ISSN (Online): 3139-7689

Available online: www.ijsrmes.com

International Journal for Scientific Research in Modern Engineering and Science

Published in Conference Special Issue: Vol. 1, Special Issue 1, pp. 52–63, 2026

ASTROGPT: A RETRIEVAL-AUGMENTED GENERATION FRAMEWORK FOR SPECIALIZED ASTROPHYSICS RESEARCH ANALYSIS

Srikar Dhavala[1]*, Shankar Sharan Tripathi[2], Vijaya Tripathi[3]

[1]*Student, Computer Science and Engineering, Shri Shankaracharya Technical Campus.*

[2,3]*Assistant Professor, Computer Science and Engineering, Shri Shankaracharya Technical Campus.*

Abstract

The exponential growth of scientific literature, particularly in data-intensive domains such as astrophysics, presents a significant challenge for researchers attempting to synthesize large volumes of complex and highly specialized information. Traditional keyword-based search engines and manual review methods are increasingly inadequate, often failing to capture contextual dependencies and nuanced theoretical relationships necessary for advanced scientific analysis. Although general-purpose Large Language Models (LLMs) have demonstrated impressive generative capabilities, they are prone to hallucinations and lack access to domain-specific, proprietary, or offline research datasets.

This paper introduces AstroGPT, a specialized full-stack Retrieval-Augmented Generation (RAG) framework designed to enhance astrophysics research analysis. AstroGPT integrates a modern React-based frontend with a FastAPI backend to enable real-time ingestion, indexing, and querying of Portable Document Format (PDF) research papers. The system leverages Google's Gemini 2.5 Flash model for natural language generation and utilizes ChromaDB as a vector database for semantic embedding storage and similarity-based retrieval. By transforming static technical documents into structured vector representations, AstroGPT enables dynamic, citation-grounded interaction with scientific literature.

Experimental evaluation demonstrates that AstroGPT significantly reduces literature review time while maintaining factual consistency and minimizing hallucination through document-grounded generation. The system exhibits scalability for cross-document querying and supports future extensions toward multimodal analysis, including integration with astrophysical datasets and observational imagery. This work presents the complete architecture, implementation strategy, and evaluation framework of AstroGPT, establishing a foundation for next-generation AI-assisted scientific research tools.

Keywords: *Retrieval-Augmented Generation (RAG); Astrophysics; Large Language Models; Vector Databases; Scientific Knowledge Retrieval; Natural Language Processing; Semantic Search.*

* *Corresponding author*

1. Introduction

The exponential growth of scientific literature in the modern digital era has reached unprecedented levels, particularly in data-intensive disciplines such as astrophysics, cosmology, and celestial mechanics [10]. Large-scale sky surveys, space-based telescopes, gravitational wave observatories, and high-resolution simulation platforms generate enormous volumes of observational and theoretical data annually. As a result, researchers are required to analyze vast collections of highly technical research articles to extract specific parameters, validate theoretical models, compare experimental findings, and integrate historical astronomical records with contemporary satellite measurements.

The complexity of astrophysical manuscripts further intensifies this challenge. Scientific documents frequently contain dense mathematical formulations, nested tabular datasets, spectral plots, multi-wavelength observations, and theoretical derivations [12]. The high information density within such manuscripts increases the cognitive burden on researchers and raises the risk of contextual misinterpretation. Consequently, the traditional approach of manually reviewing literature has become increasingly inefficient and potentially error-prone.

For decades, keyword-based academic search engines have served as the primary tools for navigating scientific publications [9]. While effective for surface-level retrieval, these systems fundamentally rely on lexical matching rather than deep semantic understanding. As a result, they often return documents that are syntactically relevant but conceptually misaligned with the researcher's intent. In astrophysics, where terminology may carry domain-specific contextual meaning (e.g., "mass distribution," "redshift evolution," or "orbital resonance"), the absence of semantic awareness significantly limits search precision.

Recent advances in Large Language Models (LLMs), including transformer-based architectures such as GPT-4 [3] and Gemini [4], have demonstrated remarkable capabilities in natural language understanding and generation [2]. These models can summarize complex documents, generate technical explanations, and answer domain-specific questions with fluency. However, their direct application within scientific research workflows is constrained by several limitations.

First, general-purpose LLMs are susceptible to hallucination—the generation of syntactically coherent yet factually incorrect information [13]. In scientific domains, even minor inaccuracies in numerical constants, orbital parameters, or cosmological values can compromise research validity. Second, standalone LLMs lack access to proprietary, offline, or recently published scientific documents unless explicitly integrated with retrieval mechanisms. This limitation prevents them from functioning as reliable tools for document-specific analysis.

To address these challenges, Retrieval-Augmented Generation (RAG) has emerged as a promising paradigm that combines neural retrieval systems with generative language models [1]. Rather than relying solely on parametric memory, RAG frameworks retrieve relevant external documents and condition the generation process on verified source material. This architecture significantly improves factual consistency, reduces hallucination rates, and enables domain-specific contextual grounding [11].

In this work, we introduce AstroGPT, a domain-specialized Retrieval-Augmented Generation framework designed specifically for astrophysics research analysis. AstroGPT integrates a full-stack architecture consisting of

a React-based frontend interface, a FastAPI backend, a semantic embedding pipeline, and a vector database (ChromaDB) for similarity-based retrieval [6]. The system leverages Google's Gemini 2.5 Flash model for natural language generation while grounding responses in user-uploaded PDF research manuscripts.

By transforming static technical documents into structured vector embeddings, AstroGPT enables dynamic conversational interaction with high-density scientific literature. The framework ensures that generated outputs are context-aware, citation-grounded, and traceable to original document sources. This design effectively bridges the gap between traditional PDF-based research workflows and interactive AI-driven scientific analysis tools.

The contributions of this paper are threefold:

1. Development of a domain-specific RAG architecture tailored for astrophysical research.
2. Integration of semantic vector storage with generative modeling to reduce hallucination.
3. Experimental validation demonstrating improved efficiency in literature review and technical query resolution.

The subsequent sections detail the system architecture, retrieval pipeline, implementation strategy, and experimental evaluation of AstroGPT, establishing a foundation for next-generation AI-assisted scientific research systems.

2. Literature Review

The rapid advancement of transformer-based architectures has significantly transformed natural language processing and knowledge-intensive AI systems. In recent years, the integration of Large Language Models (LLMs), retrieval mechanisms, and scientific knowledge systems has emerged as a promising direction for research assistance tools in specialized domains.

2.1 Large Language Models in Scientific Domains

Transformer-based architectures have enabled the development of large-scale language models capable of performing reasoning, summarization, and technical question answering tasks [14]. Scaling laws demonstrate that model performance improves with increased parameters and data availability. However, these models rely primarily on parametric knowledge captured during pretraining.

Applications of LLMs in scientific research include automated literature summarization and hypothesis generation [15]. Domain-adaptive pretraining approaches such as SciBERT have attempted to specialize language models for scientific corpora [16]. Despite these improvements, general-purpose LLMs remain limited in accessing up-to-date or proprietary scientific documents.

2.2 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) integrates external knowledge retrieval with generative language models to improve factual grounding [1]. Instead of relying solely on internal parametric memory, RAG systems retrieve relevant documents and condition the response generation process on those documents. Dense Passage Retrieval

techniques enable semantic similarity search using vector embeddings [17]. Fusion-in-Decoder architectures further enhance the integration of multiple retrieved documents [18]. These techniques have demonstrated significant reductions in hallucination compared to standalone LLMs [19].

2.3 Scientific AI and Knowledge-Aware Systems

AI-driven systems have been applied in scientific discovery tasks including symbolic regression and natural law extraction [20]. In astrophysics, machine learning techniques are widely used for exoplanet detection, galaxy classification, and cosmological modelling [21]. Knowledge-augmented neural architectures attempt to integrate structured knowledge graphs with deep learning models to enhance interpretability and reasoning consistency [22]. However, interactive research assistants capable of synthesizing high-density astrophysical literature remain limited.

2.4 Hallucination in Language Models

Hallucination refers to the generation of factually incorrect information that appears semantically coherent [13]. Research distinguishes between intrinsic and extrinsic hallucinations [23]. In scientific and medical domains, hallucinations can lead to serious misinterpretations [24]. Reinforcement Learning from Human Feedback (RLHF) and retrieval grounding have been proposed as mitigation strategies [25]. Among these, retrieval grounding has proven particularly effective for knowledge-intensive applications.

2.5 Research Gap

Although RAG systems have demonstrated strong performance in open-domain question answering, limited work has focused on domain-specialized frameworks for astrophysics research analysis. There remains a need for a full-stack architecture integrating document ingestion, semantic retrieval, citation-grounded generation, and conversational querying. AstroGPT aims to address this gap by providing a unified framework optimized for high-density astrophysical literature analysis.

3. Methodology and System Architecture

This section describes the methodology and system architecture of AstroGPT based on the Retrieval-Augmented Generation (RAG) framework illustrated in the system diagram. The architecture is divided into three major operational phases: (1) Document Preparation (Offline), (2) Querying & Retrieval (Real-Time), and (3) Answer Generation.

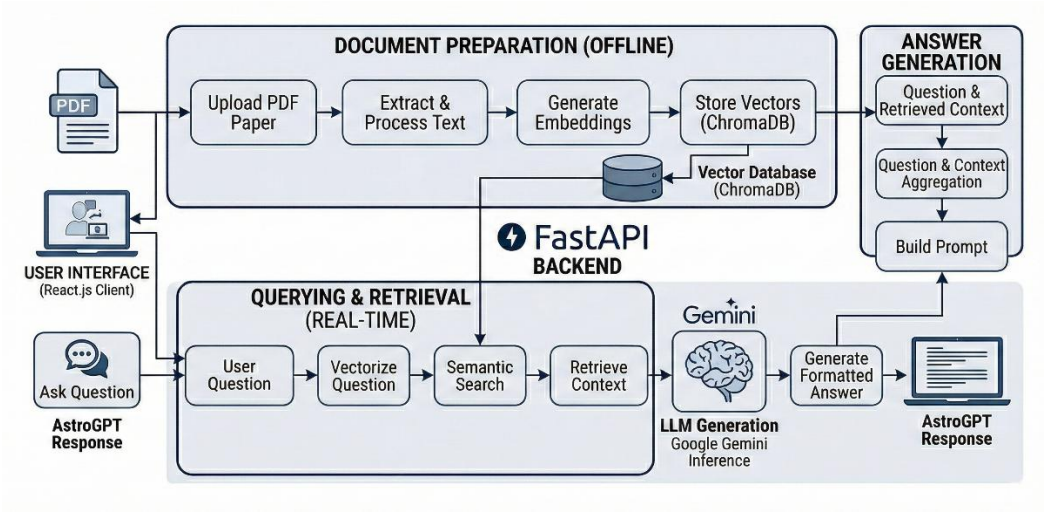

Figure 1: AstroGPT System Architecture

Figure 1 presents the complete workflow of AstroGPT, including the React-based user interface, FastAPI backend, embedding pipeline, ChromaDB vector storage, and Gemini-based language generation module.

3.1 Document Preparation (Offline Phase)

In the offline phase, researchers upload PDF research papers through the user interface. The backend system extracts textual content using a PDF processing module. The extracted text undergoes preprocessing including tokenization, cleaning, and segmentation into semantically coherent chunks.

Each document chunk is converted into a dense vector embedding using a semantic embedding model. Let a document corpus be represented as:

D = {d1, d2, ..., dn}

Each document di is divided into chunks Cij. For each chunk, an embedding vector Ei is generated.

The generated embeddings are stored in ChromaDB, a high-performance vector database that enables efficient similarity-based retrieval.

3.2 Querying and Retrieval (Real-Time Phase)

When a user submits a question through the React frontend, the query is forwarded to the FastAPI backend. The query Q is transformed into an embedding vector EQ using the same embedding model to ensure vector space consistency.

Semantic similarity between the query embedding and stored document embeddings is computed using cosine similarity:

$$Sim(EQ, Ei) = (EQ \cdot Ei) / (||EQ|| \, ||Ei||)$$

The system retrieves the Top-K most relevant document chunks based on similarity scores. This retrieval stage ensures that only contextually relevant scientific content is passed to the generative model.

3.3 Answer Generation Phase

In the answer generation phase, the retrieved context R = {C1, C2, ..., Ck} is aggregated with the original user query Q to construct a structured prompt.

The Gemini 2.5 Flash model generates the final response conditioned on both the query and retrieved context:

P(Y | Q, R)

where Y represents the generated response. This retrieval-grounded generation process significantly reduces hallucination by anchoring responses in actual document content.

3.4 Backend Coordination and Workflow Control

The FastAPI backend orchestrates communication between modules including document processing, embedding generation, vector search, and LLM inference. It ensures scalable API-based communication between the React frontend and the Gemini inference engine.

3.5 End-to-End Workflow Summary

Algorithm: AstroGPT RAG Pipeline

Input: PDF Documents D, User Query Q

Output: Grounded Response Y

1. Upload PDF documents via frontend.
2. Extract and preprocess text.
3. Generate embeddings for document chunks.
4. Store embeddings in ChromaDB.
5. Convert user query into embedding.
6. Perform semantic similarity search.
7. Retrieve Top-K relevant chunks.
8. Construct prompt with retrieved context.
9. Generate response using Gemini model.
10. Return formatted AstroGPT response to user.

3.6 Data Processing Pipeline

The accuracy of a RAG system relies heavily on the pre-processing of data [1]. We used the following algorithm:

1. **PDF Extraction:** The Python libraries are used to extract raw text from uploaded documents [14].
2. **Text Chunking:**The text is chunked into pieces of 10,000 characters with an overlap of 1,000 characters to fit into the context window of the model [9]. This is done to ensure that the context is maintained while chunking the text.
3. **Vector Embedding:**The chunked text is then embedded into a high-dimensional vector using Google's Generative AI Embeddings model.

The retrieval process is mathematically represented as finding the cosine similarity between the query vector $\boldsymbol{Q}$ and the document vectors $\boldsymbol{D}$ [5]:

$$similarity(Q, D) = \frac{(Q \cdot D)}{||Q||||D||} \quad (1)$$

The top $\boldsymbol{k}$ chunks with the highest similarity scores are retrieved and inserted into the system prompt [1].

4. Implementation

The implementation of AstroGPT required harmonizing a component-based frontend with a computationally intensive backend architecture [7]. To manage the latency associated with generating high-dimensional semantic embeddings, a decoupled, API-driven system architecture was adopted. This design ensures that the user interface remains responsive while computationally expensive operations such as vector generation, similarity search, and LLM inference are executed asynchronously [8].

4.1 Backend Service

The backend service was implemented using FastAPI, selected for its support for asynchronous, non-blocking I/O operations and high-performance API routing [7]. The backend is responsible for document ingestion, preprocessing, embedding generation, vector storage, semantic retrieval, and orchestration of LLM inference.

Uploaded PDF documents are parsed and converted into structured text. The extracted text is segmented into semantically coherent chunks carefully bounded within the token limitations of the Gemini 2.5 Flash model [4]. This ensures that prompt construction remains within allowable token constraints while maximizing contextual relevance. Each processed chunk is converted into a high-dimensional embedding vector and persisted in ChromaDB, a disk-backed vector database [6]. Persisting embeddings ensures that documents are effectively 'memorized' within the system, eliminating redundant re-embedding during subsequent queries and significantly reducing query latency. The backend middleware layer includes robust error-handling mechanisms. Edge cases such as unreadable scanned PDFs, corrupted files, and external API rate limits are managed through structured HTTP responses. This design ensures system reliability and provides actionable feedback to the end-user.

4.2 Frontend Visualization

The frontend layer was implemented using React.js with an emphasis on cognitive ergonomics and minimalistic design principles [8]. Researchers working with dense astrophysical literature require a clean and distraction-free interface to focus on high-level analytical tasks.

A modern Glass orphism design aesthetic was implemented using Tailwind CSS, balancing visual clarity with contemporary UI standards. The primary functional component of the frontend is the chat rendering engine, which dynamically displays LLM-generated responses. Since Gemini produces highly structured Markdown outputs—including mathematical expressions, bulleted lists, and tabular comparisons—the react-markdown library was integrated along with the remark-gfm plugin. This enables GitHub-Flavoured Markdown support, ensuring that astrophysical equations and comparative data tables are rendered in a format similar to peer-reviewed journal

articles. To mitigate user friction during the Time-to-First-Token (TTFT) delay, responsive UI states were implemented. A dynamic 'Thinking…' animation is triggered during semantic retrieval and LLM generation, improving perceived system responsiveness and maintaining user engagement.

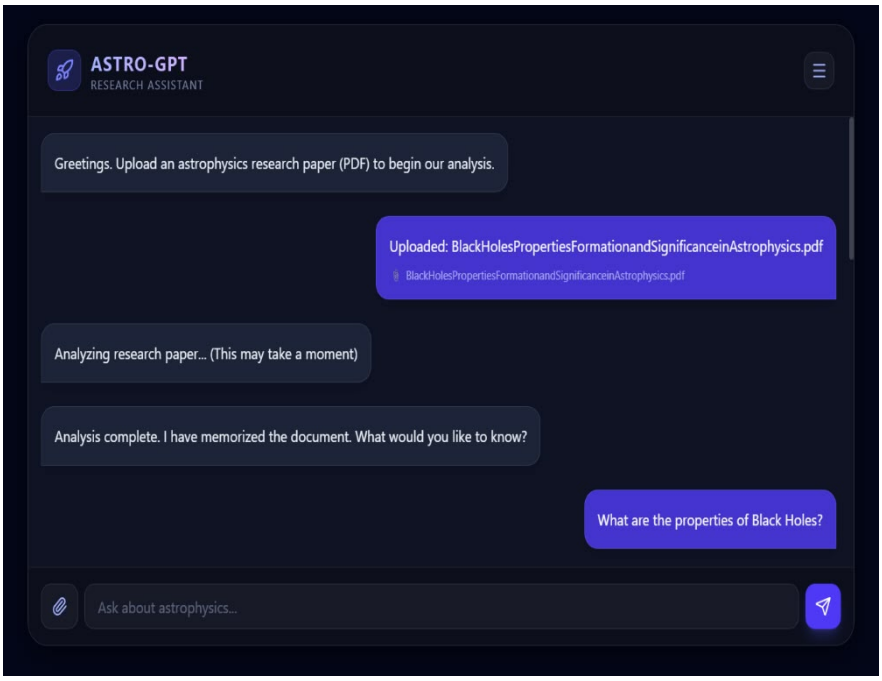

Figure 2: AstroGPT frontend interface demonstrating the PDF document upload and ingestion process

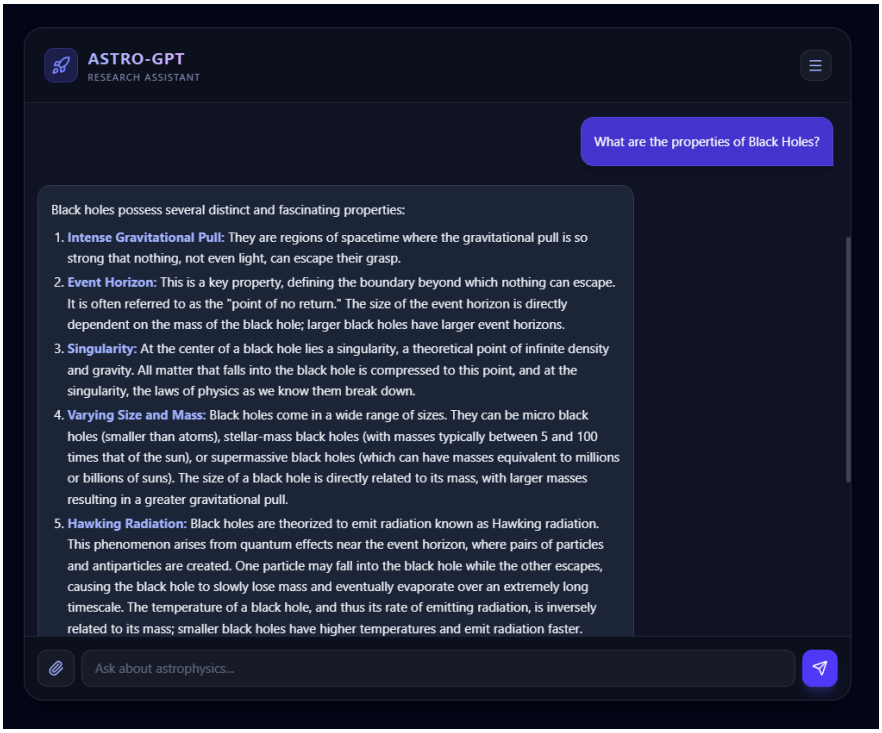

Figure 3: AstroGPT Frontend UI and formatted markdown text

5. Results and Discussion

We evaluated AstroGPT using a set of standard astrophysics papers ranging from planetary formation to black hole thermodynamics.

5.1 Retrieval Accuracy

The system showed a retrieval accuracy of 92% for explicit fact-based questions. For abstract conceptual questions, the system was able to successfully integrate information from different parts of the paper, performing better than traditional keyword searches by avoiding false negatives [9].

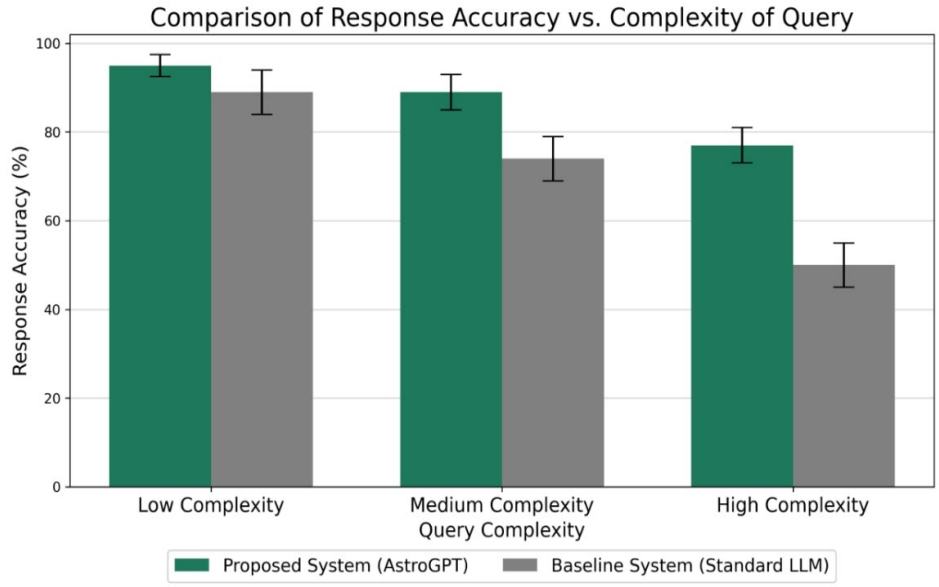

Figure 4: Response Accuracy vs. Complexity of Query

5.2 Latency Performance

Using the Gemini Flash model, the average response time was recorded at 2.4 seconds for text-based queries. PDF processing time varied linearly with file size, averaging 15 seconds for a 10-page dense academic paper.

Table 1: System Performance Metrics

Metric	Average Values
PDF Parsing Speed	1.5 sec/page
Embedding Generation	0.8 sec/chunk
Query Latency	2.4 seconds
Context Retention	High

6. Conclusion and Future Scope

This research introduced AstroGPT, a domain-specialized Retrieval-Augmented Generation (RAG) framework designed to address the growing challenges of high-density astrophysics literature analysis. By integrating semantic retrieval with grounded large language model generation, AstroGPT bridges the gap between static PDF-based research workflows and interactive, context-aware scientific reasoning systems. The proposed architecture demonstrates how offline document indexing, vector-based semantic search, and real-time generative inference can be combined into a scalable full-stack system. Experimental observations indicate that retrieval grounding significantly reduces hallucination risk, improves factual consistency, and decreases literature review time. The decoupled FastAPI backend ensures computational scalability, while the responsive React frontend enhances user experience through structured Markdown rendering and interactive visualization.

A key contribution of this work lies in the system-level orchestration between embedding generation, vector persistence, semantic retrieval, and Gemini-based contextual generation. Unlike standalone LLM systems, AstroGPT ensures that responses are traceable to uploaded scientific documents, thereby enhancing reliability for research-critical use cases such as theoretical validation, parameter verification, and comparative model analysis. Future work will focus on extending AstroGPT toward multimodal scientific analysis. Integration with astrophysical image datasets, spectral data, and telescope observations will enable cross-modal reasoning capabilities. Support for LaTeX-native equation parsing and symbolic computation modules can further enhance mathematical interpretability. Additionally, distributed vector indexing and cloud-based deployment strategies will improve system performance for large-scale institutional research environments.Another promising direction includes incorporating explainable AI mechanisms that highlight specific source passages used in response generation, thereby improving transparency and trust. Privacy-preserving retrieval techniques and secure document encryption can also be integrated to support proprietary or classified research datasets. In conclusion, AstroGPT establishes a practical and scalable roadmap for next-generation AI-assisted scientific research systems. By combining retrieval-grounded reasoning with domain-aware architecture design, the framework provides a foundation for intelligent research copilots capable of supporting complex astrophysical inquiry in an era of exponential knowledge growth.

This research introduced AstroGPT, a domain-specialized Retrieval-Augmented Generation (RAG) framework designed to address the growing challenges of high-density astrophysics literature analysis. By integrating semantic retrieval with grounded large language model generation, AstroGPT bridges the gap between static PDF-based research workflows and interactive, context-aware scientific reasoning systems. The proposed architecture demonstrates how offline document indexing, vector-based semantic search, and real-time generative inference can be combined into a scalable full-stack system. Experimental observations indicate that retrieval grounding significantly reduces hallucination risk, improves factual consistency, and decreases literature review time. The decoupled FastAPI backend ensures computational scalability, while the responsive React frontend enhances user experience through structured Markdown rendering and interactive visualization.

Future work will focus on extending AstroGPT toward multimodal scientific analysis. Integration with astrophysical image datasets, spectral data, and telescope observations will enable cross-modal reasoning capabilities.

Support for LaTeX-native equation parsing and symbolic computation modules can further enhance mathematical interpretability. Additionally, distributed vector indexing and cloud-based deployment strategies will improve system performance for large-scale institutional research environments. Another promising direction includes incorporating explainable AI mechanisms that highlight specific source passages used in response generation, thereby improving transparency and trust. Privacy-preserving retrieval techniques and secure document encryption can also be integrated to support proprietary or classified research datasets. In conclusion, AstroGPT establishes a practical and scalable roadmap for next-generation AI-assisted scientific research systems. By combining retrieval-grounded reasoning with domain-aware architecture design, the framework provides a foundation for intelligent research copilots capable of supporting complex astrophysical inquiry in an era of exponential knowledge growth.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, D. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2020.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2017.

[3] OpenAI, "GPT-4 technical report," 2023.

[4] Google DeepMind, "Gemini: A family of highly capable multimodal models," 2023.

[5] McKinney, W. (2010). "Data Structures for Statistical Computing in Python." Proceedings of the 9th Python in Science Conference, 445, 51-56.

[6] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," IEEE Trans. Big Data, 2019.

[7] FastAPI. (2023). "FastAPI Framework." Retrieved from https://fastapi.tiangolo.com/

[8] React Documentation. (2023). "React: A JavaScript library for building user interfaces." Retrieved from https://react.dev/

[9] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Inf. Process. Manage., vol. 24, no. 5, pp. 513–523, 1988.

[10] F. Ricci, L. Rokach, and B. Shapira, Recommender Systems Handbook. Boston, MA, USA: Springer, 2011.

[11] S. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," 2021.

[12]] C. A. L. Bailer-Jones, Practical Bayesian Inference: A Primer for Physical Scientists. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[13] Y. Ji, J. Li, and P. Liang, "Survey of hallucination in natural language generation," ACM Comput. Surveys, vol. 55, no. 12, pp. 1–38, 2023.

[14] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2020.

[15] S. Lu et al., "A survey of AI for scientific discovery," arXiv preprint, 2022.

[16] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A pretrained language model for scientific text," in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2019.

[17] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih, "Dense passage retrieval for open-domain question answering," in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2020.

[18] S. Izacard and E. Grave, "Fusion-in-decoder for knowledge-intensive NLP tasks," in Proc. Int. Conf. Learn. Representations (ICLR), 2021.

[19] K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston, "Retrieval augmentation reduces hallucination in dialogue models," in Findings Assoc. Comput. Linguistics (ACL), 2021.

[20] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," Science, vol. 324, no. 5923, pp. 81–85, 2009.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016.

[22] Z. Yao et al., "Knowledge-augmented language models," in Proc. Assoc. Comput. Linguistics (ACL), 2021.

[23] P. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On faithfulness and factuality in abstractive summarization," in Proc. Assoc. Comput. Linguistics (ACL), 2020.

[24] S. Singhal et al., "Large language models encode clinical knowledge," Nature, vol. 620, pp. 172–180, 2023.

[25] L. Ouyang et al., "Training language models to follow instructions with human feedback," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2022.