PaperArena: An Evaluation Benchmark for Tool-Augmented Agentic Reasoning on Scientific Literature

Daoyu Wang, Mingyue Cheng, Shuo Yu, Zirui Liu, Ze Guo, Qi Liu
State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China, Hefei, China
{daoyu.wang,yu12345,liuzirui,gz1504921411}@mail.ustc.edu.cn,{mycheng,qiliuql}@ustc.edu.cn

Abstract

Understanding and reasoning on the web-scale scientific literature is a crucial touchstone for large language model (LLM) based agents designed to support complex knowledge-intensive tasks. However, existing works are mainly restricted to tool-free tasks within isolated papers, largely due to the lack of a benchmark for cross-paper reasoning and multi-tool orchestration in real research scenarios. In this work, we propose PaperArena, an evaluation benchmark for agents to address real-world research questions that typically require integrating information across multiple papers with the assistance of external tools. Given a research question, agents should integrate diverse formats across multiple papers through reasoning and interacting with appropriate tools, thereby producing a well-grounded answer. To support standardized evaluation, we provide a modular and extensible platform for agent execution, offering tools such as multimodal parsing, context retrieval, and programmatic computation. Experimental results reveal that even the most advanced LLM powering a well-established agent system achieves merely 38.78% average accuracy. On the hard subset, accuracy drops to only 18.47%, highlighting great potential for improvement. We also present several empirical findings, including that all agents tested exhibit inefficient tool usage, often invoking more tools than necessary to solve a task. We invite the community to adopt PaperArena to develop and evaluate more capable agents for scientific discovery. Our code and data are available[1].

Keywords

Evaluation Benchmark; Agentic Reasoning; Scientific Literature

ACM Reference Format:
Daoyu Wang, Mingyue Cheng, Shuo Yu, Zirui Liu, Ze Guo, Qi Liu. 2018. PaperArena: An Evaluation Benchmark for Tool-Augmented Agentic Reasoning on Scientific Literature. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

With the web-scale scientific literature continually expanding, researchers are increasingly challenged by information overload and declining efficiency [8, 18, 64]. As shown in Figure 1, this stems largely from the need to search across numerous papers and synthesize diverse and heterogeneous content such as text, charts, and pseudocode [24, 25]. In response to this challenge, a growing trend is to use LLM-based agents to assist researchers with their knowledge-intensive tasks that typically require navigating and reasoning on this vast literature corpus [14].

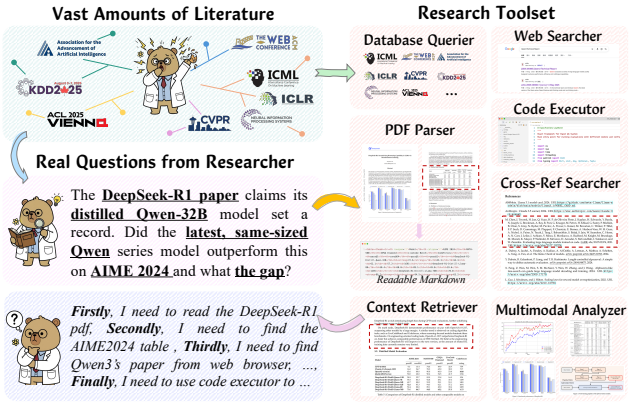

Figure 1: Confronted with vast literature, researchers must orchestrate a diverse toolset for cross-paper reasoning to answer complex scientific questions.

Despite this potential, progress in LLM-based agents for scientific research remains limited. A primary obstacle is the lack of a benchmark and an evaluation platform designed to assess agents' ability to address complex real-world research questions, which usually involve cross-document reasoning and coordinated tool usage. As shown in Table 1, existing benchmarks such as PubMedQA [22] and SPIQA [41] focus on question answering (QA) over scientific documents, but they typically restrict tasks to single-paragraph, single-modality settings, such as explaining a concept from a paragraph or extracting a value from a figure. However, modern advanced LLMs can often solve these directly using their inherent capabilities without tool invocation, offering limited challenge to tool-augmented agents [11, 30, 66]. Meanwhile, although recent efforts such as ReAct [60] and ToolLLM [43] begin to explore the tool-use capabilities of agents, they lack the specific adaptations of navigating and reasoning on scientific literature.

To bridge these gaps, we undertake the pioneering work of constructing a comprehensive benchmark and evaluation platform for LLM-based agents reasoning on scientific literature. This endeavor is challenging for several reasons. First, constructing the benchmark itself is a non-trivial task. It requires synthesize information across multiple papers, resolve inter-document citations through fine-grained search, and integrate heterogeneous modalities using

[1] https://github.com/Melmaphother/PaperArena

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

Table 1: Comparison of PaperArena with existing scientific question answering benchmarks across key dimensions.

Dataset	Agentic Tool-Use	Multi-Step Reasoning	Multimodal Understanding	Cross-doc Integration	Database Interfacing
PubMedQA [22]	✗	✗	✗	✗	✗
CharXiv [55]	✗	✗	✓	✗	✗
emrQA [39]	✗	✗	✗	✓	✗
QASA [29]	✗	✗	✗	✗	✗
ArgSciChat [45]	✗	✓	✗	✗	✗
QASPER [13]	✗	✗	✗	✗	✗
SPIQA [41]	✗	✓	✓	✗	✗
PaperArena	✓	✓	✓	✓	✓

specialized parsing tools. Second, building the evaluation platform necessitates implementing a dedicated toolset for scientific research and designing core agent execution mechanisms for planning, interaction with tools, long-context management, and reflection. Finally, a meaningful evaluation must extend beyond the correctness of the final answer. It must also assess the underlying reasoning process, including the efficiency and rationality of tool usage throughout the agent's entire reasoning trace.

With these challenges in mind, we propose PaperArena and its corresponding PaperArena-Hub. The construction of PaperArena begins with a corpus of tens of thousands of papers from premier open-access AI conferences. From this collection, we sample a representative subset and employ a multimodal large language model (MLLM) to automatically generate initial question-answer pairs. To ensure high complexity and broad coverage, we guide this generation process with a curated tool chain experience that mirrors realistic research scenarios. This process yields a final set of 784 high-quality question-answer pairs. Notably, the multi-step reasoning required for each question ensures a substantial evaluation cost. The cumulative complexity and number of required LLM calls are thus comparable to benchmarks with a much larger volume of simpler queries. Concurrently, we implement PaperArena-Hub to support the evaluation of both single-agent and multi-agent systems. The platform offers a modular tool suite with capabilities including multimodal parsing, context retrieval, and code execution, while also managing the complete agent lifecycle including planning, action, memory, and reflection.

We conduct extensive experiments on a range of leading LLM-based agents. Our results reveal a stark performance gap: the most advanced LLM, Gemini 2.5 Pro, powering a well-established multi-agent system achieves an average accuracy of only 38.78% and drops to 18.47% on the hard subset, a performance far below the 83.5% achieved by a human expert. Beyond overall accuracy, our in-depth analysis of agents' reasoning trajectories uncovers three key findings: (1) agents exhibit inefficient tool usage by often invoking more tools than necessary and showing a strong bias towards general-purpose tools; (2) despite attempting longer reasoning chains for complex tasks, their performance consistently and severely degrades; and (3) agents still struggle with both suboptimal planning and flawed tool invocation. These findings suggest clear directions for future improvements in agent capabilities. We invite the community to adopt PaperArena and PaperArena-Hub to develop and evaluate more capable agents for scientific discovery and to accelerate progress in agent-driven research.

In summary, our main contributions are as follows:

- We propose PaperArena, a novel and challenging benchmark featuring questions mirrors realistic research scenarios that demand agents to reason across multiple papers and orchestrate diverse tools.
- We present PaperArena-Hub, a modular and extensible open-source platform that enables standardized evaluation on LLM-based agents, including a dedicated scientific tool suite and full lifestyle management.
- Through extensive experiments, we reveal a significant performance gap and key failure patterns of leading LLM-based agents, offering concrete insights.

2 Related Work

In this section, we review scientific literature understanding benchmarks and tool-augmented agent with their evaluation. Finally, we explain how our work addresses the existing gaps.

2.1 Benchmarks on Scientific Literature

Benchmarks for scientific literature understanding have rapidly grown in the past few years. [16, 46]. Early benchmarks target specific tasks such as claim verification [52] and phrases extraction [6]. With the rise of LLMs, inquiries on knowledge-intensive scientific data receive increasing attention. For example, SciQA [5] focuses on scientific QA over structured academic knowledge. In specialized domains, PubMedQA [22] and BioASQ [26] focus on biomedical literature, while ChemLLMBench [15] extends evaluation to chemical challenges. Recently, LLM-based agents show promising potential, inspiring the development of expert-level benchmarks for scientific literature mining. In terms of reproducibility, ResearchCodeBench [19] evaluates machine learning code completion, whereas PaperBench [49] centers on reproducing code from scratch, providing only a paper. Numerous task-oriented benchmarks are also proposed to evaluate agents [10, 28, 34, 38]. In the domain of scientific inquiry, PeerQA [7] compiles QA pairs from peer review questions, and Humanity's Last Exam [40] assesses LLM performance in high-difficulty professional questions.

2.2 Tool-Augmented Agents

The rapid advancement of LLMs and their emerging reasoning capabilities [1, 9, 21, 30, 57] pave the way for the development of agent workflows. Early agents demonstrated the extension of reasoning through external interactions. WebGPT [35] showed LLMs retrieving and synthesizing information via web browsing, while SayCan [2] highlighted planning and acting in embodied tasks. Building on these ideas, ReAct [60] introduced a paradigm that synergizes reasoning and actions, marking the beginning of a new era for tool-augmented reasoning agents. Following ReAct, research has expanded the capabilities of agents in planning [20], tool use [44], memory [31, 65], and reflection [47]. Recently, there are also some influential works in multi-agent systems [17, 42, 56, 58, 63]. From the perspective of evaluation [54], PaperQA [27, 48] evaluate agents on real-world literature retrieval tasks and STELLA [23] evaluate agents through a dynamic tool ocean to advance biological research. However, existing works are restricted to tool-free tasks within isolated papers, with no effective benchmarks for tool-centric complex literature understanding. To bridge this gap, we propose PaperArena, a new benchmark for tool-augmented agents on scientific literature, and its corresponding evaluation platform.

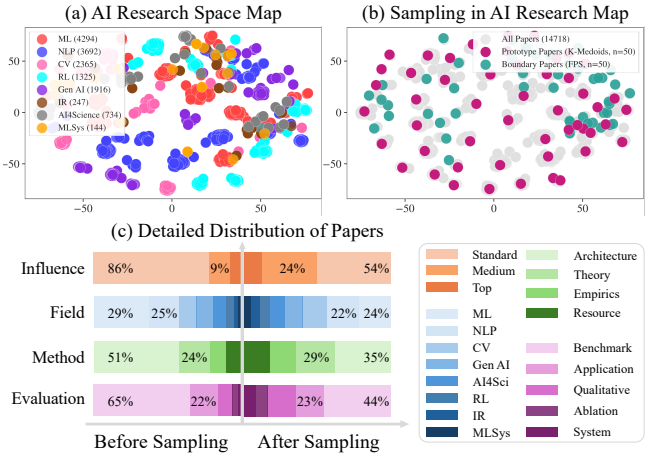

Figure 2: The effect of our sampling strategy in creating a more representative and comprehensive paper subset.

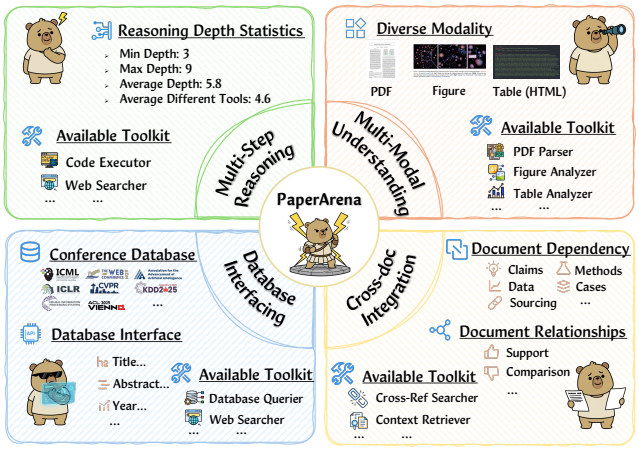

Figure 3: Key features of PaperArena, highlighting four core capabilities required for scientific reasoning agents.

3 The PaperArena Benchmark Construction

In this section, we detail the construction of the PaperArena, including a clear task formulation, the preparation of a diverse paper corpus, and a pipeline for generating QA pairs. Finally, we summarize the key features of PaperArena. Details in Appendix B.

3.1 Task Formulation

Given a complex question q, the goal in PaperArena is to produce a well-grounded answer by leveraging a predefined set of external tools $\mathcal{T}$, grounded in the content of a paper P. Each participating agent system $\mathcal{A}$ must plan and execute a multi-step reasoning process. Specifically, it first constructs an internal state s_0 from (P, q), and then iteratively applies tools $t_i \in \mathcal{T}$ to transition between intermediate states: $s_0 \xrightarrow{t_1} s_1 \xrightarrow{t_2} \cdots \xrightarrow{t_n} s_n$. At each step, the agent system must select the most suitable tool based on its evolving state and task needs. The final state s_n encodes the complete reasoning trace and yields the answer a. This evaluates the agent's ability to decompose complex questions, plan tool usage, and accumulate evidence across operations like realistic research scenarios.

3.2 Paper Corpus Preparation

We construct our benchmark on a curated corpus of 14,435 open-access latest AI papers from OpenReview[2] and Open Access[3] platforms. Each paper is labeled with a one-hot vector encoding its influence level, research field, method category and evaluation type. A t-SNE visualization of these vectors (Figure 2(a)) reveals strong domain-level clustering, indicating topic redundancy and the risk of evaluation bias. To address this, we adopt a hierarchical sampling strategy. We first apply K-Medoids to identify 50 representative prototype papers that reflect dominant research trends. Meanwhile, to ensure broader coverage, we further apply Farthest Point Sampling (FPS) to select 50 boundary papers from underrepresented regions. As shown in Figure 2(b) and (c), the resulting 100-paper subset balances representativeness and diversity, forming a solid foundation for tool-centric QA generation.

[2]https://openreview.net/
[3]https://www.openaccess.nl/

3.3 QA Generation from Papers

We construct tool-centric QA pairs from scientific papers through a three-stage pipeline that integrates tool library design, heuristic QA generation, and semi-automated verification.

3.3.1 Tool Library. To facilitate agentic interaction with scientific literature, we construct a comprehensive tool library, denoted as $\mathcal{T}$. As detailed in Figure 4, it comprises a curated suite of tools for multimodal parsing, external information retrieval, and cross-paper entity localization. This tool library is crucial for enabling advanced functionalities beyond the capabilities of traditional LLMs.

3.3.2 Heuristic QA Pair Generation. To generate diverse and challenging QA pairs, we first parse each paper's *pdf* into *markdown* format using MinerU[4] and segment it into structured sections. We then employ the state-of-the-art MLLM Gemini 2.5 Pro to produce draft QA pairs, using the paper metadata, segmented content, and extracted visual elements such as tables and figures as input. To guide this process toward high-complexity and high-coverage questions, we introduce a curated tool chain experience $\mathcal{E}$, which contains diverse and rational tool chains in real research scenarios. These experiences are designed to target multi-step, multimodal, cross-paper, and cross-database reasoning. Under the guidance of $\mathcal{E}$, the generation process yields QA pairs annotated with question type (e.g., Multiple Choice, Concise Answer) and difficulty level (Easy, Medium, Hard). This produces 1,215 high-quality QA pairs with their corresponding theoretical tool chains before filtering.

3.3.3 Semi-Automated QA Verification. By executing each generated tool chain step by step, we programmatically obtain intermediate outputs that reflect the expected reasoning trajectory for a given question. These outputs help identify invalid or underspecified QA pairs and guide the subsequent filtering process. We then refine each QA pair through two procedures: (1) *question obfuscation*, which removes explicit structural cues to increase difficulty, and (2) *answer standardization*, which unifies output formats and

[4]https://github.com/opendatalab/MinerU

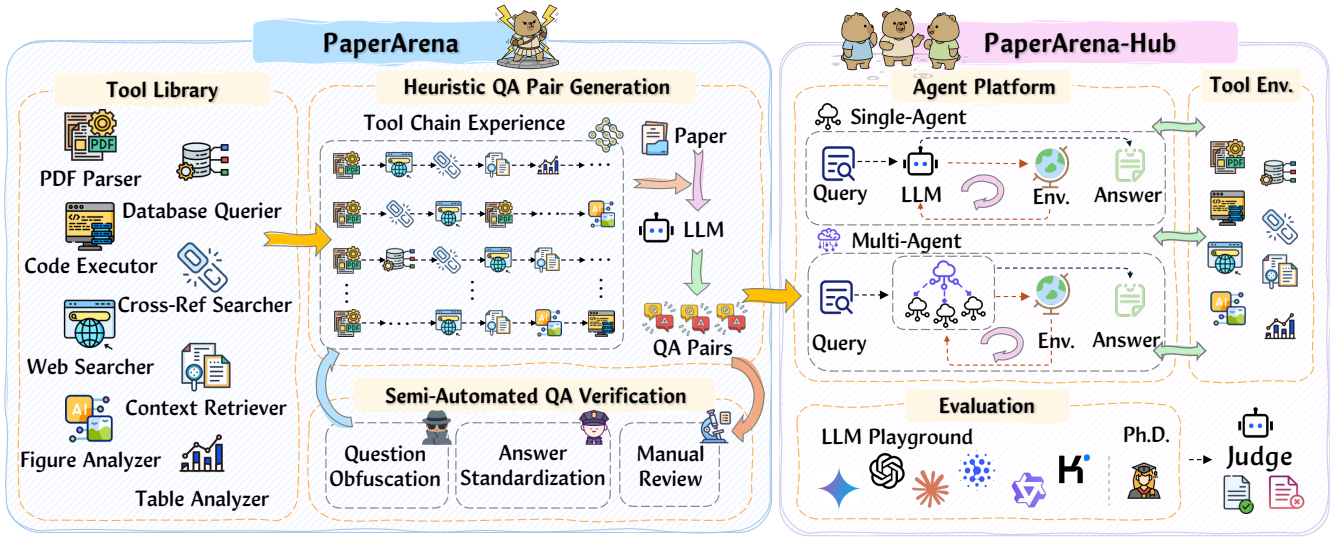

Figure 4: Illustration of the PaperArena benchmark construction and PaperArena-Hub platform details, featuring the tool-centric QA generation pipeline and the evaluation platform including single-agent and multi-agent systems.

introduces plausible distractors. During this stage, any newly discovered high-quality tool chains are added to the experience pool $\mathcal{E}$, forming a virtuous feedback loop that improves subsequent generations. After this automated refinement, all 1,215 QA pairs undergo human verification. We discard those deemed ambiguous, trivial, or infeasible, resulting in 784 ones that form our final benchmark.

3.4 Features of PaperArena

Consider that most existing benchmarks evaluate QA or retrieval over isolated papers, overlooking the multi-tool and cross-paper reasoning required in real research. As shown in Figure 3, PaperArena fills this gap by evaluating whether agents possess four essential capabilities to solve scientific problem:

Multi-Step Reasoning. PaperArena requires agents to perform multi-step reasoning that reflects real scientific workflows. An example question requires sequential tool use, such as parsing PDFs, analyzing tables, and searching the web. This process aims to trace claims, retrieve evidence, and validate results. This evaluates the agent's ability to effectively plan and then and interactively invoke tools across heterogeneous sources.

Multi-Modal Understanding. PaperArena requires agents to extract, align, and reason over multimodal content, including text, figures, tables, and equations. Tasks may involve aligning visual data with textual claims, comparing chart trends to tabular metrics, or interpreting formulas in context. These challenges test the agent's capacity for cross-modal analysis and integration.

Cross-Document Integration. PaperArena requires agents to reason over multiple papers by modeling citation relationships and synthesizing information across documents. Example Tasks include verifying whether a claim is supported or contradicted by another paper and retrieving implementation details from cited methods. This evaluates the agent's ability to synthesize coherent arguments and reconcile conflicting information from distributed sources.

Database Interfacing. PaperArena requires agents to issue precise queries over a structured paper database containing metadata such as titles, abstracts, and influence scores. Agents must interpret the returned results, extract relevant information, and incorporate it into subsequent reasoning steps. This tests their ability to interface with large knowledge sources and integrate retrieved evidence.

4 PaperArena-Hub: Agent Evaluation Platform

In this section, we introduce PaperArena-Hub, a modular and extensible platform for standardized and reproducible evaluation on PaperArena. We describe its architecture, integrated tool environment, and evaluation setup, with details in Appendix C.

4.1 Platform Overview

As shown in Figure 4, PaperArena-Hub is a modular evaluation platform built on the smolagents[5]. It supports both single-agent and multi-agent systems. In the single-agent system, we adopt the ReAct mechanism, where the LLMs reason and act in alternation. In the multi-agent setting, we use a centralized design, with a manager agent responsible for high-level planning and delegating subtasks to worker agents. The platform is designed to be extensible, enabling evaluations across different agent configurations and supporting fine-grained analysis of tool usage and reasoning strategies.

4.2 Integrated Tool Environment

PaperArena-Hub integrates the full suite of tools defined in PaperArena, allowing agents to interact with scientific papers, databases, and external resources. For example, our implementation of the database querying tool is backed by a structured dataset covering metadata from over one hundred major AI conferences in recent years. This includes paper titles, abstracts, influence scores, and document links. Agents can perform operations such as retrieving

[5]https://github.com/huggingface/smolagents

Table 2: Comprehensive evaluation of nine state-of-the-art LLMs in single- and multi-agent systems, simultaneously with Ph.D. experts. The best single-agent performance (%) is highlighted in blue, and the best multi-agent performance (%) in purple. We also report the average number of reasoning steps and reasoning efficiency (%), reflecting the tool usage of agents.

Base LLM	Slow Thinking	Open Source	Multiple Choice			Concise Answer			Open Answer			Avg. Score	Avg. Eff. (%)	Avg. Steps
			Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard			
PaperArena-Hub (Single-Agent)														
Gemini 2.5 Pro [12]	✓	✗	57.89	40.98	22.22	58.49	40.00	21.95	56.52	36.07	13.57	36.10	41.20	9.81
OpenAI o4-mini-high [37]	✓	✗	57.89	44.26	22.22	55.66	37.30	14.63	52.17	33.61	12.86	33.93	35.12	10.63
Claude Sonnet 4 [4]	✓	✗	60.53	37.70	18.52	53.77	38.92	19.51	52.17	31.97	15.00	34.18	37.34	9.55
GPT-4.1 [36]	✗	✗	55.26	34.43	14.81	49.06	35.68	17.07	52.17	28.69	10.71	30.61	35.26	8.27
Claude 3.5 Sonnet [3]	✗	✗	47.37	28.23	11.11	48.11	25.95	10.98	43.48	22.95	7.14	24.62	21.98	7.89
Qwen3-235B-Thinking [59]	✓	✓	52.63	32.79	14.81	53.77	33.51	15.85	47.83	27.87	10.00	29.97	31.21	10.20
GLM-4.5 [62]	✓	✓	47.37	29.51	11.11	49.06	30.27	13.41	43.48	27.05	8.57	27.17	25.13	11.82
Qwen3-235B-Instruct [59]	✗	✓	44.74	24.59	11.11	46.23	24.32	10.98	39.13	22.13	7.14	23.47	36.85	7.24
Kimi-K2-Instruct [51]	✗	✓	44.74	22.95	11.11	45.28	22.70	9.76	39.13	20.49	6.43	22.32	23.57	12.86
PaperArena-Hub (Multi-Agent)														
Gemini 2.5 Pro [12]	✓	✗	60.53	42.62	33.33	63.21	42.70	21.95	65.22	39.34	13.57	38.78	45.39	8.58
OpenAI o4-mini-high [37]	✓	✗	60.53	41.00	25.93	62.26	41.08	21.95	56.52	36.89	14.29	37.37	43.41	8.30
Claude Sonnet 4 [4]	✓	✗	68.42	39.34	18.52	61.32	40.00	23.17	56.52	36.07	12.86	36.73	39.13	8.92
GPT-4.1 [36]	✗	✗	57.89	37.70	18.52	59.43	38.38	19.51	52.17	34.43	12.14	34.57	34.35	7.94
Claude 3.5 Sonnet [3]	✗	✗	50.00	31.15	14.81	52.83	32.43	14.63	47.83	29.51	9.29	29.34	23.67	8.16
Qwen3-235B-Thinking [59]	✓	✓	57.89	37.70	18.52	59.43	37.84	19.51	52.17	33.61	12.14	34.31	33.79	10.48
GLM-4.5 [62]	✓	✓	52.63	32.79	14.81	54.72	34.05	15.85	47.83	30.33	10.71	30.74	27.01	10.90
Qwen3-235B-Instruct [59]	✗	✓	50.00	31.15	14.81	51.89	31.35	14.63	47.83	28.69	9.29	28.83	38.13	7.82
Kimi-K2-Instruct [51]	✗	✓	50.00	27.87	11.11	50.00	28.11	12.20	43.48	25.41	7.86	26.28	29.45	11.94
Human Baseline														
Ph.D. Experts	-	-	87.50	84.21	71.43	91.67	85.42	68.18	87.50	86.67	82.35	83.50	75.48	6.52

all papers on a topic from a specific conference and year, or extracting relevant entries based on keyword queries. More broadly, we implement standardized interfaces for all tools and incorporate robust error handling to minimize execution failures. This ensures reliable performance during long tool chains and enables accurate measurement of tool-use effectiveness across different agents.

4.3 Benchmark Evaluation

We evaluate nine state-of-the-art LLMs in both single-agent and multi-agent systems: Gemini 2.5 Pro [12], OpenAI o4-mini-high [37], Claude Sonnet 4 [4], GPT-4.1 [36], Claude 3.5 Sonnet [3], Qwen3-235B-Thinking [59], GLM-4.5 [62], Qwen3-235B-Instruct [59], and Kimi-K2-Instruct [51]. We construct a 200-question subset from PaperArena (PaperArena-Human) and recruit three computer science Ph.D. candidates to complete it using standard tools such as GPT-4o and web search engines. All answers both from LLMs and human experts are evaluated using an LLM-as-a-Judge protocol [32], where GPT-4o assigns binary correctness labels. To emphasize reasoning over surface form [33, 53], we allow flexible output styles as long as the reasoning is valid and the final answer is unambiguous. In addition to correctness, we measure agent behavior using two metrics: (1) *average reasoning steps*, defined as the mean length of executed tool chains; and (2) *average reasoning efficiency*, computed as the average overlap between the agent's executed tool chain and the predefined chain, normalized by the executed length. To ensure evaluation reliability, we manually verify all human answers and a sample of LLM responses within the human subset.

5 Experiments

In this section, we evaluate the overall performance of the agents and present three key observations regarding their reasoning processes. Finally, we conduct in-depth analysis of agent capabilities, error patterns, and human evaluation.

5.1 Experimental Results

Table 2 summarizes the performance of nine leading LLMs under both single-agent and multi-agent systems. Gemini 2.5 Pro achieves the best overall results in both settings, yet a substantial performance gap persists between all LLM-based agents and the Ph.D. expert baseline. Across all settings, performance consistently degrades from easy to hard tasks and from multiple-choice to open-ended formats, reflecting the growing challenges posed by increased reasoning and synthesis requirements. Moreover, closed-source LLMs generally outperform open-source counterparts across most metrics. Slow-thinking LLMs (with long chain-of-thought) achieve higher accuracy than fast-thinking ones, likely due to their implicit task decomposition and tool planning during thinking. Notably, the multi-agent system typically yields improvements in both accuracy and tool-use efficiency, suggesting that agent collaboration supports more effective coordination and division of reasoning labor. However, most agents still exhibit poor planning: they invoke tools far more often than needed, leading to low average efficiency. This inefficiency highlights fundamental limitations in current LLMs' ability in multi-step reasoning, tool selection, and execution control in complex scientific problem-solving.

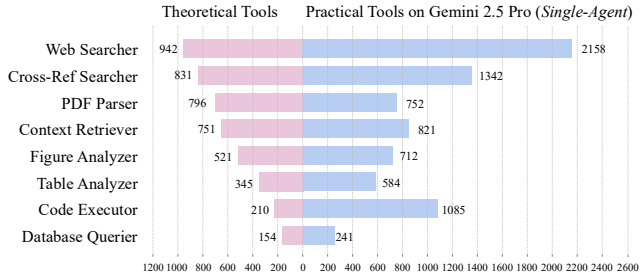

Figure 5: Comparison of theoretical and practical number of tool calls by Gemini 2.5 Pro on the single-agent system, which reveals the far greater number of practical calls.

5.2 Observation of Reasoning Trace

We observe that agents exhibit strong biases in tool usage, often favoring general-purpose tools. Moreover, although harder tasks require more tool invocations, this does not consistently improve performance. Instead, effective tool planning, rather than sheer tool quantity, plays a critical role in solving complex problems.

Observation 1: The agent's tool usage is strongly imbalanced, showing a clear preference for general-purpose tools like Web Searcher and Code Executor over more specialized ones. Figure 5, which details the gap between practical and theoretical number of tool calls, clearly illustrates this phenomenon. This biased preference is a primary driver of the significant efficiency gap between practical actions and the theoretical optimum. For instance, the practical use of the Code Executor is over five times its theoretical requirement, suggesting the agent defaults to a brute-force, trial-and-error coding approach. Similarly, the Web Searcher's practical usage is more than double its theoretical baseline, indicating that the agent's imbalanced strategy results in inefficient, exploratory searches rather than precise queries. In contrast, for deterministic tasks requiring a specific, non-preferred tool like the PDF Parser, the practical and theoretical calls are nearly identical. This suggests that the agent's inefficiency is not a universal failure of its capabilities, but rather a direct consequence of its imbalanced tool selection strategy.

Observation 2: Agents tend to employ more reasoning steps for more difficult problems requiring more tools, yet this fails to avert a decline in performance. The task difficulty distribution in Figure 6 confirms this, showing a clear positive correlation between the variety of tools required in theory and the number of reasoning steps taken. Moreover, harder tasks consistently require longer and more complex reasoning steps. However, this increased effort does not prevent a drop in performance. As shown in Figure 7, the scores of both single-agent and multi-agent systems decline as task complexity increases, suggesting that the ability to execute complex plans diminishes even as the agent expends more effort. Within this trend, multi-agent systems consistently outperform single-agent systems on complex, multi-tool tasks. This enhanced performance underscores the critical planning capability of the manager agent within a centralized design. For multi-step problems, although a single agent may execute localized planning, the comprehensive synthesis and coordination of a multi-agent architecture are essential for success.

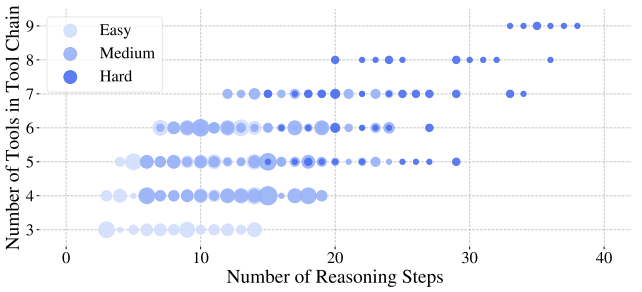

Figure 6: Task difficulty distribution in the benchmark, showing a clear positive correlation between the variety of tools required in theory and the number of reasoning steps taken.

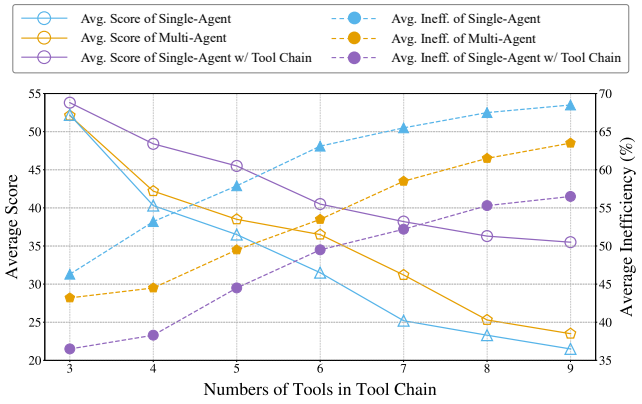

Figure 7: Performance and efficiency analysis of different agent systems. Average Inefficiency is defined as Average Ineffiency = 1 − Average Effiency, where a higher value indicates less efficient tool use.

Observation 3: The single agent, when explicitly provided with the globally optimal tool chain in its prompt, outperforms multi-agent systems guided by a planning manager. This suggests that the endogenous planning capabilities of current LLMs may be susceptible to local optima and warrant further investigation. We conduct this experiment by adding the theoretical tool chain directly into the agent's prompt during its reasoning process. As shown in Figure 7, the agent with a predefined tool chain establishes a clear performance ceiling based on near-perfect planning. The multi-agent system, while effective, operates below this ceiling, and the gap between them quantifies the performance loss from its suboptimal, endogenously generated plans. The efficiency analysis further illuminates this: while the multi-agent's planning leads to a higher efficiency rate than the standard single agent, its rate is still significantly lower than that of the agent given a perfect plan. Crucially, even the agent with the predefined tool chain shows a rising inefficiency rate. This reveals a second distinct failure mode: flawed tool invocation. Even with a perfect plan, agents can fail to use tools correctly (e.g., by providing incorrect parameters), especially as complexity increases. Therefore, overall performance degradation is a composite of both suboptimal planning and flawed tool invocation, revealing possible directions for improving agents.

5.3 In-Depth Analysis

Our in-depth analysis reveals the agent's nuanced capabilities by pinpointing individual model strengths and its upper scaling limits at test time, while also diagnosing key error sources, verifying evaluation reliability, and validating the consistency of LLM-as-a-Judge with human evaluation to provide a truly comprehensive understanding beyond aggregate metrics.

5.3.1 Analysis of Fine-Grained LLM Capabilities. To further probe the specific strengths and weaknesses of the top-performing models, we categorized the benchmark questions into four primary ability areas in Table 3: web browsing (Browsing), code execution (Coding), multimodal understanding (Multi-Modality), and complex reasoning chains (Multi-Steps). The performance breakdown, presented in the table, reveals that no single LLM universally dominates across all capabilities. Gemini 2.5 Pro demonstrates superior proficiency in browsing and multi-modality task, showcasing its strength in information retrieval and visual data interpretation. In contrast, Claude Sonnet 4 emerges as the standout performer in coding tasks, significantly surpassing its competitors in this domain. In contrast, Qwen3-235B, regardless of whether it is a slow-thinking LLM or a fast-thinking one, performs very poorly on multimodal understanding tasks because they do not have visual capability, showing the importance of multimodal perception for agents. This fine-grained analysis highlights the nuanced and often complementary strengths of state-of-the-art LLMs. This insight suggests that a promising direction for building more powerful and versatile agents is to develop heterogeneous systems that can strategically leverage the distinct advantages of different specialized LLMs for appropriate sub-tasks, rather than relying on a single one.

Table 3: Fine-grained performance of five leading LLMs on the single-agent system, revealing their specialized strengths and weaknesses across the four primary capabilities required in questions on PaperArena benchmark.

Base LLM	Browsing	Coding	Multi-Modality	Multi-Steps
Gemini 2.5 Pro	40.61	33.27	40.03	32.85
Claude Sonnet 4	36.68	41.12	32.59	31.30
GPT-4.1	27.54	35.26	29.81	25.58
Qwen3-Think.	31.22	32.85	12.54	22.39
Qwen3-Inst.	28.70	25.58	11.96	20.13

5.3.2 Analysis of Pass@k Performance. We evaluate agent test-time scaling in a pass@k scenario, where each task is executed k times under identical parameters and the final result is determined by a majority vote. As shown in Figure 8, our analysis reveals a clear positive correlation between the number of attempts and agent performance, a trend consistent across all models. Notably, this scaling effect allows fast-thinking models to match or even surpass the single-pass performance of slow-thinking ones. However, this approach introduces a significant challenge: a naive parallel implementation leads to a nearly linear cost growth with respect to k. Compounding this issue, we find that the performance gains exhibit diminishing returns; the improvement is substantial for small values of k but becomes marginal as k increases. These dual findings highlight the central problem of achieving an optimal trade-off between performance gains and evaluation cost. Future work can thus focus on designing agents with larger token budgets under controlled cost constraints and developing more efficient parallelization strategies that move beyond simple majority voting.

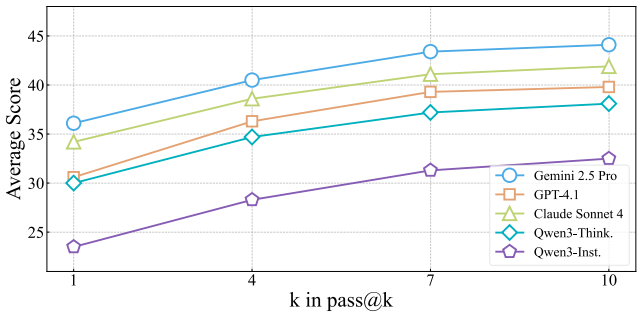

Figure 8: The performance in the pass@k evaluation reveals the limits of the agent's test-time scaling capabilities on PaperArena. By trying multiple times and taking consistent answers, the agent can achieve better performance.

5.3.3 Analysis of Error Attribution. To understand agent inefficiency, we conduct an error attribution analysis. We perform this experiment by inspecting tool outputs, error messages, and the agent's subsequent actions to attribute each failure. We then categorize these issues into four types: improperly formatted inputs (Parameter Error), actions that violate task dependencies (Logical Error), unnecessary calls for existing information (Redundant Action), and external tool execution problems (Tool Failure). Our analysis in Table 4 reveals distinct failure patterns. The low rate of tool failure across all LLMs suggests the tools themselves are reliable and not the primary performance bottleneck. Open-source LLMs exhibit more parameter and logical errors, indicating weaknesses in complex planning and reflection. In contrast, closed-source slow-thinking LLMs tend toward redundant actions, suggesting a form of cognitive overthinking where agents needlessly re-verify information they already possess. These findings inform future agent design. To mitigate these errors, agents could incorporate explicit memory modules that record the reasoning chain to guide subsequent actions and prevent repetition. Enhancing self-correction mechanisms and robust error handling would also increase overall task success and efficiency.

Table 4: Error attribution analysis of the primary failure modes for five leading LLMs on the single-agent system. This table present the proportion of each error type out of the total observed failures for that specific LLM.

Base LLM	Parameter Error	Logical Error	Redundant Action	Tool Failure
Gemini 2.5 Pro	23.71	23.28	38.35	14.82
GPT-4.1	22.69	22.90	39.14	15.46
Claude Sonnet 4	21.83	25.17	36.91	16.25
Qwen3-Think.	35.40	30.88	20.13	13.79
Qwen3-Inst.	38.96	28.12	18.77	14.34

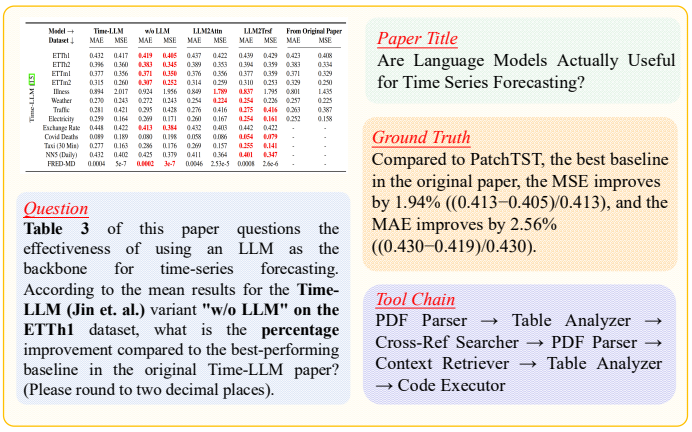

Figure 9: Case study of QA examples in PaperArena that challenge the agent's core capabilities in multi-step reasoning and information synthesis. To derive the correct answer, the agent must integrate data from diverse formats and sources.

Table 5: Consistency between GPT-4o and Human Judgments on PaperArena-Human Evaluations.

Metrics	Gemini 2.5 Pro	Ph.D. Experts
Agreement Rate (%)	98.5	98.0
Cohen's Kappa (κ)	0.97	0.93

5.3.4 Analysis of Evaluator Consistency. Assessing agent performance in complex scientific domains presents a significant evaluation challenge, often demanding costly and time-consuming human expert judgment. To develop a scalable alternative, we evaluate the consistency of GPT-4o as an automatic evaluator. We compare its binary correctness labels against independent human scoring on the PaperArena-Human subset, using both simple agreement rates and Cohen's Kappa (κ) to measure inter-rater reliability while correcting for chance agreement [50]. As shown in Table 5, GPT-4o's judgments exhibit exceptionally high consistency with those of human evaluators. When evaluating Gemini 2.5 Pro's responses, the agreement rate reaches 98.5% ($\kappa = 0.97$). Similarly, for a Ph.D. expert answers, the agreement is 98.0% ($\kappa = 0.93$). The Cohen's Kappa values are particularly compelling, as scores above 0.81 are typically interpreted as almost perfect agreement. This level of alignment, with discrepancies observed in at most three samples per category, underscores the stability and predictability of GPT-4o's evaluation standard. These results strongly indicate that GPT-4o functions as a dependable and consistent judge, providing a reproducible and scalable method for evaluating complex agent reasoning and mitigating the bottlenecks associated with manual expert assessment.

5.4 Case Study

To illustrate the challenges and capabilities involved in multi-step, cross-modal scientific reasoning, we present two representative case studies in Figure 9. Each example showcases a real-world question grounded in a scientific paper and requires agents to coordinate multiple tools such as PDF parsers, table analyzers, figure analyzers, and code executors to derive an accurate answer. The first case involves numerical comparison of model variants across tabular results, requiring fine-grained metric extraction, citation tracking, and percentage calculation. The second case centers on interpreting bar charts to assess performance improvements in meta-optimization, demanding precise visual parsing and numerical estimation. Both cases include ground-truth answers and complete tool chains, enabling systematic evaluation of intermediate reasoning steps. These examples highlight the complexity of aligning textual, tabular, and visual evidence, and underscore the need for efficient tool orchestration, reliable data interpretation, and precise numerical reasoning in automated scientific literature understanding.

6 Conclusion and Future Work

In this work, we proposed PaperArena, a novel benchmark designed to evaluate tool-augmented agentic reasoning on scientific literature. Unlike existing QA benchmarks that focus on isolated paragraphs or simple fact retrieval, PaperArena requires multi-step, multimodal, and cross-document reasoning grounded in real research questions. To enable standardized, reproducible evaluation, we develop PaperArena-Hub, an extensible platform that supports both single-agent and multi-agent system, equipped with a comprehensive tool suite such as PDF parsing, table and figure analysis and code execution. Our extensive experiments on nine state-of-the-art LLMs with mature agent systems reveal a persistent performance gap compared to human experts, especially the hard subset. Most agents still struggle with ineffcient tool usage and sub-optimal planning, highlighting the challenges of end-to-end scientific reasoning and tool invocation in current agent systems.

PaperArena and PaperArena-Hub provide a foundation for future research. Researchers can easily evaluate novel agent strategies, test custom tools using our modular interface, or extend the benchmark with new domains. Future directions include enhancing the question generation pipeline, expanding the benchmark beyond AI to other scientific disciplines, and integrating new metrics for tool rationale, planning transparency, or interpretability. We invite the community to adopt PaperArena and PaperArena-Hub to design more capable and interpretable agents for scientific discovery.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691* (2022).

[3] Anthropic. 2024. *Claude 3.5 Sonnet Model Card Addendum.* Technical Report. Anthropic. https://www-cdn.anthropic.com/fed9cc193a14b84131812372d8d5857f8f304c52/Model_Card_Claude_3_Addendum.pdf

[4] Anthropic. 2025. *System Card: Claude Opus 4 & Claude Sonnet 4.* Technical Report. Anthropic. https://www-cdn.anthropic.com/4263b940cabb546aa0e3283f35b686f4f3b2ff47.pdf

[5] Sören Auer, Dante AC Barone, Cassiano Bartz, Eduardo G Cortes, Mohamad Yaser Jaradeh, Oliver Karras, Manolis Koubarakis, Dmitry Mouromtsev, Dmitrii Pliukhin, Daniil Radyush, et al. 2023. The sciqa scientific question answering benchmark for scholarly knowledge. *Scientific Reports* 13, 1 (2023), 7240.

[6] Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. *arXiv preprint arXiv:1704.02853* (2017).

[7] Tim Baumgärtner, Ted Briscoe, and Iryna Gurevych. 2025. PeerQA: A Scientific Question Answering Dataset from Peer Reviews. *arXiv preprint arXiv:2502.13668* (2025).

[8] Kurt D Bollacker, Steve Lawrence, and C Lee Giles. 2002. Discovering relevant scientific literature on the web. *IEEE Intelligent Systems and their Applications* 15, 2 (2002), 42–47.

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[10] Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. 2024. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. *arXiv preprint arXiv:2410.05080* (2024).

[11] Mingyue Cheng, Yucong Luo, Jie Ouyang, Qi Liu, Huijie Liu, Li Li, Shuo Yu, Bohou Zhang, Jiawei Cao, Jie Ma, et al. 2025. A survey on knowledge-oriented retrieval-augmented generation. *arXiv preprint arXiv:2503.10677* (2025).

[12] George Comanici, Elad Bieber, Mike Schaekermann, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261* (2025). arXiv:2507.06261

[13] Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. *arXiv preprint arXiv:2105.03011* (2021).

[14] Shanghua Gao, Richard Zhu, Pengwei Sui, Zhenglun Kong, Sufian Aldogom, Yepeng Huang, Ayush Noori, Reza Shamji, Krishna Parvataneni, Theodoros Tsiligkaridis, et al. 2025. Democratizing AI scientists using ToolUniverse. *arXiv preprint arXiv:2509.23426* (2025).

[15] Taicheng Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh Chawla, Olaf Wiest, Xiangliang Zhang, et al. 2023. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in Neural Information Processing Systems* 36 (2023), 59662–59688.

[16] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300* (2020).

[17] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. 2024. MetaGPT: Meta programming for a multi-agent collaborative framework. International Conference on Learning Representations, ICLR.

[18] Antoine Houssard, Floriana Gargiulo, Tommaso Venturini, Paola Tubaro, and Gabriele Di Bona. 2024. The Gerontocratization of Science: How hypergrowth reshapes knowledge circulation. *arXiv preprint arXiv:2410.00788* (2024).

[19] Tianyu Hua, Harper Hua, Violet Xiang, Benjamin Klieger, Sang T Truong, Weixin Liang, Fan-Yun Sun, and Nick Haber. 2025. ResearchCodeBench: Benchmarking LLMs on Implementing Novel Machine Learning Research Code. *arXiv preprint arXiv:2506.02314* (2025).

[20] Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024. Understanding the planning of LLM agents: A survey. *arXiv preprint arXiv:2402.02716* (2024).

[21] Chuang Jiang, Mingyue Cheng, Xiaoyu Tao, Qingyang Mao, Jie Ouyang, and Qi Liu. 2025. TableMind: An Autonomous Programmatic Agent for Tool-Augmented Table Reasoning. *arXiv preprint arXiv:2509.06278* (2025).

[22] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146* (2019).

[23] Ruofan Jin, Zaixi Zhang, Mengdi Wang, and Le Cong. 2025. STELLA: Self-Evolving LLM Agent for Biomedical Research. *arXiv preprint arXiv:2507.02004* (2025).

[24] Uri Katz, Mosh Levy, and Yoav Goldberg. 2024. Knowledge Navigator: LLM-guided Browsing Framework for Exploratory Search in Scientific Literature. 8838–8855. doi:10.18653/v1/2024.findings-emnlp.516

[25] Olga Kononova, Tanjin He, Haoyan Huo, Amalie Trewartha, Elsa A Olivetti, and Gerbrand Ceder. 2021. Opportunities and challenges of text mining in materials research. *Iscience* 24, 3 (2021).

[26] Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. 2023. BioASQ-QA: A manually curated corpus for Biomedical Question Answering. *Scientific Data* 10, 1 (2023), 170.

[27] Jakub Lála, Odhran O'Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G Rodriques, and Andrew D White. 2023. Paperqa: Retrieval-augmented generative agent for scientific research. *arXiv preprint arXiv:2312.07559* (2023).

[28] Jon M Laurent, Joseph D Janizek, Michael Ruzo, Michaela M Hinks, Michael J Hammerling, Siddharth Narayanan, Manvitha Ponnapati, Andrew D White, and Samuel G Rodriques. 2024. Lab-bench: Measuring capabilities of language models for biology research. *arXiv preprint arXiv:2407.10362* (2024).

[29] Yoonjoo Lee, Kyungjae Lee, Sunghyun Park, Dasol Hwang, Jaehyeon Kim, Hong-in Lee, and Moontae Lee. 2023. Qasa: advanced question answering on scientific articles. In *International Conference on Machine Learning*. PMLR, 19036–19052.

[30] Qingchuan Li, Jiatong Li, Zirui Liu, Mingyue Cheng, Yuting Zeng, Qi Liu, and Tongxuan Liu. 2025. Are LLMs Reliable Translators of Logical Reasoning Across Lexically Diversified Contexts? *arXiv preprint arXiv:2506.04575* (2025).

[31] Zhiyu Li, Shichao Song, Chenyang Xi, Hanyu Wang, Chen Tang, Simin Niu, Ding Chen, Jiawei Yang, Chunyu Li, Qingchen Yu, et al. 2025. Memos: A memory os for ai system. *arXiv preprint arXiv:2507.03724* (2025).

[32] Zirui Liu, Jiatong Li, Yan Zhuang, Qi Liu, Shuanghong Shen, Jie Ouyang, Mingyue Cheng, and Shijin Wang. 2025. am-ELO: A Stable Framework for Arena-based LLM Evaluation. *arXiv preprint arXiv:2505.03475* (2025).

[33] Yucong Luo, Yitong Zhou, Mingyue Cheng, Jiahao Wang, Daoyu Wang, Tingyue Pan, and Jintao Zhang. 2025. Time Series Forecasting as Reasoning: A Slow-Thinking Approach with Reinforced LLMs. *arXiv preprint arXiv:2506.10630* (2025).

[34] Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. 2024. Discoverybench: Towards data-driven discovery with large language models. *arXiv preprint arXiv:2407.01725* (2024).

[35] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332* (2021).

[36] OpenAI. 2025. *Introducing GPT-4.1 in the API.* https://openai.com/index/gpt-4-1/

[37] OpenAI. 2025. *OpenAI o3 and o4-mini System Card.* Technical Report. OpenAI. https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf

[38] Jie Ouyang, Tingyue Pan, Mingyue Cheng, Ruiran Yan, Yucong Luo, Jiaying Lin, and Qi Liu. 2025. Hoh: A dynamic benchmark for evaluating the impact of outdated information on retrieval-augmented generation. *arXiv preprint arXiv:2503.04800* (2025).

[39] Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng. 2018. emrqa: A large corpus for question answering on electronic medical records. *arXiv preprint arXiv:1809.00732* (2018).

[40] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249* (2025).

[41] Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. 2024. Spiqa: A dataset for multimodal question answering on scientific papers. *Advances in Neural Information Processing Systems* 37 (2024), 118807–118833.

[42] Tingrui Qiao, Caroline Walker, Chris Cunningham, and Yun Sing Koh. 2025. Thematic-LM: a LLM-based multi-agent system for large-scale thematic analysis. In *Proceedings of the ACM on Web Conference 2025*. 649–658.

[43] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789* (2023).

[44] Jiahao Qiu, Xuan Qi, Tongcheng Zhang, Xinzhe Juan, Jiacheng Guo, Yifu Lu, Yimin Wang, Zixin Yao, Qihan Ren, Xun Jiang, et al. 2025. Alita: Generalist agent enabling scalable agentic reasoning with minimal predefinition and maximal self-evolution. *arXiv preprint arXiv:2505.20286* (2025).

[45] Federico Ruggeri, Mohsen Mesgar, and Iryna Gurevych. 2023. A dataset of argumentative dialogues on scientific papers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 7684–7699.

[46] Tanik Saikh, Tirthankar Ghosal, Amish Mittal, Asif Ekbal, and Pushpak Bhattacharyya. 2022. Scienceqa: A novel resource for question answering on scholarly articles. *International Journal on Digital Libraries* 23, 3 (2022), 289–301.

[47] Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning, 2023. *URL https://arxiv. org/abs/2303.11366* 1 (2023).
[48] Michael D Skarlinski, Sam Cox, Jon M Laurent, James D Braza, Michaela Hinks, Michael J Hammerling, Manvitha Ponnapati, Samuel G Rodriques, and Andrew D White. 2024. Language agents achieve superhuman synthesis of scientific knowledge. *arXiv preprint arXiv:2409.13740* (2024).
[49] Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, et al. 2025. PaperBench: Evaluating AI's Ability to Replicate AI Research. *arXiv preprint arXiv:2504.01848* (2025).
[50] Shuyan Sun. 2011. Meta-analysis of Cohen's kappa. *Health Services and Outcomes Research Methodology* 11, 3 (2011), 145–163.
[51] Team K, Y. Bai, Y. Bao, et al. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534* (2025). arXiv:2507.20534
[52] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. *arXiv preprint arXiv:2004.14974* (2020).
[53] Jiahao Wang, Mingyue Cheng, and Qi Liu. 2025. Can slow-thinking llms reason over time? empirical studies in time series forecasting. *arXiv preprint arXiv:2505.24511* (2025).
[54] Serena Wang, Michael Jordan, Katrina Ligett, and R Preston McAfee. 2025. Relying on the Metrics of Evaluated Agents. In *Proceedings of the ACM on Web Conference 2025*. 1468–1487.
[55] Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. 2024. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems* 37 (2024), 113569–113697.
[56] Zihan Wang, Ziqi Zhao, Yougang Lyu, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. 2025. A cooperative multi-agent framework for zero-shot named entity recognition. In *Proceedings of the ACM on Web Conference 2025*. 4183–4195.
[57] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.
[58] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. 2024. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*.
[59] A. Yang, A. Li, B. Yang, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388* (2025). arXiv:2505.09388
[60] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.
[61] Shuo Yu, Mingyue Cheng, Jiqian Yang, Jie Ouyang, Yucong Luo, Chenyi Lei, Qi Liu, and Enhong Chen. 2024. Multi-source knowledge pruning for retrieval-augmented generation: A benchmark and empirical study. *arXiv preprint arXiv:2409.13694* (2024).
[62] A. Zeng, X. Lv, Q. Zheng, et al. 2025. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471* (2025). arXiv:2508.06471
[63] Yaolun Zhang, Xiaogeng Liu, and Chaowei Xiao. 2025. MetaAgent: Automatically Constructing Multi-Agent Systems Based on Finite State Machines. *arXiv preprint arXiv:2507.22606* (2025).
[64] Yu Zhang, Yanzhen Shen, SeongKu Kang, Xiusi Chen, Bowen Jin, and Jiawei Han. 2025. Chain-of-factors paper-reviewer matching. In *Proceedings of the ACM on Web Conference 2025*. 1901–1910.
[65] Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024. A survey on the memory mechanism of large language model based agents. *ACM Transactions on Information Systems* (2024).
[66] Yitong Zhou, Mingyue Cheng, Qingyang Mao, Yucong Luo, Qi Liu, Yupeng Li, Xiaohan Zhang, Deguang Liu, Xin Li, and Enhong Chen. 2025. Benchmarking Multimodal LLMs on Recognition and Understanding over Chemical Tables. *arXiv preprint arXiv:2506.11375* (2025).

A Ethical Statement

All source papers were retrieved from public, open-access platforms using unbiased, automated sampling. The dataset is intended exclusively for non-commercial academic research, and all generated artifacts are publicly released under an open license. The benchmark is distributed under a non-commercial license. Users must cite this work and comply with the original papers' licensing terms. Any commercial use is strictly prohibited.

B Benchmark Construction Details

B.1 Paper Corpus Preparation

Our benchmark is constructed from a curated corpus of $N = 14{,}435$ open-access latest AI research papers collected from OpenReview and Open Access, all published in 2025 to minimize potential leakage from LLM pre-training data. Each paper P_i is annotated using GPT-4o-mini and mapped to a 20-dimensional binary vector $\mathbf{v}_i \in \{0,1\}^{20}$, encoding its influence, research field, method and evaluation type. Figure 2 visualizes this representation via t-SNE, highlighting a strong clustering effect around dominant topics.

To prevent this concentration from biasing our benchmark, we develop a hierarchical sampling strategy. First, we select $K = 50$ representative medoids by applying K-Medoids clustering:

$$\mathcal{P}_{\text{prototype}} = \underset{\{P_{c_1},\ldots,P_{c_K}\}\subset\mathcal{P}}{\arg\min} \sum_{i=1}^{N} \min_{k\in\{1,\ldots,K\}} d(\mathbf{v}_i, \mathbf{v}_{c_k}). \tag{1}$$

To enhance diversity, we further apply Farthest Point Sampling (FPS) to iteratively select $L = 50$ boundary papers:

$$P_{b_i} = \underset{P_j\in\mathcal{P}\setminus\mathcal{Q}_{i-1}}{\arg\max} \left(\min_{P_s\in\mathcal{Q}_{i-1}} d(\mathbf{v}_j, \mathbf{v}_s) \right), \tag{2}$$

where $\mathcal{Q}_{i-1} = \mathcal{P}_{\text{prototype}} \cup \{P_{b_1}, \ldots, P_{b_{i-1}}\}$. We use Hamming distance $d(\cdot,\cdot)$ to measure dissimilarity. The final benchmark subset $\mathcal{P}_{\text{sub}} = \mathcal{P}_{\text{prototype}} \cup \mathcal{P}_{\text{boundary}}$ thus includes 100 papers with high representativeness and topical diversity.

B.2 QA Pair Generation Process

We construct QA pairs through a structured three-stage pipeline involving tool library definition, heuristic generation, and semi-automated verification.

Tool Library. We define a comprehensive toolset $\mathcal{T}$, enabling agents to interact with scientific literature in a flexible manner:

$$\mathcal{T} = \{\mathcal{T}_{\text{pdf}}, \mathcal{T}_{\text{tab}}, \mathcal{T}_{\text{fig}}, \mathcal{T}_{\text{ref}}, \mathcal{T}_{\text{web}}, \mathcal{T}_{\text{rag}}, \mathcal{T}_{\text{db}}, \mathcal{T}_{\text{code}}\}. \tag{3}$$

Heuristic QA Generation. Guided by a tool chain experience pool $\mathcal{E} = \{\tau_1, \tau_2, \ldots\}$, we use prompt in Table 9 together with Gemini 2.5 Pro ($\mathcal{G}$) to generate QA triplets (q_k, τ_k, a_k). Each paper is parsed via MinerU into sections $\{S_1, S_2, \ldots\}$. The generation process is:

$$\{(q_k, \tau_k, a_k)\}_{k=1}^{N_{\text{gen}}} = \bigcup_{i,j} \mathcal{G}(D_i, S_j \mid \mathcal{E}). \tag{4}$$

For example, a question might be: *"Regarding Table 2, the paper X claims superior performance over Liu et al. (2024). Please locate the result in paper Y and verify this claim."* This requires the tool chain $\tau = \mathcal{T}_{\text{pdf}}(X) \to \mathcal{T}_{\text{tab}}(\text{Tab. 2}) \to \mathcal{T}_{\text{ref}}(\text{Liu et al.}) \to \mathcal{T}_{\text{web}}(Y) \to \mathcal{T}_{\text{rag}}(Y) \to \mathcal{T}_{\text{tab}}(\text{Tab. } z)$.

Semi-Automated Verification. For each candidate QA pair, we execute τ_k to retrieve intermediate outputs and assess consistency. We apply two refinements: (1) *question obfuscation*, e.g., removing "According to the abstract..." to increase difficulty; and (2) *answer standardization*, such as enforcing consistent formats and generating distractors. New tool chains discovered during this step are added to $\mathcal{E}$, forming a virtuous refinement loop. After human verification, we retain 784 high-quality QA triplets.

B.3 Dataset Statistics

We analyze our benchmark from two perspectives: question format with difficulty, and the primary capabilities required.

Question Type and Difficulty. The benchmark includes three question formats: Multiple Choice (MC), Concise Answer (CA), and Open Answer (OA). Each question is assigned a difficulty level of Easy, Medium, or Hard. Table 6 shows the distribution for both the full benchmark, which we call PaperArena (784 questions), and a 200-question subset reserved for human evaluation, PaperArena-Human. The distribution highlights a focus on Medium difficulty questions and a substantial number of open-ended questions to rigorously test generative capabilities.

Table 6: Distribution of question types and difficulty levels in PaperArena and PaperArena-Human. MC: Multiple Choice, CA: Concise Answer, OA: Open Answer.

Difficulty	PaperArena				PaperArena-Human			
	MC	CA	OA	Total	MC	CA	OA	Total
Easy	38	106	23	167	8	24	8	40
Medium	61	185	122	368	19	48	30	97
Hard	27	82	140	249	7	22	34	63
Total	126	373	285	784	34	94	72	200

Required Agent Capabilities. We also categorize questions based on the primary capability an agent must demonstrate. As shown in Table 7, these categories are: Browsing, which requires searching the web or databases; Coding, for solving numerical problems; Multi-Modality, for interpreting figures and tables; and Multi-Steps, for complex multi-step reasoning. The distribution ensures balanced evaluation across these diverse and critical capacity dimensions.

Table 7: Distribution of questions based on the primary required agent capability.

Browsing	Coding	Multi-Modality	Multi-Steps	Total
194	159	261	170	784

C Evaluation Details on PaperArena-Hub

C.1 Tool Library Implementation

To facilitate agent interaction with scientific literature and external knowledge sources, we implement a comprehensive suite of tools on the PaperArena-Hub. Each tool is designed with specific inputs, outputs, parameters and descriptions to perform distinct tasks. Table 8 provides a detailed overview of the tool library.

C.2 Evaluation Setup

We evaluate nine state-of-the-art LLMs: Gemini-2.5-Pro-2506, OpenAI-o4-mini-high-2504, Claude-Sonnet-4-2505, GPT-4.1-2504, Claude-3.5-Sonnet-2410, Qwen3-235B-Thinking-2507, GLM-4.5-2507, Qwen3-235B-Instruct-2507, and Kimi-K2-Instruct-2507. For all agent-based evaluations, we use a consistent set of inference parameters: the temperature of 0.7, top_p of 0.9, a maximum of 4096 output tokens, and a maximum of 40 reasoning steps per question.

To establish a human expert baseline, we recruit three computer science Ph.D. candidates to complete the PaperArena-Human subset. The evaluation is conducted under controlled conditions: participants are allotted 16 hours over two days and are permitted to use standard tools, including GPT-4o for document comprehension, web search for information retrieval, and Python for coding tasks.

We assess performance using three primary metrics:

Correctness. We employ an LLM-as-a-Judge protocol where GPT-4o assigns a binary score $S_{\text{corr}} \in \{0, 1\}$ to each answer. An answer is deemed correct if its final conclusion is accurate and supported by valid reasoning, regardless of stylistic format.

Reasoning Steps. To measure solution complexity, we compute the average number of tool calls in an agent's executed tool chain, τ^{exec}. For a set of questions Q, the average reasoning steps are:

$$\text{Average Steps} = \frac{1}{|Q|} \sum_{q \in Q} |\tau_q^{\text{exec}}|. \tag{5}$$

Reasoning Efficiency. To assess how concisely an agent solves a problem, we measure the overlap between the executed tool chain τ^{exec} and the ground-truth chain τ^{gt}, normalized by the length of the executed chain:

$$\text{Average Efficiency} = \frac{1}{|Q|} \sum_{q \in Q} \frac{|\tau_q^{\text{exec}} \cap \tau_q^{\text{gt}}|}{|\tau_q^{\text{exec}}|}. \tag{6}$$

To ensure reliability, we manually verify all human answers and a sample of LLM responses on the PaperArena-Human subset.

C.3 Analysis of Reasoning Errors

Beyond quantitative metrics, we conduct a experiment to understand the sources of inefficiency in agent reasoning. By examining tool outputs, error messages, and the agent's subsequent actions, we perform a detailed attribution of erroneous and redundant tool calls. We categorize these issues into four primary types:

Parameter Error. This error occurs when an agent invokes a tool with incorrect or improperly formatted parameters. For example, an agent might mistakenly provide a year when a month is expected in a database query, leading to a failed retrieval.

Logical Error. This category includes actions that violate the logical dependencies of a task. A common example is an agent attempting to analyze a figure from a paper before executing the PDF Parser to load the document's content.

Redundant Action. This refers to tool calls that are unnecessary because the required information has already been obtained. For instance, an agent may fail to recall a previous action and re-execute the PDF parser on a document that has already been processed.

Tool Failure. This type of error covers cases where a tool fails to run correctly despite being called with valid parameters. Such failures are often external, such as a web search that returns no results due to network issues or content filters on sensitive keywords.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009

Table 8: Detailed specification of the implemented tool library on the PaperArena-Hub.

Tool Name	Description	Input	Output	Parameters	Detailed Operation
PDF Parser	Parses the non-readable PDF into an LLM-readable structured format.	PDF binary file	Markdown content and each layout block (JSON).	`pdf_path`	Employs the MinerU to extract text, tables, and figures along with their layout.
Database Querier	Retrieves papers from a vector database based on topic or keywords.	Conference name, paper status (e.g., Poster).	List of relevant paper information.	`conference`, `status`	First applying BM25 for sparse retrieval, followed by dense retrieval using cosine similarity.
Code Executor	Executes code snippets in a sandboxed environment for calculations or replication.	String containing code to be executed.	Standard output (stdout) and standard error (stderr)	`code`, `timeout`	Utilizes a secure, sandboxed Python environment to execute arbitrary code strings and return the execution results.
Cross-Ref Searcher	Finds and resolves bibliographic citations within a paper using arxiv APIs.	Citation key string from the paper text.	Structured citation data including title, abstract, and URL.	`citation_key`	Extracts reference text from the document and leverages GPT-4o to parse and retrieve corresponding bibliographic details.
Web Searcher	Queries web search engines to find background information on concepts, terms, or techniques.	Search query string.	Snippets of information and corresponding source URLs.	`query`, `n_nums`	Interfaces with the SerpAPI to perform web searches and parse the returned results into a structured format.
Context Retriever	Performs semantic search over long texts to find relevant passages [61].	Query string.	Relevant text chunks.	`query`, `top_k`	Uses `text-embedding-3-small` to embed the query and chunks, then performs similarity search.
Figure Analyzer	Analyzes visual content within figures and charts to extract information.	A specific question about the figure.	Extracted textual or numerical data from the figure.	`figure_id`, `query`	Feeds the figure and its caption directly to a multimodal LLM. If the base LLM is not multimodal, it first enriches the caption via GPT-4o.
Table Analyzer	Extracts specific data points or verifies trends from tables by parsing their structure.	A specific question about the table.	Extracted textual or numerical data from the table.	`table_id`, `query`	Converts the table image into structured HTML for LLM processing to avoid context limits of image input.

Table 9: Core components of the prompt for QA generation.

Component	Content
Expert Role	An AI Research Analyst and creative Content Creator, tasked with crafting insightful questions that test an agent's complex reasoning abilities on scientific literature.
Primary Task	Generate Question-Answer pairs with associated metadata, including the required tool chain and difficulty level, specifically for an Agentic Paper QA benchmark.
Core Principle	Questions must be unanswerable by a LLM's direct reasoning, compelling an agent to execute a workflow, interact with knowledge, and perform external verifications.
Precision Requirement	All entities must be precise (e.g., Table 1, [Smith et al., 2023]) to provide unambiguous identifiers for tool interaction and data retrieval operations.
QA Metadata	Each QA pair must be assigned a Type (e.g., Open Answer) and Difficulty (e.g., Hard) to ensure a diverse and well-structured evaluation benchmark.
QA Type Example	Open Answer: Requires synthesizing, analyzing, and reasoning to produce a structured, logical paragraph of 50 to 100 words, going beyond simple fact retrieval.
Difficulty Level Example	Hard: Involves in-depth verification, causal analysis, or synthesizing heterogeneous sources, thus requiring four or more sequential tool calls to derive the final answer.
Available Tools	A sandbox environment with tools like PDF Parser, Code Executor, Cross-Ref Searcher, and Table Analyzer to query, process, and verify information from documents.
Question Templates	Provide inspiration for complex reasoning patterns, such as verifying a paper's claims against its citations or programmatically replicating quantitative results reported in tables.
Final Output Format	A structured JSON list where each object contains the question, answer, all metadata, the required tool_chain, and a clear reasoning text explaining tool necessity.