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

Anonymous ACL submission

Abstract

Understanding and reasoning on the large-scale scientific literature is a crucial touchstone for large language model (LLM) based agents. However, existing works are mainly restricted to tool-free tasks within single papers, largely due to the lack of a benchmark that evaluates cross-paper reasoning and multi-tool orchestration in authentic research scenarios. In this work, we propose PaperArena, a benchmark to evaluate LLM-based agents on questions that require integrating information across multiple papers with the assistance of external tools. Given a research question, agents should formulate a reasoning plan, interact with multiple papers, and invoke appropriate tools to produce a well-grounded answer. To support standardized evaluation, we provide a platform for agent execution, offering a modular tool environment including multimodal parsing, context retrieval, and programmatic computation. Experiments reveal that even the leading LLM powering a well-established agentic workflow achieves merely 38.78% average accuracy, while on the hard subset, accuracy drops to only 18.47%. We further conduct empirical analyses of reasoning traces and agent behavior, providing the community with insights to develop and evaluate more capable scientific agents. Our code and data are available [1].

1 Introduction

With the large-scale scientific literature continually expanding, researchers are increasingly challenged by information overload (Houssard et al., 2024; Bollacker et al., 2002). As shown in Figure 1, addressing an authentic research question is a cognitively demanding task that requires researchers to formulate plans, leverage tools, and interact with heterogeneous content across multiple papers (Kononova et al., 2021; Katz et al., 2024). In response to this challenge, a growing trend is to use LLM agents to automate these complex workflows (Gao et al., 2025; Zhang et al., 2025).

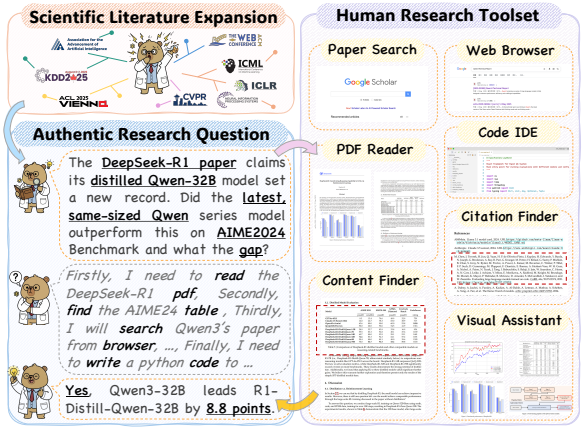

Figure 1: Facing the rapid expansion of scientific literature, researchers increasingly rely on reasoning and research tools to solve complex questions, while LLM agents show growing potential to assist this process.

Despite this potential, progress in LLM agents for scientific literature understanding and reasoning remains limited. A primary obstacle is the lack of a benchmark designed to evaluate agents in addressing complex authentic research questions. As shown in Table 1, existing benchmarks on scientific literature are often limited to single-document tasks such as explaining a concept from a single paragraph or extracting a specific value from a chart (Jin et al., 2019; Pramanick et al., 2024). However, these tasks mainly examine the knowledge of LLMs themselves rather than the tool-augmented reasoning abilities of advanced LLM agents (Cheng et al., 2025; Li et al., 2025a).

This critical gap motivates our effort to construct a benchmark specifically for assessing tool-augmented reasoning on scientific literature for LLM agents. This endeavor is challenging for several reasons. First, unlike conventional QA benchmarks, scientific literature reasoning requires cross-paper synthesis, fine-grained citation matching, and

[1] https://anonymous.4open.science/r/PaperArena-2026/

structured grounding over text, tables, and figures, which makes benchmark construction non-trivial and labor-intensive (Mialon et al., 2023; Siegel et al., 2024). Second, agent evaluation differs from conventional single-turn LLM assessment. It must support external tool interaction, long-horizon state tracking, and multi-step reasoning trajectories (Yue, 2025; Starace et al., 2025). Finally, final answer accuracy alone cannot reveal how an agent reaches an answer. Meaningful evaluation should also examine intermediate reasoning traces and agent behavior (Wang et al., 2022).

With these challenges in mind, we propose PaperArena and its accompanying PaperArena-Hub. The construction of PaperArena begins with a vast corpus comprising tens of thousands of papers from premier open-access AI conferences in 2025. From this corpus, we sample a representative subset of 100 papers and employ LLMs to synthesize initial question drafts. To ensure authenticity and difficulty, a team of expert annotators filter and rewrite these drafts into authentic formats and label their answers, minimal viable tool chains, difficulty, and question types. This rigorous pipeline yields a final set of 784 high-quality questions. Concurrently, we provide PaperArena-Hub to support the evaluation of both single- and multi-agent systems on PaperArena. The platform offers a modular tool environment including multimodal parsing, context retrieval, and code execution, while also supporting further agent behavior analysis.

We conduct extensive experiments on various leading LLM agents. Results reveal a stark performance gap: Gemini 2.5 Pro (Comanici et al., 2025), powering a well-established multi-agent system achieves an average accuracy of only 38.78% and drops to 18.47% on the hard subset, far below the 83.5% achieved by the human experts. When restricted to the same PaperArena-Hub toolset, human experts still perform strongly, showing that the tool environment provides sufficient access to the required evidence, while exposing substantial room for improving LLMs' tool-augmented reasoning abilities. Moreover, our empirical analysis of intermediate reasoning traces reveals three meaningful insights: tool usage imbalance, reasoning bottlenecks, and execution inefficiency. Further failure attribution shows that agents are more often limited by parameter errors, logical errors, and redundant actions. We invite the community to adopt PaperArena and PaperArena-Hub to develop and evaluate more capable scientific agents.

Table 1: Comparing PaperArena with existing scientific QA benchmarks across multi-step reasoning, multimodal understanding, cross-paper integration, and database interfacing.

Benchmark	Multi-Step Reasoning	Multimodal Understanding	Cross-paper Integration	Database Interfacing
PubMedQA (Jin et al., 2019)	✗	✗	✗	✗
CharXiv (Wang et al., 2024b)	✓	✓	✗	✗
emrQA (Pampari et al., 2018)	✗	✗	✓	✗
QASA (Lee et al., 2023)	✗	✗	✗	✗
ArgSciChat (Ruggeri et al., 2023)	✓	✗	✗	✗
QASPER (Dasigi et al., 2021)	✗	✗	✗	✗
SPIQA (Pramanick et al., 2024)	✓	✓	✗	✗
PaperMind (Zhao et al., 2026)	✓	✓	✓	✗
PaperArena	✓	✓	✓	✓

In summary, our contributions are as follows:

- We introduce PaperArena, a challenging benchmark for tool-augmented agentic reasoning on scientific literature, integrated with PaperArena-Hub to enable standardized evaluation across diverse LLM agents.
- We conduct extensive evaluations of leading LLM agents across diverse settings. The results reveal a significant performance gap between current agents and human experts, even under the same tool environment.
- Through empirical analysis of agent behavior, we provide systematic insights into the capabilities and limitations of current LLM agents in scientific literature reasoning.

2 Related Work

2.1 Benchmarks on Scientific Literature

Benchmarks for scientific literature understanding and reasoning grow rapidly in recent years (Chiang et al., 2024; Saikh et al., 2022). Early works focus on localized tasks, such as claim verification and phrase extraction (Wadden et al., 2020; Augenstein et al., 2017). With the rise of LLMs, research attention shifts toward knowledge-intensive scientific data, particularly structured academic charts (Zhou et al., 2025). Driven by the potential of LLM-based agents, benchmarks now target from-scratch scientific reproduction (Hua et al., 2025; Starace et al., 2025). Subsequent works expand into specialized fields like biomedicine and chemistry, prioritizing domain-specific professional challenges (Jin et al., 2019; Krithara et al., 2023; Guo et al., 2023). To better simulate authentic research, modern evaluation incorporates autonomous knowledge discovery, peer-review critique, and expert-level exams (Majumder et al., 2024; Baumgärtner et al., 2025; Phan et al., 2025).

2.2 Tool-Augmented Agents

The evolution of LLM reasoning enable complex agent workflows (Wei et al., 2022; Achiam et al., 2023; Li et al., 2025a). Early efforts expand reasoning boundaries through external web-based retrieval and synthesis (Nakano et al., 2021; Ahn et al., 2022). As reasoning and action capabilities converge, agents evolve further through planning, tool orchestration, memory management, and iterative self-reflection. (Yao et al., 2023; Shinn et al., 2023; Huang et al., 2024; Li et al., 2025b; Qiu et al., 2025). Furthermore, multi-agent systems push the frontiers of collaborative problem-solving (Hong et al., 2023; Wu et al., 2024; Wang et al., 2025b). Recent works start to focus on evaluating tool-augmented agents in specialized research domains, such as academic search and biological research (Lála et al., 2023; Skarlinski et al., 2024; Jin et al., 2025). However, most existing works remain restricted to tool-free or single-paper settings, leaving cross-paper, multi-tool reasoning in authentic research workflows underexplored. To bridge this gap, we propose PaperArena and its accompanying evaluation platform PaperArena-Hub.

3 PaperArena

In this section, we formulate the task, describes the construction and key features of PaperArena.

3.1 Task Formulation

PaperArena presents an agent with an authentic research question q and a source paper P in *pdf* format. The agent formulates a plan and iteratively leverages a predefined toolset $\mathcal{T}$. The process involves diverse content, especially when q refers to associated papers $\mathcal{P}_{\text{assoc}}$. After each tool execution, the agent decides whether to invoke another tool or return the final answer to q.

3.2 Benchmark Construction

Paper Corpus Preparation. We construct PaperArena on a curated corpus of 14,435 open-access latest AI papers from OpenReview (OpenReview, 2026) and Open Access (OpenAccess, 2026). Each paper is labeled with a 20-dimensional concatenated feature vector, comprising four one-hot encodings: influence (3-D), research field (8-D), method category (4-D), and evaluation type (5-D). A t-SNE visualization (Maaten and Hinton, 2008) 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 (Martins et al., 2024) to identify 50 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.

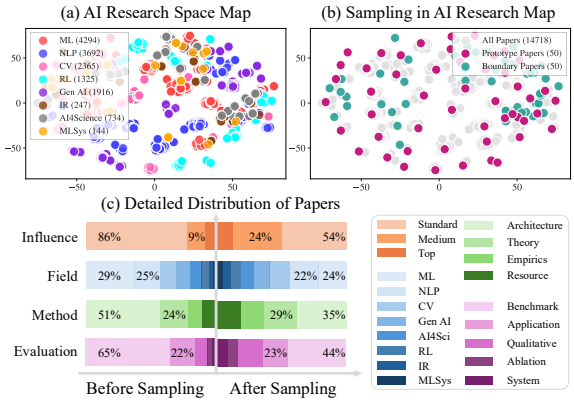

Figure 2: Our sampling strategy builds a balanced paper subset, as shown by the distribution changes across research fields before and after sampling.

QA Generation from Papers. We first define the external toolset $\mathcal{T}$ to align the generated questions with solvable operational boundaries, encompassing web searcher, cross-ref searcher, pdf parser, context retriever, figure/table analyzers, code executor, and database querier. To preserve critical structural information such as section headers and citations, we convert each paper's *pdf* into *markdown* format using MinerU (Wang et al., 2024a). We then prompt three distinct advanced LLMs with these structured segments and the specific definitions of $\mathcal{T}$ to independently generate candidate questions, yielding a total of 2,875 initial drafts. These LLM-generated drafts serve solely as preliminary seeds. A team of Ph.D. experts rigorously processes these drafts by filtering out trivial entries, balancing tool usage frequencies, removing obvious indicators to ensure sufficient difficulty and ultimately rewriting the drafts into authentic research question format. Finally, each question is carefully annotated with the ground-truth answer, the minimal viable tool chain, the difficulty level (easy, medium, hard), and the question type (multiple choice, concise answer, open answer). Further elaboration process can be found in Appendix A.

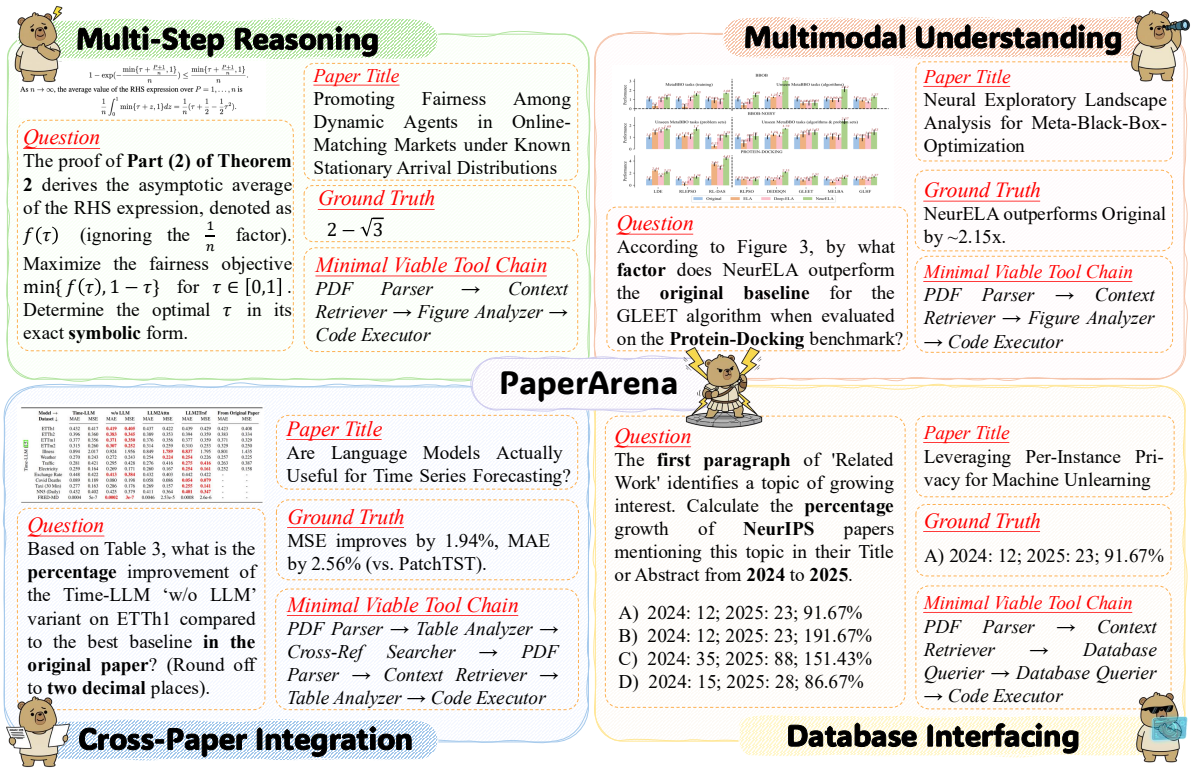

Figure 3: Cases of the four key features in PaperArena comprising multi-step reasoning, multimodal understanding, cross-paper integration, and database interfacing. Each quadrant presents a sample question with the associated paper title, ground truth, and the minimal viable tool chain.

3.3 Key Features of PaperArena

In Figure 3, we summarize four representative key features of PaperArena:

Multi-Step Reasoning. This category involves theoretical problems requiring sequential deduction. The example demands deriving a symbolic solution for a mathematical theorem to maximize an objective function. Such tasks require the agent to execute multi-step reasoning and use code tools to ensure mathematical precision when handling abstract logic and formal proofs.

Multimodal Understanding. This feature targets the interpretation of scientific charts and tables alongside text. In the shown case the agent must quantify performance gaps by extracting values from a bar chart. This demonstrates the need for the agent to align visual information with specific queries and perform precise quantitative analysis on complex experimental data.

Cross-Paper Integration. This dimension addresses the challenge of synthesizing fragmented information from multiple papers. The sample requires calculating improvements by comparing local results against baselines in an external citation. This scenario highlights the need for the agent to navigate between papers and integrate cross-reference context to verify claims across sources.

Database Interfacing. This aspect focuses on retrieving and processing data from external repositories. The example task involves determining the growth rate of a specific topic by analyzing metadata from conference proceedings. It illustrates the requirement for the agent to interface with databases via executable queries to extract statistical information from large structured database.

4 PaperArena-Hub

PaperArena-Hub enables standardized evaluation on PaperArena through an agent-centric architecture and specialized evaluation settings.

4.1 Platform Architecture

Agentic Workflow. PaperArena-Hub supports both single- and multi-agent workflows. In the single-agent setting, we adopt the ReAct-style workflow (Yao et al., 2023), where LLMs reason and act in alternation. In the multi-agent setting, we employ a centralized design with a manager agent responsible for high-level planning and delegating

Table 2: Comprehensive evaluation of nine leading LLMs in single- and multi-agent workflows on PaperArena against Ph.D. experts on PaperArena-Human. Performance is stratified by question type and difficulty level. The best accuracy (%) is in bold. We also report the average number of reasoning steps and reasoning efficiency (%).

Base LLM	Thinking	Open Src.	Multiple Choice			Concise Answer			Open Answer			Average		
			Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard	Acc.	Steps	Eff.
PaperArena-Hub (Single-Agent)														
Gemini 2.5 Pro	✓	✗	57.89	40.98	**22.22**	**58.49**	**40.00**	**21.95**	**56.52**	**36.07**	13.57	**36.10**	9.81	41.20
OpenAI o4-mini-high	✓	✗	57.89	**44.26**	**22.22**	55.66	37.30	14.63	52.17	33.61	12.86	33.93	10.63	35.12
Claude Sonnet 4	✓	✗	**60.53**	37.70	18.52	53.77	38.92	19.51	52.17	31.97	**15.00**	34.18	9.55	37.34
GPT-4.1	✗	✗	55.26	34.43	14.81	49.06	35.68	17.07	52.17	28.69	10.71	30.61	8.27	35.26
Claude 3.5 Sonnet	✗	✗	47.37	28.23	11.11	48.11	25.95	10.98	43.48	22.95	7.14	24.62	7.89	21.98
Qwen3-235B-Thinking	✓	✓	52.63	32.79	14.81	53.77	33.51	15.85	47.83	27.87	10.00	29.97	10.20	31.21
GLM-4.5	✓	✓	47.37	29.51	11.11	49.06	30.27	13.41	43.48	27.05	8.57	27.17	11.82	25.13
Qwen3-235B-Instruct	✗	✓	44.74	24.59	11.11	46.23	24.32	10.98	39.13	22.13	7.14	23.47	7.24	36.85
Kimi-K2-Instruct	✗	✓	44.74	22.95	11.11	45.28	22.70	9.76	39.13	20.49	6.43	22.32	12.86	23.57
PaperArena-Hub (Multi-Agent)														
Gemini 2.5 Pro	✓	✗	60.53	**42.62**	**33.33**	**63.21**	**42.70**	21.95	**65.22**	**39.34**	13.57	**38.78**	8.58	45.39
OpenAI o4-mini-high	✓	✗	60.53	41.00	25.93	62.26	41.08	21.95	56.52	36.89	**14.29**	37.37	8.30	43.41
Claude Sonnet 4	✓	✗	**68.42**	39.34	18.52	61.32	40.00	**23.17**	56.52	36.07	12.86	36.73	8.92	39.13
GPT-4.1	✗	✗	57.89	37.70	18.52	59.43	38.38	19.51	52.17	34.43	12.14	34.57	7.94	34.35
Claude 3.5 Sonnet	✗	✗	50.00	31.15	14.81	52.83	32.43	14.63	47.83	29.51	9.29	29.34	8.16	23.67
Qwen3-235B-Thinking	✓	✓	57.89	37.70	18.52	59.43	37.84	19.51	52.17	33.61	12.14	34.31	10.48	33.79
GLM-4.5	✓	✓	52.63	32.79	14.81	54.72	34.05	15.85	47.83	30.33	10.71	30.74	10.90	27.01
Qwen3-235B-Instruct	✗	✓	50.00	31.15	14.81	51.89	31.35	14.63	47.83	28.69	9.29	28.83	7.82	38.13
Kimi-K2-Instruct	✗	✓	50.00	27.87	11.11	50.00	28.11	12.20	43.48	25.41	7.86	26.28	11.94	29.45
Human Baseline (PaperArena-Human)														
Ph.D. Experts	PaperArena-Hub Env.		87.50	84.21	71.43	91.67	83.33	68.18	87.50	83.33	79.41	82.00	6.74	73.92
Ph.D. Experts	Open Env.		87.50	84.21	71.43	91.67	85.42	68.18	87.50	86.67	82.35	83.50	6.52	75.48

subtasks to worker agents within independent context spaces. Regardless of the setting, PaperArena-Hub manages the entire lifecycle of the agents.

Tool Environment. PaperArena-Hub integrates the full suite of tools defined in Section 3.2, allowing agents to interact with multiple papers, databases, and external resources. For example, our implementation of the database querying tool is backed by a structured dataset covering metadata from hundreds of major AI conferences in recent years. This includes paper titles, abstracts, and links. Agents can perform operations such as retrieving all papers on a topic from a specific conference and year, or extracting relevant entries based on keyword queries. Additionally, we incorporate robust error handling to minimize tool execution failures, ensuring reliable performance during long tool chains.

4.2 Evaluation Setup

Baselines. We evaluate representative nine leading LLMs in both single-agent and multi-agent workflows: Gemini 2.5 Pro (Comanici et al., 2025), OpenAI o4-mini-high (OpenAI, 2025b), Claude Sonnet 4 (Anthropic, 2025), GPT-4.1 (OpenAI, 2025a), Claude 3.5 Sonnet (Anthropic, 2024), Qwen3-235B-Thinking (Yang et al., 2025), GLM-4.5 (Zeng et al., 2025), Qwen3-235B-Instruct (Yang et al., 2025), and Kimi-K2-Instruct (Team K et al., 2025). We construct a 200-question subset from PaperArena (PaperArena-Human) and recruit three Ph.D. candidates, distinct from the annotators, under two settings: Open Environment, where experts may use common tools such as GPT-4o (Hurst et al., 2024), Google search, and Python; and PaperArena-Hub Environment, where experts are restricted to the same PaperArena-Hub toolset used by agents.

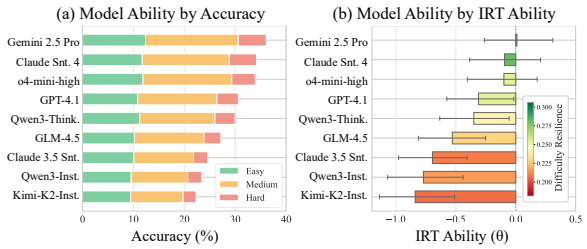

Figure 4: Comparison of model performance using accuracy and supplementary IRT estimation, showing a positive correlation and difficulty resilience across LLMs.

Evaluation Metrics. We use diverse metrics. Firstly, regarding outcome accuracy, all answers are evaluated using the LLM-as-a-Judge (Liu et al., 2025), where GPT-4o assigns binary labels. We prioritize validity over formatting, allowing for flexible output styles instead of strict exact match-

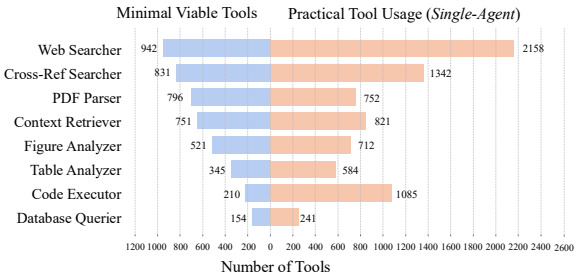

Figure 5: Comparison of minimal viable and practical tool usage by Gemini 2.5 Pro in the single-agent system. The imbalance shows that agents overuse general-purpose tools while underusing specialized tools.

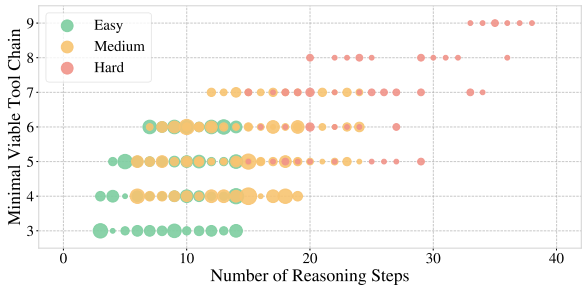

Figure 6: Correlation between minimal viable tool chain length and reasoning steps across difficulty levels, showing that harder tasks trigger longer traces but not necessarily effective progress.

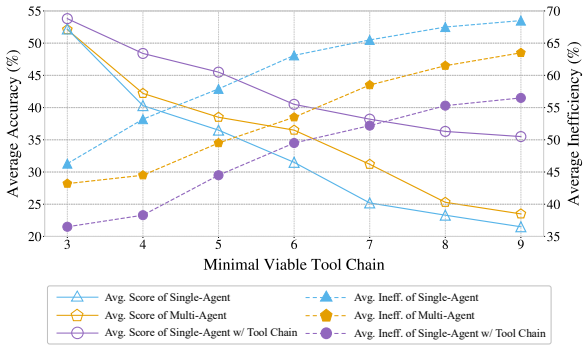

Figure 7: Efficiency analysis across minimal viable tool chain. Solid lines denote accuracy; dashed lines show average inefficiency ($1-$efficiency), revealing persistent execution gaps beyond planning.

ing (Wang et al., 2025a). Secondly, to analyze the reasoning process, we measure: (1) reasoning steps, the mean length of executed tool chains; and (2) efficiency, the normalized overlap between the tool chain executed by the agent and the minimal viable chain. As a supplementary diagnostic lens beyond aggregate statistics, we employ item response theory (IRT) (Liu et al., 2024) to analyze agent behavior. Specifically, we utilize a response matrix that records the binary correctness for each agent-question pair. By applying markov chain monte carlo (MCMC) sampling, we infer the item difficulty parameters β and latent agent ability θ based on these responses. Finally, to mitigate potential biases in LLM-as-a-judge, we manually verify all human answers and LLM responses to the questions within the PaperArena-Human. Detailed metrics definitions are available in Appendix B.

5 Experiments

In this section, we provide the overall performance of the agents, present three key observations and conduct an in-depth model diagnosis showing their distinctive failure patterns.

5.1 Main Results

Table 2 summarizes the performance of nine leading LLM agents under both single- and multi-agent workflows. Gemini 2.5 Pro achieves the best result in the multi-agent setting, yet its accuracy reaches only 38.78% and drops to 18.47% on the hard subset, far below the 83.5% achieved by human experts. When restricted to the same PaperArena-Hub toolset, human experts still perform strongly, showing that the tool environment provides sufficient access to required evidence, while exposing substantial room for improving LLMs' tool-augmented reasoning abilities. Performance degrades sharply on hard and open-answer questions, reflecting the difficulty of authentic scientific literature reasoning. Thinking models and multi-agent workflows generally improve performance, but the remaining gap indicates that current agents still struggle with planning efficiency and tool orchestration. As a supplementary view, Figure 4 shows that the IRT estimate broadly correlates with accuracy while revealing cases where similar aggregate scores exhibit different difficulty resilience.

5.2 Observation of Reasoning Traces

We analyze intermediate reasoning traces showing three meaningful insights: tool usage imbalance, reasoning bottlenecks, and execution inefficiency.

Tool Usage Imbalance. Agents show a clear preference for general-purpose tools such as the web searcher and code executor, causing practical reasoning traces to deviate from the minimal viable tool chain. As shown in Figure 5, practical use of the code executor is over five times the minimal viable requirement, while web searcher calls are more than double the minimal viable baseline. In

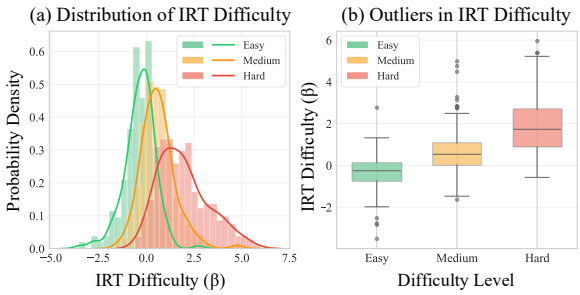

Figure 8: IRT-estimated difficulty broadly aligns with manual difficulty levels, supporting the task stratification. Outliers reveal cases where agents fail on questions that human annotators perceive as simpler.

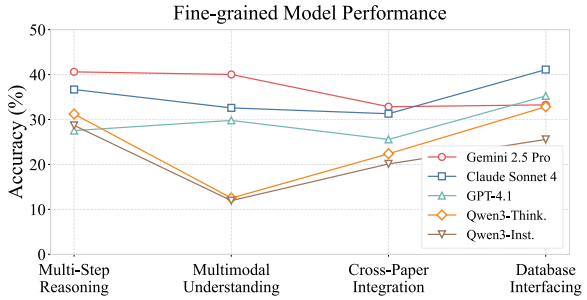

Figure 9: Fine-grained performance across the four primary domains in PaperArena. No single LLM consistently dominates all domains, suggesting the value of routing sub-tasks to specialized models.

contrast, specialized tools that directly target structured evidence are used much more conservatively. This tool usage imbalance indicates that current agents often rely on broad exploratory actions instead of selecting the specialized tools needed for efficient tool-augmented reasoning.

Reasoning Bottleneck. Agents tend to employ more reasoning steps for more difficult problems, yet longer traces do not yield proportional accuracy gains. Figure 6 shows a positive correlation between minimal viable tool chain length and reasoning steps, while Figure 7 shows that accuracy still declines as task complexity increases. This suggests that additional reasoning steps often reflect increased effort rather than effective progress toward the correct answer. Multi-agent workflows consistently outperform single-agent baselines in complex, multi-tool scenarios, suggesting that separating high-level planning from concrete tool execution is helpful but not sufficient to remove the reasoning bottleneck.

Execution Inefficiency. When explicitly prompted with the globally minimal viable tool chain, the single agent outperforms standard multi-agent workflows, establishing a planning ceiling in Figure 7. This result suggests that current agents are still limited by endogenous planning, but it also reveals a second failure mode: even with a near-perfect plan, agents can fail through flawed tool invocation, such as incorrect parameters. Thus, tool-augmented reasoning requires not only selecting an appropriate plan, but also executing each tool call with sufficient precision. The resulting execution inefficiency becomes more pronounced on complex tasks, highlighting room for improving LLMs' tool-augmented reasoning abilities beyond plan generation alone.

5.3 Diagnostics for Agent Behavior

Beyond quantitative metrics, we diagnose problem difficulty and agent capabilities. This analysis provides insights into agent behavior and reveals unique failure modes.

Supplementary Item-Level Analysis. To examine task complexity, we estimate the IRT difficulty parameter β for each question. As shown in Figure 8 (a), the estimated difficulty distributions align with the manual easy, medium, and hard annotations, supporting the difficulty stratification. This suggests that the annotated levels capture meaningful variation in agent performance rather than merely reflecting question formats. Meanwhile, Figure 8 (b) reveals upper outliers where agents fail on questions perceived as simpler by human annotators. These cases are useful for diagnosis because they expose reasoning requirements that appear manageable to experts but remain unexpectedly difficult for current LLM agents, highlighting a gap between agent and human expectations.

Fine-Grained Agent Performance. To probe model strengths and weaknesses, we analyze performance across the four primary capabilities of PaperArena: multi-step reasoning, multimodal understanding, cross-paper integration, and database interfacing. As shown in Figure 9, no single LLM universally dominates across all capabilities. Gemini 2.5 Pro performs best on multi-step reasoning and multimodal understanding, Claude Sonnet 4 is strongest on database interfacing, and both Qwen3 variants exhibit clear weaknesses in multimodal understanding due to the lack of native visual capabilities. These complementary profiles suggest the potential of routing sub-tasks to specialized LLMs rather than relying on a single model.

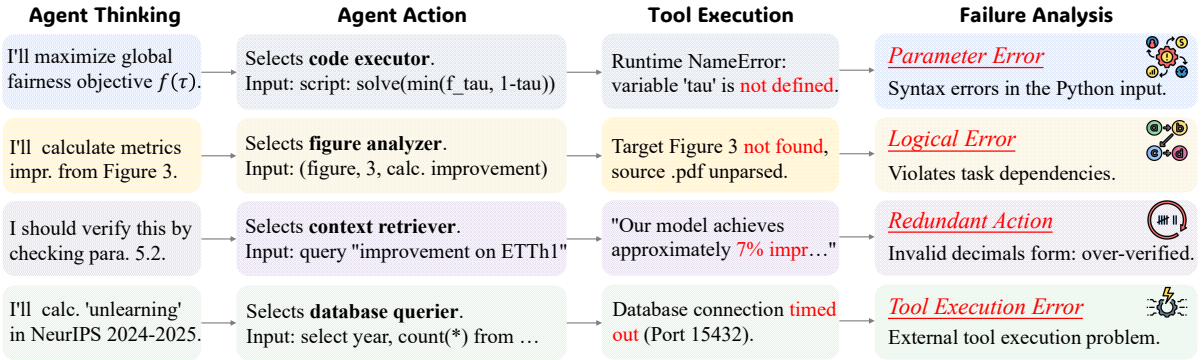

Figure 10: Failure analysis tracing the path from thinking to tool execution. The figure classifies failures into four categories and shows representative cases: parameter errors such as syntax issues, logical errors like dependency violations, redundant actions such as over-verification, and tool execution errors like connection timeouts.

Failure Analysis. We conduct a failure attribution analysis by inspecting tool outputs, error messages, and subsequent agent actions. As shown in Figure 10, parameter errors, logical errors, and redundant actions dominate the error landscape, while tool-side failures form only a minor portion. Open-source LLMs exhibit more parameter and logical errors, indicating weaknesses in complex planning, whereas closed-source thinking LLMs tend toward redundant actions, suggesting a form of over-verification. These patterns further support the reliability of PaperArena-Hub and point to tool-use and planning as central limitations of current LLM agents. The complete distribution of failures is detailed in Appendix C.

5.4 Analysis of Evaluator Alignment

Annotator Agreement. We established a reliable ground truth through human verification. A team of five Ph.D. experts refined the preliminary drafts by filtering trivial entries and balancing tool usage frequencies. The experts removed obvious indicators to maintain sufficient difficulty, and rewrote questions into authentic research formats. Subsequently, the annotators labeled each sample with the ground truth answer and the minimal viable tool chain, alongside the difficulty level and question type. To validate the reliability of this process, three independent experts evaluated a random subset of the data. The inter-annotator agreement (IAA) yields 0.83, which demonstrates high consistency among human evaluators.

Alignment of LLM and Human Judgement. We subsequently evaluate the consistency of GPT-4o as an automatic evaluator by comparing binary correctness labels against independent human scoring on the PaperArena-Human subset. We employ both simple agreement rates and the Cohen Kappa to measure inter-rater reliability, while correcting for chance agreement (Sun, 2011). The analysis reveals significant alignment between the model and human evaluators. We observe an agreement rate of 98.5% and a Kappa of 0.97 for responses from Gemini 2.5 Pro. Similarly, the metrics for answers provided by Ph.D. experts are 98.0% and 0.93, respectively. These values exceed the 0.8 threshold, which typically denotes near perfect agreement. The low number of discrepancies suggests that GPT-4o functions as a reliable proxy for assessing agent performance in this domain. The detailed calculation of metrics is provided in Appendix D.

6 Conclusion

We introduce PaperArena and PaperArena-Hub to evaluate tool-augmented reasoning on scientific literature through authentic research questions requiring multi-step reasoning, multimodal understanding, cross-paper integration, and database interfacing. Experiments reveal a gap between current LLM agents and human experts, even under the same tool environment, suggesting that current agents still have considerable room for improvement. Beyond accuracy, we provide in-depth analysis showing tool usage imbalance, reasoning bottlenecks, execution inefficiency, and agent-side errors as key limitations. By combining a challenging benchmark, a standardized tool environment, and behavior-level diagnostics, PaperArena provides a foundation for comparing agent workflows and understanding where current systems fail. We hope these findings help the community develop and evaluate more capable scientific agents.

Limitations

PaperArena has several limitations. First, its source corpus and questions focus on the AI domain, and future versions should expand to broader scientific areas. Second, the construction process relies on expert labor, which limits scalable expansion to much larger training-scale datasets. Third, the evaluation depends on the custom PaperArena-Hub tool environment; although this environment provides controlled and reproducible access to evidence, results may vary with different tool designs or implementations. Finally, despite the strong measured alignment between the LLM judge and human evaluators, LLM-based judging may still introduce instability in individual cases.

References

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*.

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*.

Anthropic. 2024. Claude 3.5 sonnet model card addendum. Technical report, Anthropic.

Anthropic. 2025. System card: Claude opus 4 & claude sonnet 4. Technical report, Anthropic.

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*.

Tim Baumgärtner, Ted Briscoe, and Iryna Gurevych. 2025. Peerqa: A scientific question answering dataset from peer reviews. *arXiv preprint arXiv:2502.13668*.

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):42–47.

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*.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*.

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*.

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*.

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*.

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*, 36:59662–59688.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. 2023. Metagpt: Meta programming for a multi-agent collaborative framework. In *The twelfth international conference on learning representations*.

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*.

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*.

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*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

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*.

Ruofan Jin, Zaixi Zhang, Mengdi Wang, and Le Cong. 2025. Stella: Self-evolving llm agent for biomedical research. *arXiv preprint arXiv:2507.02004*.

Uri Katz, Mosh Levy, and Yoav Goldberg. 2024. Knowledge navigator: LLM-guided browsing framework for exploratory search in scientific literature. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8838–8855, Miami, Florida, USA. Association for Computational Linguistics.

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).

Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. 2023. Bioasq-qa: A manually curated corpus for biomedical question answering. *Scientific Data*, 10(1):170.

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*.

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*, pages 19036–19052. PMLR.

Qingchuan Li, Jiatong Li, Zirui Liu, Mingyue Cheng, Yuting Zeng, Qi Liu, and Tongxuan Liu. 2025a. Are llms reliable translators of logical reasoning across lexically diversified contexts? *arXiv preprint arXiv:2506.04575*.

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

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*.

Zirui Liu, Yan Zhuang, Qi Liu, Jiatong Li, Yuren Zhang, Zhenya Huang, Jinze Wu, and Shijin Wang. 2024. Computerized adaptive testing via collaborative ranking. *Advances in Neural Information Processing Systems*, 37:95488–95514.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

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*.

Antônio Sobrinho Campolina Martins, Leandro Ramos De Araujo, and Débora Rosana Ribeiro Penido. 2024. K-medoids clustering applications for high-dimensionality multiphase probabilistic power flow. *International Journal of Electrical Power & Energy Systems*, 157:109861.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*.

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*.

OpenAccess. 2026. Openaccess.nl. https://www.openaccess.nl/. Accessed: 2026-01-27.

OpenAI. 2025a. Introducing gpt-4.1 in the api.

OpenAI. 2025b. Openai o3 and o4-mini system card. Technical report, OpenAI.

OpenReview. 2026. Openreview.net. https://openreview.net/. Accessed: 2026-01-27.

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*.

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*.

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

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*.

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)*, pages 7684–7699.

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):289–301.

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.

Zachary S Siegel, Sayash Kapoor, Nitya Nagdir, Benedikt Stroebl, and Arvind Narayanan. 2024. Core-bench: Fostering the credibility of published research through a computational reproducibility agent benchmark. *arXiv preprint arXiv:2409.11363*.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

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*.

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*.

Shuyan Sun. 2011. Meta-analysis of cohen's kappa. *Health Services and Outcomes Research Methodology*, 11(3):145–163.

Team K, Y. Bai, Y. Bao, et al. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

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*.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, Bo Zhang, Liqun Wei, Zhihao Sui, Wei Li, Botian Shi, Yu Qiao, Dahua Lin, and Conghui He. 2024a. Mineru: An open-source solution for precise document content extraction.

Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yu Yin, Shijin Wang, and Yu Su. 2022. Neuralcd: a general framework for cognitive diagnosis. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8312–8327.

Jiahao Wang, Mingyue Cheng, and Qi Liu. 2025a. Can slow-thinking llms reason over time? empirical studies in time series forecasting. *arXiv preprint arXiv:2505.24511*.

Zihan Wang, Ziqi Zhao, Yougang Lyu, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. 2025b. A cooperative multi-agent framework for zero-shot named entity recognition. In *Proceedings of the ACM on Web Conference 2025*, pages 4183–4195.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. 2024b. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 37:113569–113697.

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*, 35:24824–24837.

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*.

A. Yang, A. Li, B. Yang, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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)*.

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*.

Murong Yue. 2025. A survey of large language model agents for question answering. *arXiv preprint arXiv:2503.19213*.

A. Zeng, X. Lv, Q. Zheng, et al. 2025. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*.

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*, pages 1901–1910.

Yanjun Zhao, Tianxin Wei, Jiaru Zou, Xuying Ning, Yuanchen Bei, Lingjie Chen, Simmi Rana, Wendy H Yang, Hanghang Tong, and Jingrui He. 2026. Papermind: Benchmarking agentic reasoning and critique over scientific papers in multimodal llms. *arXiv preprint arXiv:2604.21304*.

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*.

A Benchmark Construction Details

In this section, we provide a granular breakdown of the construction process for the PaperArena. We detail the preparation of the paper corpus, the hierarchical sampling strategy, and the question generation pipeline involving human experts.

A.1 Paper Corpus Preparation

The foundation of the PaperArena benchmark is a large-scale corpus of academic papers. We adhere to a rigorous data processing pipeline consisting of collection, representation, and sampling stages to ensure validity and timeliness.

Data Collection and Preprocessing. We perform a comprehensive crawl of open-access repositories, specifically targeting OpenReview[2] and Open Access[3] platforms. To minimize data contamination where models might have memorized content during pre-training, we strictly filter for papers published in the year 2025. The raw collection contains 14,435 papers. We subsequently remove papers with incomplete metadata, non-English content, or abstract-only availability to ensure the quality of the papers.

Feature Representation. To analyze the distribution of the corpus, we map each paper to a semantic feature vector. We employ GPT-4o-mini as an annotator to extract key attributes from the abstract and introduction of each paper. These attributes are encoded into a 20-dimensional binary vector structured across four primary categories. The research field category covers eight domains: machine learning (ML), natural language processing (NLP), computer vision (CV), reinforcement learning (RL), generative AI (GenAI), information retrieval (IR), AI for science (AI4Science), and machine learning systems (MLSys). The remaining dimensions encode the methodology category, such as theoretical analysis or architecture design; the evaluation paradigm, including human evaluation or standard benchmarks; and proxy metrics for potential impact. We visualize these vectors using t-SNE with a perplexity of 30 and 1,000 iterations. As shown in Figure 2(a), the visualization reveals dense clusters representing mainstream research topics, highlighting the necessity for a balanced sampling strategy.

[2]https://openreview.net/

[3]https://www.openaccess.nl/en

Hierarchical Sampling Strategy. To construct a benchmark of 100 papers that balances representativeness and diversity, we design a two-stage hierarchical sampling method.

We first identify papers that act as prototypes for the dominant research clusters. We apply the K-Medoids algorithm to select 50 centers. Unlike K-Means, K-Medoids selects actual data points as centers, ensuring the chosen papers are real and valid. The objective is to minimize the intra-cluster distance:

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

where d denotes the Hamming distance suitable for the binary feature vectors.

To avoid evaluation bias where models are only tested on popular topics, we select 50 boundary papers from the sparse regions of the feature space. We employ an iterative Farthest Point Sampling (FPS) strategy. At each step, we select the paper that maximizes the minimum distance to the current selected set:

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

The final subset comprises 100 papers. As illustrated in Figure 11, this strategy shifts the distribution from being heavily skewed towards dominant fields and standard methods to a more uniform coverage across influence levels, research fields, methodologies, and evaluation types.

A.2 QA Generation Pipeline

Transforming the selected papers into high-quality QA pairs involves a sophisticated pipeline that combines automated parsing, LLM generation, and rigorous expert verification.

Definition of the Toolset. We define a comprehensive toolset to simulate a realistic agentic workflow, ensuring that the agent has sufficient capabilities to address complex queries while operating within clear constraints. The comprehensive toolset $\mathcal{T}$ is formally defined as the union of the following specialized modules:

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

Here, $\mathcal{T}_{\text{pdf}}$, $\mathcal{T}_{\text{tab}}$, and $\mathcal{T}_{\text{fig}}$ denote tools for parsing PDF documents, tables, and figures, respectively.

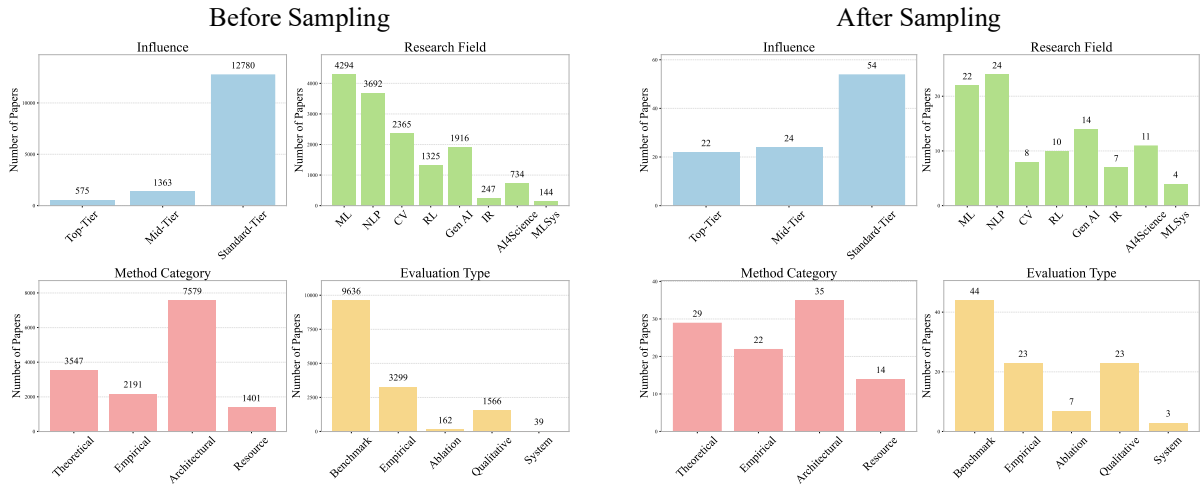

Figure 11: Comparison of paper distributions before and after sampling across four dimensions (influence, research field, method category, evaluation type). The sampled subset remains representative of the full corpus while preserving diversity across all categories.

$\mathcal{T}_{\text{ref}}$ handles reference management, while $\mathcal{T}_{\text{web}}$ and $\mathcal{T}_{\text{rag}}$ provide access to external web content and retrieval-augmented generation pipelines. Finally, $\mathcal{T}_{\text{db}}$ and $\mathcal{T}_{\text{code}}$ allow for structured database queries and executable code generation.

Structure-Aware Parsing. Standard text extraction often discards structural cues and visual information vital for understanding scientific papers. We utilize MinerU2.5 (Wang et al., 2024a) to convert the raw PDF of each paper into a structured Markdown format. Crucially, this process goes beyond text: it extracts and crops figures and charts into high-resolution image files to support multimodal analysis, and explicitly parses references into structured citation data. By preserving hierarchical headers, block equations, visual assets, and citation links, we establish a comprehensive grounding context for high-fidelity question generation.

Multi-Model Draft Generation. Relying on a single model for question generation can introduce specific cognitive biases. To ensure diversity in reasoning patterns, we prompt three distinct frontier LLMs: GPT-5 (Singh et al., 2025), Claude Sonnet 4 (Anthropic, 2025), and Gemini 2.5 Pro (Comanici et al., 2025). Acting as independent question proposers, these models receive the structured paper segments and the precise definitions of the toolset $\mathcal{T}$. We employ prompt 1 to enforce constraints that require questions to necessitate multi-hop reasoning and tool execution, specifically avoiding simple lookup queries. This ensemble approach leverages the unique strengths of each model architecture, yielding a robust and diverse pool of 2,875 initial drafts.

Human-in-the-Loop Refinement. The raw drafts generated by LLMs often exhibit triviality, hallucinated content, or rigid formatting that fails to mimic real-world inquiries. To bridge this gap, we assemble a team of domain experts with Ph.D. degrees to conduct a labor-intensive, three-pass refinement process. In the initial filtration phase, experts manually scrutinize thousands of drafts, discarding entries that are answerable via the abstract alone, logically flawed, or factually unsupported by the source text. Subsequently, in the rewriting phase, the team transforms the remaining candidates from mechanical, pointer-heavy instructions (e.g., "According to Table 3...") into authentic, intent-driven research questions. This step requires significant cognitive effort to ensure questions remain solvable while removing direct clues, thereby compelling agents to perform autonomous information retrieval. Finally, during the annotation and verification phase, experts rigorously validate the ground truth for each question by tracing the exact evidence within the paper. They further analyze the problem-solving logic to define the minimal viable tool chain and calibrate the difficulty level (Easy, Medium, Hard) based on the depth of reasoning required. This comprehensive human-in-the-loop workflow ensures the benchmark upholds the highest standards of scientific rigor and complexity.

To mitigate potential domain mismatch during

Table 3: 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

Table 4: Distribution of questions across key features in the PaperArena.

Multi-Step Reasoning	Multimodal Understanding	Cross-paper Integration	Database Interfacing	Total
170	261	194	159	784

refinement, we route each question to at least three different annotators and resolve remaining inconsistencies through consensus meetings with all five core experts. We also intentionally assign papers outside an expert's primary subfield. Since PaperArena focuses on multi-step tool orchestration rather than niche domain knowledge, researchers can effectively develop and grade high-quality tasks with the assistance of LLMs and web search.

A.3 Benchmark Statistics

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 3 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.

Key Features. We categorize the questions into four primary dimensions as illustrated in Table 4: Multi-Step Reasoning, Multimodal Understanding, Cross-Paper Integration, and Database Interfacing. The data distribution across these categories is notably balanced, with each feature represented by a substantial yet focused sample size. This well-proportioned allocation ensures that the evaluation remains comprehensive across all critical research tasks without being skewed toward any single dimension, thereby maintaining both the breadth and the rigor of the benchmark.

B Evaluation Details on PaperArena-Hub

B.1 Tool Environment Implementation

We implement a comprehensive tool suite on PaperArena-Hub to facilitate interaction with scientific literature and external knowledge sources. Table 5 presents the specific tool inputs and parameters. To ensure security and reproducibility, we deploy all computational tools within isolated Docker containers. These environments enforce strict constraints on network access and computing resources to prevent malicious operations. Additionally, the execution layer features a robust error handling mechanism. Upon failure, the system captures standard error output and returns a structured message, which enables the agent to debug and self correct within the allocated reasoning steps.

B.2 Evaluation Setup

Agent Baselines. 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: a temperature of 0.7, top_p of 0.9, a maximum of 4096 output tokens, and a maximum of 40 reasoning steps per question. Additionally, we employ a standardized chain-of-thought (CoT) (Wei et al., 2022) prompting strategy. The system prompt instructs the agent to think step-by-step, explicitly stating the rationale before invoking any tool. We conduct three independent runs for each experimental setting.

Human Baselines. 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 receive 16 hours over two days and may use standard tools, including GPT-4o for document comprehension, web search for information retrieval, and Python for coding tasks.

B.3 Evaluation Metric Definitions

We assess performance using three primary metrics and a supplementary item response theory (IRT) analysis.

Accuracy. We employ the LLM-as-a-Judge where GPT-4o assigns a binary score $S_{\text{corr}} \in \{0, 1\}$ to each answer. We deem an answer correct if its

Table 5: 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.	SQL queries using Python code	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 (Yu et al., 2024).	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.

final conclusion is accurate and supported by valid reasoning, regardless of stylistic format.

We additionally compute 95% bootstrap confidence intervals with 10,000 resamples for all accuracy metrics. The resulting intervals are narrow, and the overall rankings are broadly stable across resamples. This confirms that the main performance gap between current agents and human experts is robust.

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:

$$\text{Reasoning Steps} = \frac{1}{|Q|} \sum_{q \in Q} |\tau_q^{\text{exec}}|. \quad (4)$$

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:

$$\text{Efficiency} = \frac{1}{|Q|} \sum_{q \in Q} \frac{|\tau_q^{\text{exec}} \cap \tau_q^{\text{gt}}|}{|\tau_q^{\text{exec}}|}. \quad (5)$$

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

IRT Analysis. To disentangle the intrinsic ability of the models from the difficulty of the benchmark questions, we employ a 2-parameter logistic (2PL) item response theory model. This allows us to estimate the probability of a model j correctly answering question i. The probability $P_{ij}(\theta_j)$ is defined as:

$$P_{ij}(\theta_j) = \frac{1}{1 + e^{-a_i(\theta_j - \beta_i)}}, \quad (6)$$

where θ_j represents the latent ability of model j (higher θ indicates a more capable model); β_i denotes the difficulty parameter of question i, representing the point on the ability scale where an agent has a 0.5 probability of answering correctly; and a_i is the discrimination parameter of question i, indicating how well the question differentiates between models of varying abilities (a higher a_i implies a steeper slope at the inflection point). We estimate parameters using marginal maximum likelihood (MML) estimation with markov chain

Table 6: Fine-grained performance of five leading LLMs on the single-agent workflow, revealing their specialized strengths and weaknesses across the four primary features in questions on PaperArena.

Base LLM	Multi-Step Reasoning	Multimodal Understanding	Cross-paper Integration	Database Interfacing
Gemini 2.5 Pro	32.85	40.03	40.61	33.27
Claude Sonnet 4	31.30	32.59	36.68	41.12
GPT-4.1	25.58	29.81	27.54	35.26
Qwen3-Think.	22.39	12.54	31.22	32.85
Qwen3-Inst.	20.13	11.96	28.70	25.58

monte carlo (MCMC) sampling. This supplementary analysis provides supporting evidence that PaperArena maintains a balanced difficulty distribution and can reveal differences between models with similar aggregate performance.

C Detailed Results

Fine-grained Agent Performance. Table 6 presents the performance of five large language models across four capabilities. The data indicates that Gemini 2.5 Pro achieves the highest proficiency in multi-step reasoning and multimodal understanding. Claude Sonnet 4 demonstrates superior capability in database interfacing and outperforms others in code-centric interactions. The results also confirm the significant deficit of Qwen3 variants in multimodal understanding due to the lack of native visual processing. This breakdown reinforces that different models possess complementary advantages suitable for distinct sub-tasks.

Data Saturation Analysis. A natural concern is whether PaperArena will quickly saturate as newer models absorb more recent papers into their parameters. As a supplementary single-agent evaluation, we test recent models from late 2025 to early 2026 to further assess the continued difficulty of PaperArena. As shown in Table 7, advanced models improve incrementally over earlier systems, but their average accuracy remains around 36–43%, still far below the Ph.D.-level expert baselines of 82.0–83.5%. This gap suggests that PaperArena is not merely testing memorized paper content; it continues to require reliable tool use, evidence retrieval, and multi-step verification beyond parametric knowledge alone.

Failure Analysis. Figure 12 illustrates a comprehensive failure attribution analysis regarding the agent reasoning process. We perform this experiment by systematically inspecting tool output error messages and subsequent agent actions to

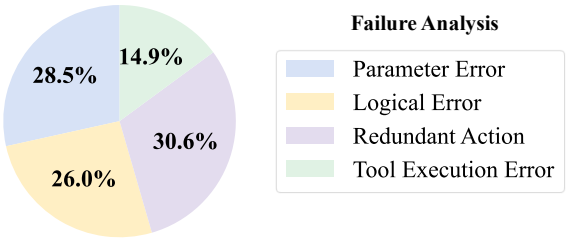

Figure 12: Failure analysis of five leading LLMs on the single-agent workflow, presenting the proportion of each error type in observed failures.

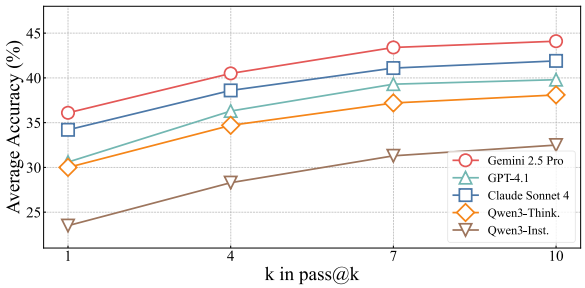

Figure 13: Performance in the pass@k evaluation reveals limits of test-time scaling capabilities on PaperArena. By trying multiple times and taking consistent answers, the agent achieves better performance.

accurately attribute each failure instance. These quantitative results are derived by averaging the error rates across the five models listed in Table 6. Crucially, our statistics indicate that tool execution errors account for less than 15% of observed failures. This finding attests to the robustness of our platform, suggesting that the underlying tools remain highly reliable and do not constitute the primary performance bottleneck.

Beyond error-type attribution, PaperArena reveals a substantial gap between the strongest LLM agents, which achieve approximately 40–45% accuracy, and Ph.D.-level experts, who score around 83.5%. Table 8 provides a direct comparison of their problem-solving processes, clarifying why models such as Gemini 2.5 Pro currently plateau around this performance range.

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 result is determined by a majority vote. Figure 13 reveals a clear positive correlation between attempts and agent performance, a trend consistent across all models. Notably, this scaling effect allows fast-thinking models to match or surpass the single-pass performance of slow-thinking

Table 7: Extended single-agent evaluation of recent models on PaperArena-Human for assessing potential data saturation. Best model-side accuracy values are in bold; human baselines are shaded.

Base LLM	Multiple Choice			Concise Answer			Open Answer			Average		
	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard	Acc.	Steps	Eff.
Gemini 2.5 Pro	57.89	40.98	22.22	58.49	40.00	21.95	56.52	36.07	13.57	36.10	9.8	41.2
Gemini 3 Pro	73.68	44.26	37.04	**75.47**	38.38	34.15	60.87	39.34	17.86	42.22	7.6	43.1
GPT-5.2	68.42	**47.54**	**44.44**	71.70	**40.54**	32.93	**65.22**	**42.62**	18.57	**43.11**	10.7	**45.2**
Claude Opus 4.5	**76.32**	45.90	40.74	70.75	39.46	**36.59**	**65.22**	38.52	**20.00**	42.86	8.6	44.9
Claude Opus 4	57.89	40.98	29.63	64.15	37.84	26.83	56.52	34.43	15.71	37.24	9.7	38.7
Kimi-K2.5	55.26	39.34	25.93	60.38	**40.54**	23.17	56.52	36.89	13.57	36.61	11.7	27.6
Ph.D. Experts (PaperArena-Hub Env.)	87.50	84.21	71.43	91.67	83.33	68.18	87.50	83.33	79.41	82.00	6.74	73.92
Ph.D. Experts (Open Env.)	87.50	84.21	71.43	91.67	85.42	68.18	87.50	86.67	82.35	83.50	6.52	75.48

Table 8: Process-level comparison between Gemini 2.5 Pro and Ph.D.-level experts on PaperArena.

Aspect	Gemini 2.5 Pro	Ph.D.-level Experts
Tool invocation	Favors quick web searches for metrics instead of downloading source documents, causing factual errors from imprecise web summaries.	Rigorously downloads and reads the source PDF to extract exact tables or text.
Planning	Loses the global objective when context grows or tools fail, often derailing into irrelevant exploratory paths.	Maintains strict to-do lists, allowing recovery from intermediate failures without losing sight of the goal.
Self-reflection	Either blindly accepts the first conclusion or over-verifies, introducing external noise into the context window that causes hallucinations.	Exercises calibrated skepticism, performing reasonable checks while actively filtering out contradictory noise.

ones. However, this approach introduces a significant challenge: a naive parallel implementation leads to nearly linear cost growth with respect to k. Compounding this issue, we find that performance gains exhibit diminishing returns; the improvement is substantial for small 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 focus on designing agents with larger token budgets under controlled cost constraints and developing more efficient parallelization strategies that move beyond simple majority voting.

D Human Annotation and Evaluation Details

D.1 Human Annotation Guideline

To ensure the high quality and difficulty of the benchmark, we employ a Human-in-the-Loop refinement process following the human annotator instruction 2, where a team of five Ph.D. experts transforms preliminary LLM-generated drafts into authentic, solvable, and challenging research questions. To strictly validate the reliability of this process, three independent experts, distinct from the annotators, subsequently evaluate a random subset of the data. We quantify the consistency among these three evaluators using Krippendorff's alpha (α), calculated as:

$$\alpha = 1 - \frac{D_o}{D_e}. \tag{7}$$

In this formula, D_o represents the observed disagreement derived from weighted discrepancies among the three raters, while D_e denotes the disagreement expected by chance. The resulting α of

Table 9: Summary of human participant roles in benchmark construction and evaluation.

Group	Size	Role and Relation
Annotators	5	Refine drafts, verify answers, and annotate tool chains, difficulty, and question types.
Validators	3	Distinct from annotators; re-evaluate a random subset for Krippendorff's alpha.
Human baselines	3	Distinct from annotators; solve PaperArena-Human under PaperArena-Hub and open environments.
Testing experts	2	Distinct from annotators and human baselines; score 400 Gemini and human responses for GPT-4o alignment analysis.

0.83 demonstrates high consistency. The comprehensive guidelines provided to the annotators are presented below.

D.2 Calculation of Alignment Metrics

To rigorously quantify the consistency between annotators (for the ground truth dataset) and between the LLM evaluator and human experts, we utilize two primary metrics: Simple Agreement Rate and Cohen Kappa.

The agreement metrics are based on 400 responses, including 200 responses from Gemini 2.5 Pro and 200 responses from one human testing expert, distinct from the annotators, on the PaperArena-Human subset. Two other testing experts, also distinct from the annotators, assess all 400 responses to establish the human judgment, which we then compare against GPT-4o's judgments. Disagreements between the two human evaluators are resolved through discussion to reach consensus. Such disagreements are rare because even open-ended answers are highly concise, typically under 100 words, making correctness evaluation straightforward.

Simple Agreement Rate. The Simple Agreement Rate (denoted as p_o) measures the raw proportion of instances where two evaluators assign the identical label (e.g., Correct or Incorrect) to a given response. It is calculated as:

$$p_o = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(y_i^{(A)} = y_i^{(B)}), \tag{8}$$

where N is the total number of samples, $y_i^{(A)}$ and $y_i^{(B)}$ represent the binary labels assigned by evaluator A and evaluator B for the i-th sample, respectively. $\mathbb{I}(\cdot)$ is the indicator function, which equals 1 if the condition inside is true and 0 otherwise.

Cohen Kappa. While simple agreement is intuitive, it does not account for the possibility of agreement occurring by random chance. To address this, we employ Cohen Kappa coefficient (κ) (Sun, 2011), a robust statistic for inter-rater reliability. The coefficient is defined as:

$$\kappa = \frac{p_o - p_e}{1 - p_e}. \tag{9}$$

Here, p_o corresponds to the observed agreement rate defined in Eq. (1). The term p_e represents the hypothetical probability of chance agreement, calculated based on the marginal distributions of the labels. For a binary classification task (e.g., Correct/Incorrect) with two evaluators, p_e is:

$$p_e = \left(\frac{n_C^{(A)}}{N} \times \frac{n_C^{(B)}}{N} \right) + \left(\frac{n_I^{(A)}}{N} \times \frac{n_I^{(B)}}{N} \right), \tag{10}$$

where $n_C^{(j)}$ and $n_I^{(j)}$ denote the count of samples labeled as correct and incorrect by evaluator j (where $j \in \{A, B\}$), respectively. A κ value greater than 0.8 is typically interpreted as indicating near-perfect agreement.

D.3 Annotator Compensation

All human annotators and testing experts are compensated for their work. The compensation rate is set to no less than ten times the local minimum hourly wage for each participant, reflecting both the specialized expertise required and the time-intensive nature of benchmark refinement and evaluation.

E Ethical Considerations

All source papers are retrieved from public, open-access platforms via unbiased, automated sampling. To ensure data integrity and mitigate potential biases, both the dataset construction and the experimental evaluation are conducted in an anonymized manner, with all author and institutional information strictly excluded. The resulting dataset is intended solely for non-commercial academic research, with all generated artifacts released under an open license to foster reproducibility.

Agentic QA Generation Prompt

1. ROLE
You are a Senior AI Research Scientist designing a Benchmark for Agentic Reasoning. Your task is to generate **10 extremely challenging, multi-hop questions** based on the provided paper.

CRITICAL CONSTRAINT:
Every question must require a **complex tool-chain of 4 to 10 steps**. Questions must be impossible to answer by simply reading the provided PDF. They **must** require retrieving external papers, verifying official documentation, or performing code-based re-calculations.

2. STRICT TOOL DEFINITIONS
Your questions must be solvable *only* by combining the following specific tools:
`PDF Parser`: Parses the content of a PDF file.
`Context Retriever`: Search and retrieve text chunks from a parsed document.
`Table Analyzer`: Extracts structured data from tables.
`Figure Analyzer`: Extracts visual information from charts/diagrams.
`Cross-Ref Searcher`: Finds metadata/links for citations (e.g., "[12]") mentioned in the text.
`Web Searcher`: Searches the live internet for documentation or general info.
`Database Querier`: Queries specific academic databases for paper lists (e.g., by conference).
`Code Executor`: Runs Python code for calculation or algorithm verification.

3. QUESTION TEMPLATES
You can use the following templates as reference:
Templates 1
Chain: `PDF Parser` → `Table Analyzer` → `Cross-Ref Searcher` → `PDF Parser` → `Table Analyzer` → `Code Executor`
Example: "Table 2 reports the 'Llama-2' baseline as 65.4. Retrieve the original Llama-2 paper referenced in Citation [12], find the reported score on this **exact same dataset split** in that paper's results table, and uses code to **calculate the precise numerical difference**."
Templates 2
Chain: `PDF Parser` → `Context Retriever` → `Cross-Ref Searcher` → `PDF Parser` → `Context Retriever` → `Code Executor`
Example: "Equation (4) uses a weighting factor λ which the text states is adopted from [Citation 3]. Find the value of λ defined in [Citation 3], then use the raw inputs from Table 1 of the current paper to **re-calculate the loss result** via code. Does it match?"
Templates 3
Chain: `PDF Parser` → `Figure Analyzer` → `Web Searcher` → `Context Retriever`
Example: "Figure 3 illustrates the architecture using a 'Multi-Head Attention' block. Search for the official PyTorch documentation for `nn.MultiheadAttention`. **Identify which input parameter** shown in Figure 3's diagram is NOT present in the standard PyTorch v2.0 API implementation."
Templates 4
Chain: `PDF Parser` → `Table Analyzer` → `Database Querier` → `PDF Parser` → `Table Analyzer` → `Code Executor`
Example: "The paper claims SOTA performance on Dataset X compared to methods up to 2024. Use the database to find top-tier papers from CVPR 2025 that evaluate on Dataset X. **Compare the best score** from those 2025 papers against this paper's result and report the difference."

Human Annotator Guideline

1. ROLE
You are a Senior Research Scientist and Benchmark Evaluator. Your task is to review, refine, and annotate question-answer pairs derived from scientific literature.

2. OBJECTIVE
Transform raw, LLM-generated draft questions into a high-quality benchmark dataset that tests the capabilities of autonomous research agents.

PHASE 1: FILTERING & REWRITING

Validity Check: Discard questions that are hallucinated, trivial (lookup of a single keyword), or unsolvable even with the provided toolset $\mathcal{T}$.
De-contextualization: Remove phrases that reveal the source immediately. The question must simulate a genuine user query (e.g., "What is the specific F1-score achieved by Method X in the dataset Y?").
Authenticity Refinement: Rewrite the question to sound natural yet professional, ensuring clearly defined operational boundaries.

PHASE 2: ANNOTATION & LABELING

Ground Truth: Provide the exact answer and specify the acceptable margin of error or unit for numerical values.
Minimal Viable Tool Chain: List the strictly necessary sequence of tools (e.g., `PDF_Parser` $\rightarrow$ `Code_Executor`) required for efficiency.
Question Type: Classify into *Multiple Choice* (select best option), *Concise Answer* (specific entity/number), or *Open Answer* (synthesis/explanation).
Difficulty Level: Categorize as *Easy* (direct retrieval), *Medium* (multi-hop reasoning), or *Hard* (complex synthesis, code execution, or cross-referencing).