Quasar: An AI-Powered Research Assistant for Astronomy Archive Workflows

Adam Zacharia Anil
National Radio Astronomy Observatory
NSF-Simons AI Institute for Cosmic Origins
adamandspace@gmail.com

Adele Plunkett
National Radio Astronomy Observatory
NSF-Simons AI Institute for Cosmic Origins
aplunket@nrao.edu

Brian Mason
National Radio Astronomy Observatory
NSF-Simons AI Institute for Cosmic Origins
bmason@nrao.edu

Stella Offner
The University of Texas at Austin
NSF-Simons AI Institute for Cosmic Origins
soffner@utexas.edu

Abstract

The Atacama Large Millimeter/submillimeter Array (ALMA) Science Archive contains petabytes of data products spanning over a decade of observations. Despite rich programmatic interfaces, translating a research question into the right combination of archive queries, metadata filters, and analysis steps requires some amount of domain knowledge in the area of radio interferometry, and this can act as a hurdle particularly for students and interdisciplinary researchers unfamiliar with radio astronomy conventions. We present **Quasar**, an open-source AI research assistant that makes ALMA Science Archive search and data retrieval as simple as asking a question in natural language. Quasar pairs a registry of 35+ domain-specific tools covering a wide variety of use cases of Archive data search, retrieval and analysis with a **Conductor** orchestration engine that decomposes complex, multi-step research queries into Directed Acyclic task Graphs (DAGs). By externalizing task planning into the Conductor, Quasar enables even smaller or non-reasoning LLMs to reliably execute sophisticated archive workflows through structured tool composition. We demonstrate the framework's end-to-end capabilities through qualitative case studies highlighting complex tabular ALMA data retrieval, precise sandboxed mathematical computation, and literature extraction.

1 Introduction

Radio astronomy is experiencing an unprecedented era of data growth. The Atacama Large Millimeter/submillimeter Array (ALMA) [Wootten and Thompson, 2009], the most powerful ground-based facility for observations at millimeter and submillimeter wavelengths, has generated over 2 PB of science-ready data products since its inauguration in 2013. These data are publicly available through the ALMA Science Archive (ASA), which offers both web and programmatic interfaces for discovery and access. It also offers a corpus of multi-modal astrophysics data for testing AI methods.

Despite such rich interfaces, efficient archival exploration remains challenging for people without prior expertise in radio interferometry. A seemingly simple research question *"How many ALMA observations have been made of protostellar disks in Taurus with resolution better than 0.1 arcsec in Band 6?"* requires a researcher to: (1) know that "Band 6" maps to a specific frequency range in the ALMA metadata schema (or otherwise know the specific frequency range of interest), (2) resolve target names to coordinates and choose between name-based and cone-search strategies, (3) interpret angular resolution columns and apply the correct filter, and (4) cross-reference results with the

Conference on Physics and AI at Stanford University (PAI 2026).

literature to identify which observations target protostellar disks specifically. These steps collectively demand substantial domain knowledge and represent a significant barrier to entry for interdisciplinary researchers and students.

The emergence of large language models (LLMs) with tool-use capabilities [Yao et al., 2023, Schick et al., 2023, Patil et al., 2023] presents an opportunity to bridge this gap. While general-purpose LLMs possess broad language understanding, they lack the domain-specific tools, structured prompts, and orchestration logic needed to reliably navigate radio astronomy archives. A general purpose LLM may know what ALMA Band 6 is, but it cannot query the archive, resolve a target name to coordinates, filter by angular resolution, or extract metadata directly from remote FITS files; these capabilities must be engineered around the model through purpose-built tooling and workflow infrastructure. Moreover, even when equipped with tools, most LLMs struggle to plan across dependent invocations [Joseph et al., 2025].

In this paper, we present version 1 of **Quasar**[1], an open-source AI-powered research assistant for the ALMA Science Archive (ASA). It is a *domain-specialized agent framework* that wraps any general-purpose LLM with the tools, knowledge, and orchestration needed to perform real radio astronomy tasks. Our key insight is that **a general-purpose LLM becomes a capable scientific assistant not through model fine-tuning only but in combination with careful domain engineering**, with our main contributions described here:

- **An End-to-End Domain Architecture**: **Quasar** is an open-source framework demonstrating how generic LLMs can be adapted for expert-level astronomy through a curated suite of 35+ domain-specific tools.
- **Robust Workflow Orchestration**: Externalizing task planning into a Conductor-driven DAG reduces the cognitive burden on the backbone model.
- **Bridging Radio Data with NASA ADS and arXiv**: The framework connects ALMA data retrieval and astrophysical calculations with the scientific literature, including native integration of NASA ADS [Kurtz et al., 2000] and full-text PDF extraction from arXiv, to track scientific provenance.

Together, these components transform a model that "knows about" radio astronomy into one that can *interact* with the radio data ecosystem end-to-end.

2 System Design

Figure 1 shows the Quasar architecture. Incoming queries first enter the **QuasarAgent** via a web interface, which assembles context from three sources: (1) a shared RAG [Lewis et al., 2020] store indexing ALMA technical documentation (Technical Handbook[2], proposer's guide[3]) and optionally live web search results; (2) a *per-user personal vector collection* into which researchers can upload their own documents (observing proposals, scripts, notes), whose contents are retrieved alongside system knowledge when relevant; and (3) a *preference memory* built by summarizing past conversations, allowing the agent to learn each user's observational focus, preferred output formats, and common targets for more context-aware responses over time.

The query then passes through a **two-stage complexity detector**. First, a fast heuristic scan checks for pre-defined keywords and structural templates to classify the query as simple or complex. Ambiguous cases are forwarded to a secondary, smaller LLM (separate from the backbone) for a definitive complexity score. Simple queries are routed to the **Agent Tool Loop**, where the backbone LLM iterates over tool selection, execution, and observation until the answer is synthesized. Complex queries are escalated to the **Conductor**, which decomposes the request into a dependency-aware task graph and dispatches subtasks to specialist sub-agents (Archive, Literature, Analysis) sequentially or in parallel as dependencies permit.

The Conductor's sub-agents, particularly the Analysis agent, have access to a **sandboxed Python REPL** for running scripts. Rather than relying on the LLM to perform arithmetic or data transformations in-context, sub-agents can generate Python code that executes in a restricted environment with access to `numpy` [Harris et al., 2020], `astropy` [Astropy Collaboration et al., 2022], `pandas` [McK-

[1] Live demo: `https://www.quasarassistant.com`; source code: `https://github.com/adamzacharia/Quasar`

[2] `http://almascience.org/documents-and-tools/cycle13/alma-technical-handbook`

[3] `http://almascience.org/proposing/proposers-guide`

inney, 2010], and other common dependencies. This is critical for tasks that need more than just LLM reasoning to compute answers, such as numerical calculations [Tian et al., 2024]. The sandbox connects back to the full tool registry via a `call_tool()` interface, enabling hybrid workflows that combine programmatic computation with live archive access within a single subtask. The final response is accompanied by a downloadable Jupyter Notebook that reproduces the workflow, giving researchers full transparency and reproducibility.

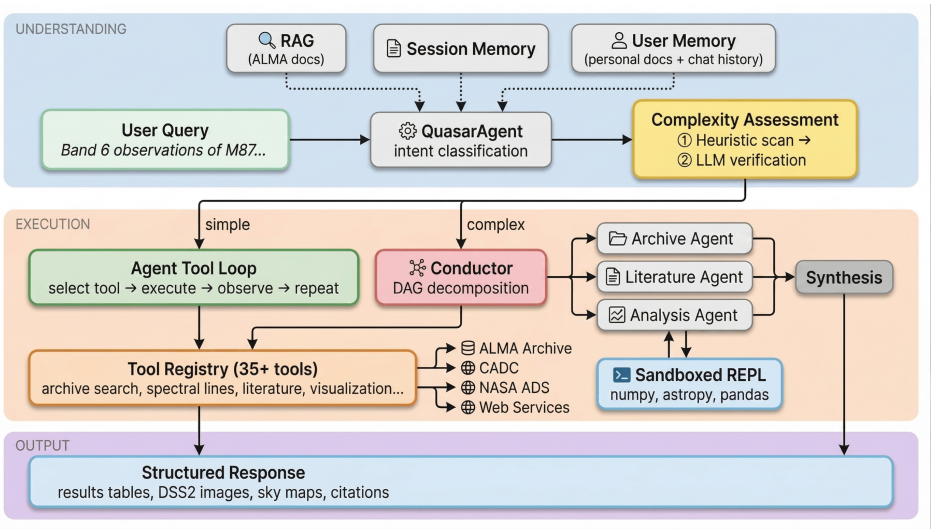

Figure 1: Quasar architecture. Simple queries (green) are handled by the Agent Tool Loop; complex queries are decomposed by the Conductor into a DAG dispatched to specialist agents (red). Both paths share the tool registry connecting to ALMA, CADC, NASA ADS and others.

3 Qualitative Evaluation & Case Studies

Instead of relying solely on static benchmarks, we evaluate Quasar's effectiveness by examining its behavior on real-world radio astronomy workflows. While version 1.0 is specifically tuned for ALMA, its underlying architecture is archive-agnostic and fully open-source.

3.1 Simple Workflow: Direct Archive Search

For straightforward queries, the system prioritizes speed and direct execution. Consider the query: *"Find ALMA observations of M87 in Band 6."* The complexity detector classifies this as simple, routing it to the Agent Tool Loop. The backbone LLM invokes the search by target tool, translating the natural language into structured Astronomical Data Query Language(ADQL) parameters (`target: M87, band: 6`). If the initial name-based query fails due to an informal or unresolved target name, the agent automatically catches the error, resolves the target to sky coordinates via SIMBAD, and re-issues the query as a coordinate-based cone search without requiring user intervention.

3.2 Complex Workflow: Orchestration and Sandboxing

Quasar's primary advantage emerges in multi-step scenarios where a significant amount of structural planning and continuous step-by-step execution is required. Consider the complex query: *"Check CO(2-1) line coverage for M87."*

The complexity detector flags this for escalation, routing it to the **Conductor**, which generates a task DAG:

1. **Archive Agent:** Retrieve all Band 6 observations for M87 from the ASA and extract their spectral window limits.
2. **Analysis Agent:** Calculate the redshifted frequency of the CO(2-1) line ($\nu_{\text{rest}} \approx 230.538\,\text{GHz}$) given M87's systemic velocity, and cross-match it against the observed frequency ranges.

3. **Synthesis:** Aggregate the matching projects and render a Markdown summary of the specific ALMA program codes and datasets.

During Step 2, the Analysis Agent evaluates whether to use its native reasoning or the Sandboxed REPL. For trivial operations such as a single redshift calculation, the LLM handles the math directly. For complex operations like intersecting dense arrays of spectral bounds, the agent invokes the REPL to run a Python script, bypassing unreliable LLM arithmetic. During Synthesis (Step 3), all steps are compiled into a downloadable Jupyter Notebook for full transparency and reproducibility.

3.3 Literature-Augmented Workflow: NASA ADS Integration

Beyond data retrieval, researchers frequently need to cross-reference archival data with the published scientific literature. Consider the query: *"Find the 2018 DSHARP survey overview paper by Andrews et al. and extract the exact angular resolution achieved for AS 209."*

Because this request is atomic, Quasar's routing layer bypasses formal DAG generation and directly dispatches the **Literature Agent**. Rather than hallucinating numbers from its pre-training weights, the agent uses its NASA ADS tool suite to identify the correct bibcode, retrieve the paper's full text, and accurately extract the requested synthesized beam parameters. By natively integrating ADS capabilities, Quasar allows users to seamlessly pivot from archive workflows to interrogating the published literature, all within a single conversational interface.

4 Related Work

LLM-Powered Astronomical Data Exploration. The intersection of LLMs and astronomy has primarily focused on text comprehension and knowledge extraction. AstroLLaMA [Nguyen et al., 2023] and its conversational extension [Ciuca et al., 2024] fine-tune foundation models on astronomical literature, while AstroMLab [Pan et al., 2024] evaluates frontier LLMs on astronomy knowledge. Pathfinder [Iyer et al., 2024] provides a semantic literature search framework over 350,000 ADS papers. Recent work [Hyk et al., 2025] offers recommendations for building better astronomy LLM benchmarks. In the physical sciences more broadly, Gravity-Bench [Koblischke et al., 2025] evaluates LLM agents on gravitational physics discovery tasks, while MARVEL [Mukund et al., 2026] presents a domain-aware multi-agent research assistant for gravitational-wave research.

More recently, systems have emerged that actively interface with astronomical data archives. The most closely related work is **StarAI** [Damian et al., 2025], which uses the Model Context Protocol [Anthropic, 2024] (MCP) to access the Canadian Astronomy Data Centre (CADC) and CANFAR science platform. StarAI employs RAG over proposal texts and translates natural language into ADQL via astroquery [Ginsburg et al., 2019] for TAP service querying. However, it relies primarily on standard LLM tool-calling loops to orchestrate queries iteratively.

Quasar differs from StarAI in scope: archive query and retrieval is just one of Quasar's capabilities, alongside sandboxed numerical analysis and literature extraction. Quasar externalizes planning through the **Conductor**, which parses dependencies into a DAG prior to execution, a paradigm pioneered by HuggingGPT [Shen et al., 2023]. Furthermore, Quasar tackles precise client-side scientific computation by embedding a **Sandboxed REPL**, allowing specialist sub-agents to mathematically intersect datasets in Python, bypassing the LLM's unreliable arithmetic.

5 Conclusion

We have presented Quasar, an open-source AI research assistant that synthesizes domain-specific tool use with robust task planning to make astronomical data retrieval and analysis accessible through natural language.

Limitations. The system depends on proprietary foundation models [OpenAI, 2023, Gemini Team et al., 2024] for reasoning and tool-calling, introducing cost and latency constraints. Additionally, the Conductor's plan relies on a single LLM generation call; if this step is flawed, errors cascade. While Quasar mitigates failures through iterative recovery loops, this single point of failure remains a bottleneck.

Future work. We plan to upgrade the Conductor with multi-LLM consensus protocols and enable long-horizon autonomous workflow pipelines. We also intend to migrate the backbone to open-weight LLMs for local deployment and complement the current qualitative evaluation with structured quantitative benchmarks [Hyk et al., 2025]. Finally, we aim to expand beyond ALMA to integrate multi-wavelength archives (such as those associated with National Radio Astronomy Observatory, Mikulski Archive for Space Telescopes, and NOIRLab's Astro Data Lab Science Platform), evolving Quasar into a cross-compatible assistant for multi-messenger astronomy.

References

Anthropic. Introducing the Model Context Protocol. `https://www.anthropic.com/news/model-context-protocol`, 2024. Accessed: 2026-03-25.

Astropy Collaboration, Adrian M. Price-Whelan, Pey Lian Lim, Nicholas Earl, Nathaniel Starkman, Larry Bradley, David L. Shupe, Aarya A. Patil, Lia Corrales, C. E. Brasseur, et al. The Astropy project: Sustaining and growing a community-oriented open-source project and the latest major release (v5.0) of the core package. *The Astrophysical Journal*, 935(2):167, 2022. doi: 10.3847/1538-4357/ac7c74.

Ioana Ciuca et al. AstroLLaMA-Chat: Scaling AstroLLaMA with conversational and diverse datasets. *Research Notes of the AAS*, 8(1):36, 2024.

Adrian Damian, Hossen Teimoorinia, Mrunal Mustapure, and Serhii Zautkin. StarAI - empowering access to astronomy data handled by the Canadian Astronomy Data Centre and its CANFAR platform with artificial intelligence. In *Submitted to ADASS XXXV*, Görlitz, Germany, 2025.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, et al. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2024.

Adam Ginsburg, Brigitta M. Sipőcz, C. E. Brasseur, Philip S. Cowperthwaite, Matthew W. Craig, Christoph Deil, James Guillochon, Giannina Guzman, Simon Liedtke, Pey Lian Lim, et al. astroquery: An astronomical web-querying package in Python. *The Astronomical Journal*, 157(3):98, 2019. doi: 10.3847/1538-3881/aafc33.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi: 10.1038/s41586-020-2649-2.

Alina Hyk, Kiera McCormick, Michael Zhong, Ioana Ciuca, Sanjib Sharma, John F. Wu, J. E. G. Peek, Kartheik G. Iyer, Zijian Xiao, and Anjalie Field. From queries to criteria: Understanding how astronomers evaluate LLMs. *arXiv preprint arXiv:2507.15715*, 2025.

Kartheik G. Iyer, Mikaeel Yunus, Charles O'Neill, Christine Ye, Alina Hyk, Kiera McCormick, Ioana Ciuca, John F. Wu, Alberto Accomazzi, et al. pathfinder: A semantic framework for literature review and knowledge discovery in astronomy. *The Astrophysical Journal Supplement Series*, 2024. doi: 10.3847/1538-4365/ad7c43. arXiv:2408.01556.

Sebastian Antony Joseph, Syed Murtaza Husain, Stella S. R. Offner, Stéphanie Juneau, Paul Torrey, Adam S. Bolton, Juan P. Farias, Niall Gaffney, Greg Durrett, and Junyi Jessy Li. AstroVisBench: A code benchmark for scientific computing and visualization in astronomy. *arXiv preprint arXiv:2505.20538*, 2025.

Nolan Koblischke, Hyunseok Jang, Kristen Menou, and Mohamad Ali-Dib. Gravity-Bench-v1: A benchmark on gravitational physics discovery for agents. *arXiv preprint arXiv:2501.18411*, 2025.

Michael J. Kurtz, Guenther Eichhorn, Alberto Accomazzi, Carolyn S. Grant, Markus Demleitner, and Stephen S. Murray. The NASA astrophysics data system: Overview. In *Astronomy and Astrophysics Supplement Series*, volume 143, pages 41–59, 2000.

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. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.

Wes McKinney. Data structures for statistical computing in Python. In *Proceedings of the 9th Python in Science Conference*, pages 56–61, 2010. doi: 10.25080/Majora-92bf1922-00a.

Nikhil Mukund, Yifang Luo, Fan Zhang, Lisa Barsotti, and Erik Katsavounidis. Marvel: a multi-agent research validator and enabler using large language models. *Machine Learning: Science and Technology*, 7(3):035023, 2026. doi: 10.1088/2632-2153/ae68ba.

Tuan Dung Nguyen, Marc Huertas-Company, Laurence Perreault-Levasseur, et al. AstroLLaMA: Towards specialized foundation models in astronomy. *arXiv preprint arXiv:2309.06126*, 2023.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Ting-Yun Pan et al. AstroMLab 1: Who wins astronomy jeopardy!? *arXiv preprint arXiv:2407.11194*, 2024.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive APIs. *arXiv preprint arXiv:2305.15334*, 2023.

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

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving AI tasks with ChatGPT and its friends in Hugging Face. *Advances in Neural Information Processing Systems*, 36, 2023.

Minyang Tian, Luyu Gao, Shizhuo Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, et al. SciCode: A research coding benchmark curated by scientists. *Advances in Neural Information Processing Systems*, 37:30624–30650, 2024.

Al Wootten and A. Richard Thompson. The Atacama Large Millimeter/submillimeter Array. *Proceedings of the IEEE*, 97(8):1463–1471, 2009.

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