M. (2024). Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*. <https://arxiv.org/abs/2401.11817>

Yu, X., Jian, P., & Chen, C. (2025). TableRAG: A retrieval augmented generation framework for heterogeneous document reasoning. *arXiv preprint arXiv:2506.10380*. <https://arxiv.org/abs/2506.10380>

Zhao, S., Jin, Z., Li, S., & Gao, J. (2025). FinRAGBench-V: A benchmark for multimodal RAG with visual citation in the financial domain. *arXiv preprint arXiv:2505.17471*. <https://arxiv.org/abs/2505.17471>