DeepScholar-Bench: A Live Benchmark and Automated Evaluation for Generative Research Synthesis

Anonymous Author(s)
Affiliation
Address
email

1 Introduction

A crucial underpinning of human knowledge and innovation is the ability of human experts to *research and synthesize* known facts and new findings to support others in comprehending, verifying and building upon existing work. Recently, new systems designed for *generative research synthesis* offer exciting capabilities and promises to automate challenging research synthesis tasks, which often require laborious hours of literature search, reading, and writing from human experts. These systems include a wide array of both commercial DeepResearch offerings, including ones from OpenAI OpenAI (2025a), Gemini Gemini (2025a), Anthropic Anthropic (2025a), Grok xAI (2025), and Perplexity Perplexity (2025), as well as open-source methods, such as STORM Shao et al. (2024), DeepResearcher Zheng et al. (2025), and OpenScholar Asai et al. (2024). These systems push the frontier of AI capabilities, demonstrating promising performance on existing factuality and question-answering benchmarks Wei et al. (2024); Krishna et al. (2025); Mialon et al. (2023); Wei et al. (2025).

Yet, as this new class of systems emerges, a key question remains: *how should we benchmark and evaluate generative research synthesis?* The progress of these systems requires benchmarks that carefully evaluate their critical capabilities. Specifically, these systems provide three core functions: (1) *retrieval*, typically from a large, complex, and constantly-evolving corpus, such as the live web, to collect key information (2) *knowledge synthesis*, to generate coherent long-form answers that surface key facts, integrating general knowledge and findings from many retrieved sources, and (3) *verifiability*, providing citations that allow readers to trace each stated claim in the synthesized answer to a reputable source from the retrieved set. The ideal benchmark must holistically evaluate across all three of these dimensions, while providing a realistic and challenging research synthesis task, which the community can reliably and repeatedly benchmark on over time.

Unfortunately, existing benchmarks fall-short of these goals. Many prior works evaluate generative research synthesis systems using existing question answering benchmarks Wei et al. (2025, 2024); Mialon et al. (2023); Krishna et al. (2025); Wu et al. (2025); Wadden et al. (2020); Jin et al. (2019); Yang et al. (2018); Joshi et al. (2017); Kwiatkowski et al. (2019); Ho et al. (2020); Trivedi et al. (2022); Lee et al. (2023), which do not reflect realistic research synthesis tasks and instead focus on questions with short-form, easily-verifiable answers, making them severely limited for this setting. These question-answering benchmarks do not capture the complexity of long-form answers synthesized from many sources, a key component of research synthesis. To address this limitation, several recent works Asai et al. (2024); Zheng et al. (2024); you.com (2025) instead leverage expert-curated datasets with open-ended research questions and exemplar answers. Unfortunately, these benchmarks quickly become stale and outdated as new information emerges. Furthermore, these datasets risk data contamination as new models are trained on snapshots of the web, including public datasets. The prohibitive expense of curating, maintaining, and updating expert-curated benchmarks further limits their utility towards realistic, scalable evaluation.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

In this work, we introduce DeepScholar-bench, a live benchmark and holistic, automated evaluation framework designed to evaluate generative research synthesis. DeepScholar-bench draws queries from recent, high-quality ArXiv papers and focuses on a real research synthesis task: generating the related work sections of a paper by retrieving, synthesizing, and citing prior research. We develop an automated evaluation framework that holistically assesses performance across three key dimensions, *knowledge synthesis*, *retrieval quality*, and *verifiability*, using metrics that show strong agreement with expert human judgments. We also develop DeepScholar-base, a reference pipeline for generative research synthesis, implemented efficiently using the LOTUS API.

Using the DeepScholar-bench framework, we systematically evaluate the performance of existing systems, including open-source research synthesis systems, search AI's with strong proprietary models, OpenAI DeepResearch, and DeepScholar-base. We find that all of these existing methods exhibit significant opportunity for improvement, with no baseline exceeding a score of .19 across all metrics. Furthermore, on several key metrics, including Nugget Coverage, Reference Coverage and Document Importance, each evaluated method's performance remains well below .45, reflecting the inherent difficulty of the DeepScholar-bench task, which requires systems to navigate the live web, reasoning about the relevance and importance of documents as well as surfacing key facts into a concise final answer. Notably, OpenAI's DeepResearch offers strong performance relative to other baselines, outperforming many prior methods on knowledge synthesis and retrieval quality, with scores of .392 on Nugget Coverage, .187 on Reference Coverage and .124 on Document Importance; however, it struggles to provide strong verifiability relative to many other methods. We also find that DeepScholar-base, a relatively simple reference pipeline implemented efficiently using the LOTUS API noa (2025b); Patel et al. (2025), consistently improves upon performance of prior open-source systems and search AI's, as well as achieving competitive performance and up to $6.3\times$ higher verifiability compared to OpenAI's DeepResearch. Nevertheless, DeepScholar-bench remains far from saturated, representing exciting opportunities for further work. We hope that our benchmark framework and reference pipeline support the progress of new systems, and we believe that resolving DeepScholar-bench represents a critical milestone towards more capable AI systems.

Overall, our main contributions are the following:

- We propose DeepScholar-bench, a live benchmark dataset with real research synthesis tasks and an automated evaluation that holistically assesses performance across three key dimensions: knowledge synthesis, retrieval quality, and verifiability. Our analysis demonstrates that our evaluation metrics are exhibit strong human agreement scores.
- We develop DeepScholar-base, a reference pipeline for generative research synthesis, providing a strong baseline with competitive performance compared to prior open-source systems, search AI's, and OpenAI's DeepResearch.
- We perform a systematic evaluation of existing baselines on DeepScholar-bench, including open-source systems, search AI's, OpenAI's DeepResearch, and DeepScholar-base. We find that DeepScholar-bench is far from saturated, demonstrating significant opportunity for improvement, with no baseline achieving a score greater than .19 across all metrics.

2 The DeepScholar Dataset

We study the task of generating a related works section of an academic paper, a fundamental research synthesis task. We choose this task for two key reasons. First, this task is a *real* research task performed by academic experts, allowing our benchmark to reflect realistic, difficult and useful queries. Second, the online availability of diverse, high-quality academic papers allows us to develop an *automated* dataset construction pipeline that we can continuously run to obtain new queries over time. We construct our dataset by scraping papers from ArXiv arxiv (2025), which continuously posts thousands of new pre-print papers across a wide array of scientific domains each week. We formalize our dataset task as follows: given a description, d of a paper, the goal is to retrieve a set of relevant sources, S, and generate a related works sections, W, for the paper by synthesizing and citing the retrieved documents. We provide further details of our automated data collection framework in the Appendix Section 6.2.

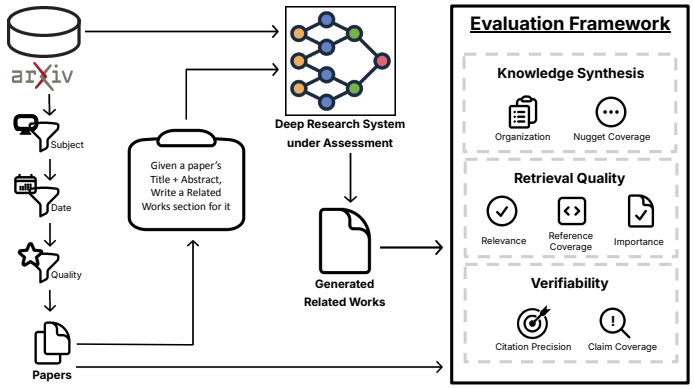

Figure 1: DeepScholarBench Overview. To curate our dataset with real and challenging research tasks, we scrape recent, high-quality ArXiv papers from diverse domains, and extracting key attributes from each paper through an automated data pipeline that can easily be re-run. Our dataset task is to generate a related works section given information about a paper, such as it's title and abstract. The DeepScholar-bench evaluation framework then holistically measures performance of generated reports on three key dimensions: knowledge synthesis, retrieval quality and verifiability.

3 The DeepScholar Evaluation Framework

To assess performance on DeepScholar-bench, we develop an automated evaluation framework, which holistically measures the performance of system answers across the three key dimensions: knowledge synthesis, retrieval quality and verifiability. Evaluating the accuracy of our long-form synthesis task is inherently difficult since each query admits many possible answers, lacking a straightforward notion of correctness. Moreover, developing an automated evaluation requires reliable metrics that exhibit high agreement with expert human annotators, another significant challenge. To address these challenges, our holistic evaluation assesses each system response across seven key metrics (Table 3), which permit many possible correct answers, often leveraging human-written exemplars from our dataset. Specifically, on the knowledge synthesis dimension, we evaluate a generated answer's Organization, using pairwise comparisons to human exemplars, and Nugget Coverage Pradeep et al. (2025); Upadhyay et al. (2024b); Faggioli et al. (2023); Rahmani et al. (2024a); Upadhyay et al. (2024a), assessing the efficiency of the generated response in capturing key information and facts. To assess retrieval quality, we measure the Relevance Rate of retrieved sources, the Document Importance of sources, according to citation counts of each reference, and Reference Coverage, by assessing the generated report's coverage of notable important references recovered from the human-written exemplar. To assess the Verifiability of each report, we measure its Citation Precision, whether each citation supports the given claim, and Citation Coverage, measuring whether each claim is fully supported by the cited sources Gao et al. (2023); Worledge et al. (2024); Liu et al. (2023). Our human agreement study validates that these automated metrics are effective, demonstrating strong agreement between our model-based judges and expert human annotators. We provide a detailed overview of each metric in Appendix Section 6.3.

4 DeepScholar-base

We introduce DeepScholar-base, an open-source reference pipeline designed to perform generative research synthesis. Given a user's query, DeepScholar-base iteratively generates web-search queries, summarizes the search results in each round before generating a new query. The system then post-processes the search results leveraging a series of semantic operators Patel et al. (2025), which we implement efficiently using the LOTUS API noa (2025b). This includes a semantic filtering step, semantic top-k ranking step and a semantic aggregation. We provide further details of each step of our reference pipeline in Appendix Section 6.5.

5 Experimental Results

In this section, we evaluate recent state-of-the-art generative research systems as well as DeepScholar-base on DeepScholar-bench. We benchmark open-source research systems, including DeepRe-

Table 1: Main Results.

	Knowledge Synthesis		Retrieval Quality			Verifiability	
	Org.	Nug. Cov.	Rel. Rate	Ref Cov.	Doc Imp.	Cite-P	Claim Cov.
Human-written Exemplars							
Human-written Exemplars	.500	1.000	.585	1.000	1.000	.278[1]	.205[1]
Open Source Research Systems							
DeepResearcher (Llama-4)	.206	.230	.385	.047	.008	.312	.396
STORM (Llama-4)	.119	.183	.218	.003	.006	.238	.586
OpenScholar (Llama-4)	.309	.278	.017	.008	.013	.010	.138
Search AI's							
Search AI (Llama-4-Scout)	.151	.193	.445	.060	.009	.316	.368
Search AI (GPT-4.1)	.556	.265	.490	.050	.009	.498	.470
Search AI (o3)	.849	.348	.610	.165	.026	.425	.495
Search AI (Claude)	.698	.307	.583	.131	.008	.701	.760
Search AI (Gemini)	.706	.277	.583	.061	.010	.415	.398
Commercial Systems							
OpenAI DeepResearch	.857	.392	.629	.187	.124	.399	.138
DeepScholar Baseline							
DeepScholar-base (Llama-4)	.254	.262	.421	.103	.008	.648	.826
DeepScholar-base (GPT-4.1)	.825	.407	.608	.162	.007	.652	.636
DeepScholar-base (GPT-4.1, o3)	.857	.405	.659	.162	.008	.617	.614
DeepScholar-base (GPT-4.1, Claude)	.786	.370	.586	.167	.007	.936	.817
DeepScholar-base (GPT-4.1, Gemini)	.762	.332	.602	.181	.006	.851	.865

Table 2: Human Agreement Evaluation.

Evaluation Metric	LLM-Classified Labels	Human-Agreement Score with LLM
Organization	Pairwise Comparison (Lose / Tie / Win)	78%
Nugget Coverage	Nugget Importance (Vital / Non-vital)	72%
Retrieval Relevance Score	Graded Relevance (0/1/2)	70%
Reference Coverage	Reference Importance (Not Imp./ Imp.)	82%

searcher Zheng et al. (2025), STORM Shao et al. (2024) and OpenScholar Asai et al. (2024), search AI's, with Llama-4-Scout-17B-16E-Instruct noa (2025c), GPT-4.1-2025-04-14 OpenAI (2025b), o3-2025-04-16 OpenAI (2025b), Claude-opus-4-20250514 Anthropic (2025b), and Gemini-2.5-pro Gemini (2025b) models, OpenAI's o3-deep-research OpenAI (2025b), and DeepScholar-base. We provide details of our setup in Appendix 6.6. Overall, we find the following:

- Existing baselines for generative research synthesis, including strong open-source LLM systems, search AI's, and commercial systems, demonstrate significant room for improvement across all three key dimensions: knowledge synthesis, retrieval quality and verifiability, with no method exceeding a score of 19% across all metrics, as shown in Table 1.
- DeepScholar-base provides a strong baseline for generative research synthesis, providing competitive performance compared to all other methods and up to 6.3× higher verifiability compared to OpenAI's DeepResearch, shown in Table 1.
- Our automated evaluation approach is effective, demonstrating high agreement with over 200 human annotations, shown in Table 2.

We provide an extended analysis and ablations in Section Appendix 6.8.

[1]The verifiability metrics we use in our evaluation likely under-estimate the performance of human writing. This is because the metrics we measure, Citation Precision and Claim Coverage, require us to verify entailment relations between claims, within the report, and snippets from the cited reference. For each LLM-based system, we are able to track the precise snippet and context from each cited source. On the other hand, for the human-written exemplars, we lack gold labels pointing to the precise snippet of text that each reference refers to. Our measurements for the human-written exemplars instead rely on the title and abstract of each cited source when testing for entailment relations.

References

[n.d.]. TREC CAR ← TREC CAR. https://trec-car.cs.unh.edu/

2025a. Llama 4 - a meta-llama Collection. https://huggingface.co/collections/meta-llama/llama-4-67f0c30d9fe03840bc9d0164

2025b. lotus-data/lotus. https://github.com/lotus-data/lotus original-date: 2024-07-16T16:39:06Z.

2025c. meta-llama/Llama-4-Scout-17B-16E-Instruct · Hugging Face. https://huggingface.co/meta-llama/Llama-4-Scout-17B-16E-Instruct

2025d. Wikipedia:Good article criteria. https://en.wikipedia.org/w/index.php?title=Wikipedia:Good_article_criteria&oldid=1303221295 Page Version ID: 1303221295.

Salaheddin Alzubi, Creston Brooks, Purva Chiniya, Edoardo Contente, Chiara von Gerlach, Lucas Irwin, Yihan Jiang, Arda Kaz, Windsor Nguyen, Sewoong Oh, Himanshu Tyagi, and Pramod Viswanath. 2025. Open Deep Search: Democratizing Search with Open-source Reasoning Agents. https://arxiv.org/abs/2503.20201v1

Anthropic. 2025a. Claude takes research to new places. https://www.anthropic.com/news/research

Anthropic. 2025b. Introducing Claude 4. https://www.anthropic.com/news/claude-4

Negar Arabzadeh and Charles L. A. Clarke. 2025. Benchmarking LLM-based Relevance Judgment Methods. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Padua, Italy) *(SIGIR '25)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3726302.3730305

arxiv. 2025. arXiv.org e-Print archive. https://arxiv.org/

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'arcy, David Wadden, Matt Latzke, Minyang Tian, Pan Ji, Shengyan Liu, Hao Tong, Bohao Wu, Yanyu Xiong, Luke Zettlemoyer, Graham Neubig, Dan Weld, Doug Downey, Wen-tau Yih, Pang Wei Koh, and Hannaneh Hajishirzi. 2024. OpenScholar: Synthesizing Scientific Literature with Retrieval-augmented LMs. https://doi.org/10.48550/arXiv.2411.14199 arXiv:2411.14199 [cs].

Guglielmo Faggioli, Laura Dietz, Charles Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, and Henning Wachsmuth. 2023. Perspectives on Large Language Models for Relevance Judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*. 39–50. https://doi.org/10.1145/3578337.3605136 arXiv:2304.09161 [cs].

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling Large Language Models to Generate Text with Citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 6465–6488. https://doi.org/10.18653/v1/2023.emnlp-main.398

Gemini. 2025a. Gemini Deep Research — your personal research assistant. https://gemini.google/overview/deep-research/

Gemini. 2025b. Gemini models | Gemini API. https://ai.google.dev/gemini-api/docs/models

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, Donia Scott, Nuria Bel, and Chengqing Zong (Eds.). International Committee on Computational Linguistics, Barcelona, Spain (Online), 6609–6625. https://doi.org/10.18653/v1/2020.coling-main.580

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A Dataset for Biomedical Research Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 2567–2577. https://doi.org/10.18653/v1/D19-1259

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Regina Barzilay and Min-Yen Kan (Eds.). Association for Computational Linguistics, Vancouver, Canada, 1601–1611. https://doi.org/10.18653/v1/P17-1147

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2025. Fact, Fetch, and Reason: A Unified Evaluation of Retrieval-Augmented Generation. https://doi.org/10.48550/arXiv.2409.12941 arXiv:2409.12941 [cs].

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics* 7 (2019), 452–466. https://doi.org/10.1162/tacl_a_00276 Place: Cambridge, MA Publisher: MIT Press.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. https://doi.org/10.48550/arXiv.2309.06180 arXiv:2309.06180 [cs].

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 *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 19036–19052. https://proceedings.mlr.press/v202/lee23n.html

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2025. From Generation to Judgment: Opportunities and Challenges of LLM-as-a-judge. https://doi.org/10.48550/arXiv.2411.16594 arXiv:2411.16594 [cs].

Ruosen Li, Teerth Patel, and Xinya Du. 2024. PRD: Peer Rank and Discussion Improve Large Language Model based Evaluations. https://doi.org/10.48550/arXiv.2307.02762 arXiv:2307.02762 [cs].

Nelson Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating Verifiability in Generative Search Engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 7001–7025. https://doi.org/10.18653/v1/2023.findings-emnlp.467

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. GAIA: a benchmark for General AI Assistants. https://doi.org/10.48550/arXiv.2311.12983 arXiv:2311.12983 [cs].

Federico Nanni, Bhaskar Mitra, Matt Magnusson, and Laura Dietz. 2017. Benchmark for Complex Answer Retrieval. https://doi.org/10.48550/arXiv.1705.04803 arXiv:1705.04803 [cs].

OpenAI. 2025a. Introducing deep research | OpenAI. https://openai.com/index/introducing-deep-research/

OpenAI. 2025b. Model - OpenAI API. https://platform.openai.com

OpenAlex. 2025. OpenAlex: The open catalog to the global research system | OpenAlex. https://openalex.org/

Liana Patel, Siddharth Jha, Melissa Pan, Harshit Gupta, Parth Asawa, Carlos Guestrin, and Matei Zaharia. 2025. Semantic Operators: A Declarative Model for Rich, AI-based Data Processing. https://doi.org/10.48550/arXiv.2407.11418 arXiv:2407.11418 [cs].

Perplexity. 2025. Introducing Perplexity Deep Research. https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research

Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. 2025. The Great Nugget Recall: Automating Fact Extraction and RAG Evaluation with Large Language Models. https://doi.org/10.48550/arXiv.2504.15068 arXiv:2504.15068 [cs].

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 Technical Report. https://doi.org/10.48550/arXiv.2412.15115 arXiv:2412.15115 [cs].

Hossein A. Rahmani, Clemencia Siro, Mohammad Aliannejadi, Nick Craswell, Charles L. A. Clarke, Guglielmo Faggioli, Bhaskar Mitra, Paul Thomas, and Emine Yilmaz. 2024a. LLM4Eval: Large Language Model for Evaluation in IR. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Washington DC USA, 3040–3043. https://doi.org/10.1145/3626772.3657992

Hossein A. Rahmani, Emine Yilmaz, Nick Craswell, Bhaskar Mitra, Paul Thomas, Charles L. A. Clarke, Mohammad Aliannejadi, Clemencia Siro, and Guglielmo Faggioli. 2024b. LLM-Judge: LLMs for Relevance Judgments. https://doi.org/10.48550/arXiv.2408.08896 arXiv:2408.08896 [cs].

Yijia Shao, Yucheng Jiang, Theodore A. Kanell, Peter Xu, Omar Khattab, and Monica S. Lam. 2024. Assisting in Writing Wikipedia-like Articles From Scratch with Large Language Models. https://doi.org/10.48550/arXiv.2402.14207 arXiv:2402.14207 [cs].

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: an Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 15725–15788. https://doi.org/10.18653/v1/2024.acl-long.840

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. https://doi.org/10.48550/arXiv.2104.08663 arXiv:2104.08663 [cs].

Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2024. Large language models can accurately predict searcher preferences. https://doi.org/10.48550/arXiv.2309.10621 arXiv:2309.10621 [cs].

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop Questions via Single-hop Question Composition. https://doi.org/10.48550/arXiv.2108.00573 arXiv:2108.00573 [cs].

Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Daniel Campos, Nick Craswell, Ian Soboroff, Hoa Trang Dang, and Jimmy Lin. 2024a. A Large-Scale Study of Relevance Assessments with Large Language Models: An Initial Look. https://doi.org/10.48550/arXiv.2411.08275 arXiv:2411.08275 [cs].

Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Nick Craswell, and Jimmy Lin. 2024b. UMBRELA: UMbrela is the (Open-Source Reproduction of the) Bing RELevance Assessor. https://doi.org/10.48550/arXiv.2406.06519 arXiv:2406.06519 [cs].

Ellen M. Voorhees. 2009. I Come Not To Bury Cranfield, but to Praise It. *NIST* (Oct. 2009). https://www.nist.gov/publications/i-come-not-bury-cranfield-praise-it Last Modified: 2017-02-19T20:02-05:00 Publisher: Ellen M. Voorhees.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or Fiction: Verifying Scientific Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 7534–7550. https://doi.org/10.18653/v1/2020.emnlp-main.609

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. https://doi.org/10.48550/arXiv.2411.04368 arXiv:2411.04368 [cs].

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. BrowseComp: A Simple Yet Challenging Benchmark for Browsing Agents. https://doi.org/10.48550/arXiv.2504.12516 arXiv:2504.12516 [cs].

Theodora Worledge, Tatsunori Hashimoto, and Carlos Guestrin. 2024. The Extractive-Abstractive Spectrum: Uncovering Verifiability Trade-offs in LLM Generations. https://doi.org/10.48550/arXiv.2411.17375 arXiv:2411.17375 [cs].

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. 2025. WebWalker: Benchmarking LLMs in Web Traversal. https://doi.org/10.48550/arXiv.2501.07572 arXiv:2501.07572 [cs].

xAI. 2025. Grok 3 Beta — The Age of Reasoning Agents | xAI. https://x.ai/news/grok-3

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. https://doi.org/10.48550/arXiv.1809.09600 arXiv:1809.09600 [cs].

you.com. 2025. Su-Sea/ydc-deep-research-evals: you.com's framework for evaluating deep research systems. https://github.com/Su-Sea/ydc-deep-research-evals

Haozhen Zhang, Tao Feng, Pengrui Han, and Jiaxuan You. 2024. AcademicEval: Live Long-Context LLM Benchmark. (Oct. 2024). https://openreview.net/forum?id=iRYExPKnxm

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025. DeepResearcher: Scaling Deep Research via Reinforcement Learning in Real-world Environments. https://arxiv.org/abs/2504.03160v4

Yuxiang Zheng, Shichao Sun, Lin Qiu, Dongyu Ru, Cheng Jiayang, Xuefeng Li, Jifan Lin, Binjie Wang, Yun Luo, Renjie Pan, Yang Xu, Qingkai Min, Zizhao Zhang, Yiwen Wang, Wenjie Li, and Pengfei Liu. 2024. OpenResearcher: Unleashing AI for Accelerated Scientific Research. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Delia Irazu Hernandez Farias, Tom Hope, and Manling Li (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 209–218. https://doi.org/10.18653/v1/2024.emnlp-demo.22

FIELD NAME	DESCRIPTION	EXAMPLE
arxiv_id	ArXiv paper ID	2506.02838v1
title	Paper title	TaxAgent: How Large Language Model Designs Fis...
authors	Comma-separated list of authors	Jizhou Wang, Xiaodan Fang, Lei Huang, Yongfeng...
abstract	Paper abstract from ArXiv	Economic inequality is a global challenge, int...
categories	ArXiv categories	cs.AI, econ.GN, q-fin.EC, I.2.11, I.6.5
published_date	First Publication date	2025-06-03T13:06:19+00:00
clean_latex_related_works	Related Works section derived from LATEX of paper	\subsection{ Traditional Tax Systems}\nProgress ...

FIELD NAME	DESCRIPTION	EXAMPLE
Parent Paper	*Info about the paper in which the citation appears.*	
parent_paper_title	Title of the parent paper.	TaxAgent: How Large Language Model Designs Fis...
parent_paper_arxiv_id	ArXiv ID of the parent paper.	2506.02838v1
Cited Paper	*Details from the reference entry.*	
citation_shorthand	Citation key in the bibliography.	NBERw21340
cited_paper_title	Title as listed in the reference list.	Effective Policy for Reducing Inequality? The ...
cited_paper_arxiv_link	ArXiv link if provided.	NaN
Bibliographic Data	*Metadata from the attached bibliography.*	
bib_paper_authors	Authors from external metadata.	Hoynes, Hilary W and Patel, Ankur J
bib_paper_year	Publication year	2015
bib_paper_month	Publication month	July
bib_paper_url	URL from bibliographic records.	http://www.nber.org/papers/w21340
bib_paper_doi	DOI from external metadata.	10.3386/w21340
bib_paper_journal	Journal or series name.	NaN
original_title	Official title from bibliographic databases.	Effective Policy for Reducing Inequality? The ...
Search Results (Verified)	*Info obtained by searching online.*	
search_res_title	Title from the search result.	Effective Policy for Reducing Inequality? The ...
search_res_url	URL from the search result.	https://www.nber.org/papers/w21340
search_res_content	Abstract snippet from the search result page.	We use a quasi-experiment approach, using vari...

Figure 2: DeepScholar-bench dataset schema.

6 Appendix

6.1 Discussion of Related Work

Long-form Synthesis Benchmarks. Several recent benchmarks offer datasets designed to evaluate long-form research synthesis tasks, however, their design often involves manually-curated queries, which are prone to data staleness, contamination and limited scalability. DeepScholar-bench addresses these challenges by providing an automated data pipeline and live benchmark with realistic, challenging and recent research synthesis tasks, in contrast to prior works. Specifically, ScholarQABench Asai et al. (2024) creates realistic literature review questions with detailed answer rubrics, written by expert PhD annotators from the computer science, biomedicine, physics and neuroscience domains. Similarly, OpenResearcher Zheng et al. (2024) constructs a dataset of around 38 scientific text summarization queries, among other research-style questions, by recruiting experts to write and annotate queries. Likewise, DeepConsult you.com (2025) provides a suite of expert-curated, open-ended research queries related to business and consulting. Unfortunately, these expert-curated benchmarks, are expensive to construct, and difficult to update, causing them to quickly become outdated, as new information becomes available, These prior benchmark datasets also risk data contamination, as new models are trained on publically available data. Our work instead proposes a scalable, automated pipeline to provide a live, evergreen dataset that reflects diverse and recent research queries.

Similarly to our approach, the FreshWiki dataset Shao et al. (2024) is constructed using an automated data pipeline; however, the dataset task focuses on the generation of Wikipedia-like articles, whereas this work focuses on a significantly different and difficult synthesis task based on cutting-edge research. The FreshWiki framework focuses on evaluating the article pre-writing stage as well as the generated full-length article based on a ground truth Wikipedia article and established criteria of a good Wikipedia article noa (2025d). This task reflects an interesting area of study; however, in this work, we instead focus on generative research synthesis, and our dataset task focuses on a complex research synthesis task derived from high-quality academic papers. Moreover, our automated evaluation approach is tailored to holistically assess the three key capabilities of generative research synthesis: retrieval, knowledge synthesis and verifiability.

AcademicEval Zhang et al. (2024) evaluates long-context generation tasks using an ArXiv-derived dataset, similar to ours, however, its task differs substantially, focusing on summarization *without retrieval*, which is a key component of our task and of generative research synthesis systems more broadly. While the AcademicEval task provides a fixed set of references to a summarization system, our task requires a system to navigate the live web to collect information, and we evaluate this crucial capability of generative research synthesis.

Question Answering Benchmarks. Several recent works on generative research synthesis focus their evaluation on question answering (QA) benchmarks, which, unlike DeepScholar-bench, do not evaluate complex long-form research synthesis tasks and instead focus on short-form answers, whcih can be easily evaluated for correctness. These question answering benchmarks include SimpleQA Wei et al. (2024), FRAMES Krishna et al. (2025), GAIA Mialon et al. (2023), as BrowserComp Wei et al. (2025), WebWalkerQA Wu et al. (2025) or others traditionally used to evaluate retrieval-augmented generation Wadden et al. (2020); Jin et al. (2019); Yang et al. (2018); Joshi et al. (2017); Kwiatkowski et al. (2019); Ho et al. (2020); Trivedi et al. (2022); Lee et al. (2023). While these benchmarks involve a retrieval component, their synthesis task differs substantially from the generative research synthesis task we study. Specifically, each of these QA benchmarks focus on short-form questions with easily verifiable answers and straightforward notions correctness. In contrast, our benchmark provides a framework for studying complex, long-form research synthesis tasks, which lack an absolute notion of correctness and admit many possible reasonable answers.

6.2 Extended Description of DeepScholar-bench Dataset

We study the task of generating a related works section of an academic paper, a fundamental research synthesis task. We choose this task for two key reasons. First, this task is a *real* research task performed by academic experts, allowing our benchmark to reflect realistic, difficult and useful queries. Second, the online availability of diverse, high-quality academic papers allows us to develop an *automated* dataset construction pipeline that we can continuously run to obtain new queries over time.

We construct our dataset by scraping papers from ArXiv arxiv (2025), which continuously posts thousands of new pre-print papers across a wide array of scientific domains each week. We formalize our dataset task as follows: given a description, d of a paper, the goal is to retrieve a set of relevant sources, S, and generate a related works sections, W, for the paper by synthesizing and citing the retrieved documents. We briefly overview our automated data collection framework (Section 6.2.1) and describe the dataset instantiation (Section 6.2.2) used in our evaluation (Section 5).

6.2.1 Automated Data Collection Framework

Our automated data collection framework aims to achieve the following design goals:

1. Inclusion of *diverse* paper topics across a wide variety of research domains.
2. Focus on *recent* research papers, both to provide realistic, timely benchmark queries and to control data contamination when benchmarking models trained on snapshots of the web.
3. Control for *quality* of the scraped ArXiv papers and extracted data

Figure 1 provides an overview of our dataset pipeline, which collects and extracts metadata about each paper (e.g., the title, abstract, and ArXiv link), the paper's related works section, and information on each citation from the paper's related works section. Our data collection pipeline extracts this information by scraping and selectively filtering ArXiv papers according to a number of configured settings.

Specifically, the pipeline loads papers from a list of configured ArXiv domain categories (e.g., cs.ML) and filters paper according to the configured publication-date range. To avoid possible data contamination arising from multiple ArXiv versions, some of which may have been released prior to the configured publication-date range, we exclusively include v1 ArXiv papers. To control for paper quality, our pipeline optionally provides a configuration setting which filter's out papers which are not listed as "accepted" or "published" at a conference within the paper's comment metadata, which often lists updates to the paper's status. We also disclude papers that do not have an explicit "Related Works" section and .bib file, containing well-formatted bibliography entries. For each paper, we then extract the Related Works section, from both the LaTex files, and PDF file, if available. We clean the extracted LaTex section to remove figures, sub-figures, labels and comments. We also extract all citations found in the related work section from the LaTex .bib file. For each citation we use the ArXiv API and the OpenAlex API to recover more detailed information, such as abstracts, authors, and links for ArXiv and non-ArXiv references respectively.

Table 3: Summary of Evaluation Metrics.

Metric	Description
Knowledge Synthesis	
Organization	assesses organization and coherence of system answer
Nugget Coverage	assesses the answer's coverage of essential facts
Retrieval Quality	
Relevance Rate	measures avg. relevance among all referenced sources
Document Importance	measures how notable referenced sources are, using citation counts
Reference Coverage	assesses the referenced set's coverage of key, important references
Verifiability	
Citation Precision	measures percent of cited sources that support their accompanying claim
Claim Coverage	measures percent of claims that are fully supported by cited sources

6.2.2 Dataset Description and Statistics

We now briefly summarize the dataset we use in our evaluation in Section 5 which represents an instantiation of our automated data collection pipeline. Our datasets take ArXiv papers with a publication date between April and June, following the April 5th, 2025 release date of the Llama-4 models noa (2025a), the main open-source model we benchmark in our evaluation. Our dataset consist of papers scraped from a diverse set of 18 disticnt ArXiv domains, including, cs.AI, cs.CV, cs.DB, cs.LG, cs.AR, cs.CG, cs.DC, cs.DS, cs.IR. To control for quality, we filter out papers not accepted at a conference, and we additionally exclude papers with related works sections longer than 1,000 words. Our final dataset instantiation includes 63 ArXiv papers, each providing a single query and expert-written exemplar for our benchmark. We make our scripts available to allow others to configure different datasets, and we plan to update our dataset to provide a continual evaluation with recent queries. Our experiments leverage the abstract of each paper as the paper's description d, provided to each baseline system as context within the query. We analyze the human-written exemplars from our dataset, and we find that, on average, each related work section contains 23 unique references, and we find over 63% of all cited references on ArXiv.

We provide a detailed overview and schema of the DeepScholar-bench dataset in Figure 2.

6.3 Extended Description of DeepScholar-bench Evaluation Framework

Evaluating research synthesis systems is challenging due to the complexity of the task and the variability of possible correct answers. Research synthesis systems generate complex, long-form reports, which are difficult to evaluate and lack a notion of "ground truth" correctness. The task we consider departs significantly from traditional question answering and RAG-based evaluations Joshi et al. (2017); Lee et al. (2023); Jin et al. (2019); Kwiatkowski et al. (2019); Trivedi et al. (2022); Yang et al. (2018); Ho et al. (2020), requiring a carefully designed and holistic evaluation framework. An exemplar research report must retrieve important relevant sources, synthesize an informative and well-organized answer, and provide appropriate references allowing readers to verify and re-trace facts. Our holistic evaluation framework thus analyzes three key dimensions, providing an automated, scalable approach for each: *knowledge synthesis* (Section 6.3.1), *retrieval quality* (Section 6.3.2), and *verifiability* (Section 6.3.3). While our experiments in Section 5 evaluate one specific task, our evaluation framework may extend to a wide range of research synthesis tasks Shao et al. (2024); you.com (2025); Zheng et al. (2024); Asai et al. (2024), which exhibit similar properties and challenges. We provide a detailed overview of our evaluation framework in this section.

6.3.1 Knowledge Synthesis

We evaluate both the information content surfaced in each synthesized report and the overall organization and coherence of the report.

Organization and Coherency. Our automatic evaluation adopts an LLM-as-a-judge approach to assess the organization and coherence of each system answer. We use preference-based pairwise comparisons, where the LLM-judge is presented with the details of the criteria to judge, the human-written exemplar, and a generated report and is asked to mark which is better. To avoid position bias Li et al. (2025), we evaluate each pair twice, permuting their positions. This model-based evaluation provides scalability while also serving as a strong surrogate for human preferences Rahmani et al. (2024b); Li et al. (2024, 2025); Arabzadeh and Clarke (2025), which we validate in our experiments in Section 5. We report the win-rate of each system using the prompt shown in Box 1 in the Appendix.

Nugget Coverage. To assess the quality of the information content presented by a generated report, we use a nugget-based evaluation. An *information nugget* Pradeep et al. (2025); Upadhyay et al. (2024b); Faggioli et al. (2023); Rahmani et al. (2024a); Upadhyay et al. (2024a) is an essential fact or components relevant for an answer. The process of *nuggetization* involves decomposing information-dense text into essential components, which aid in evaluation. For our task, we generate nuggets from the human-written exemplar related-work section for each query, following the automated, LLM-based methodology of Pradeep et al. Pradeep et al. (2025). In section 5 we validate that the model-based approach has strong agreement with expert annotations when labeling nuggets. For, each report, we compute the nugget coverage score, the fraction of nuggets that are present in each system answer.

6.3.2 Retrieval Quality

While traditional information retrieval (IR) evaluations typically leverage gold labels for document relevance scores and a controlled corpus Thakur et al. (2021); noa ([n.d.]); Nanni et al. (2017), the research synthesis task we study in this work differs substantially. An expert-written report section and its sources provide *one* reasonable retrieved set, but there may be many possible alternative sets that are likewise high-quality. Moreover, live web search is a key component of research synthesis tasks and obtaining gold relevance labels over this corpus is prohibitively expensive. To address these challenges, our evaluation measures three components of the retrieved set: the relevance rate, reference coverage of key sources, and document importance.

Relevance Rate. We asses the relevance of each retrieved document, following the Cranfield model Voorhees (2009), which is standard in IR evaluations and considers relevance of individual documents given a query, independent of other documents. Due to the significant cost of obtaining human-annotated relevance judgments, recent works Upadhyay et al. (2024b); Faggioli et al. (2023); Rahmani et al. (2024b); Thomas et al. (2024); Asai et al. (2024) study leveraging an LLM-as-a-judge for relevance judgment task, demonstrating their effectiveness on traditional IR datasets. Building on these works, we adopt an LLM-based approach for assigning graded relevance scores from 0 to 2 to each generated research report using the prompt in Box 2. For each retrieved set, S, for a given query, we compute the average document relevance over the set, following the below formula:

$$RR(S) = \frac{1}{2|S|} \sum_{s \in S} Rel(s),$$

where $Rel(s)$ is the graded relevance score of source s. We validate the agreement between LLMs and human annotators for this task in our experiments in Section 5.

Reference Coverage. We introduce a metric to measure the *reference coverage* of the retrieved set for each report. A key challenge in measuring this value is in defining a set of important references for each generated report, that a good research report should cite. To build this set, we take all references from the high-quality, human-written exemplar and label each as either "important" or "not-important", considering a "not-important" reference as one that could be omitted or substituted by a different reference. We find that a LLM-based judge is effective in assigning these labels. For a given report, we then compute its reference coverage by taking the ratio of the number of important references in the system-generated report to the number of important references in the

human-written exemplar, following the below formula, where E is the set of "important" references from the human-written exemplar:

$$RC(S, E) = \frac{1}{|E|} \sum_{s \in S} I[s \in E].$$

Document Importance. While the above relevance and coverage metrics assess *topical* matches between the retrieved set and the user query, an ideal research synthesis system must also retrieve *notable and important* sources. Exemplar human-written reports typically contain ample references of primary-sources and highly-cited academic publications. While the ideal notion of *document importance* depends on the particular task and user, in our task, we adopt the following metric by considering the number of citations that each reference retrieved by the system has. We consider the median number of citations per reference over the set of all retrieved sources, S, provided by a given baseline. We compute document importance as the ratio of this value for the given baseline compared to the median citations per reference over the set of sources, S^* provided by the human-written exemplars, following the formula below:

$$DI(S, S^*) = min\left(\frac{\text{median}\{\text{num-cites}(x) | x \in S\}}{\text{median}\{\text{num-cites}(x^*) | x^* \in S^*\}},\ 1 \right),$$

where $\text{num-cites}(x)$ is the number of citations for source x. We set an upper-bound of one, although in practice, we find this ratio to remains far below one for all measured generative research synthesis systems.

6.3.3 Verifiability

To evaluate the verifiability of the generated report given the retrieved set, we calculate the citation precision and claim coverage based on the definitions provided by prior work Gao et al. (2023); Worledge et al. (2024); Liu et al. (2023).

Citation Precision. Specifically, we measure sentence-level precision, where a citation is considered precise if the referenced source supports at least one claim made in the accompanying sentence. For a full report, citation precision is computed by averaging the precision of each citation in the report.

Claim Coverage. Claim coverage assigns a sentence-level score, assigning a value of one if the cited sources accompanying the sentence supports all claims made in the sentence. We make two adaptations to the original definition posed in prior work Gao et al. (2023); Worledge et al. (2024); Liu et al. (2023) for our long-form synthesis task. First, we relax the original claim coverage definition to consider a sliding window of sentences with supporting references, assigning a coverage value of 1 to each sentence that is either fully supported by the sources cited within the sentence or any cited source that is in a window of w preceding or following sentences. Additionally, since our task query provides context describing a paper, we consider this context as an implicitly cited reference for each sentence and compute claim coverage for each sentence by considering the explicitly cited sources, and the context provided by the query. We compute the citation coverage for the full report by averaging the value computed for each sentence. Our model-based evaluation uses an LLM judge to assess each entailment relation, following prior work Gao et al. (2023); Liu et al. (2023). We provide further details and the prompt in the Appendix Section 6.4 and Box 3 respectively.

6.4 Discussion of DeepScholar-bench Evaluation Framework

6.4.1 Reference Coverage

Figure 3 illustrates the distribution of important citations across the human exemplar reports in DeepScholar-Bench. For each exemplar, we used the LOTUS program shown in Figure 4 to identify which citations are *important* and therefore essential to include in a high-quality related work section. We then separate these important citations into two groups: those that appear on ArXiv (shown in red) and those that do not (shown in orange). The blue portion of each bar corresponds to *non-essential* citations, as determined by the same Lotus-based procedure.

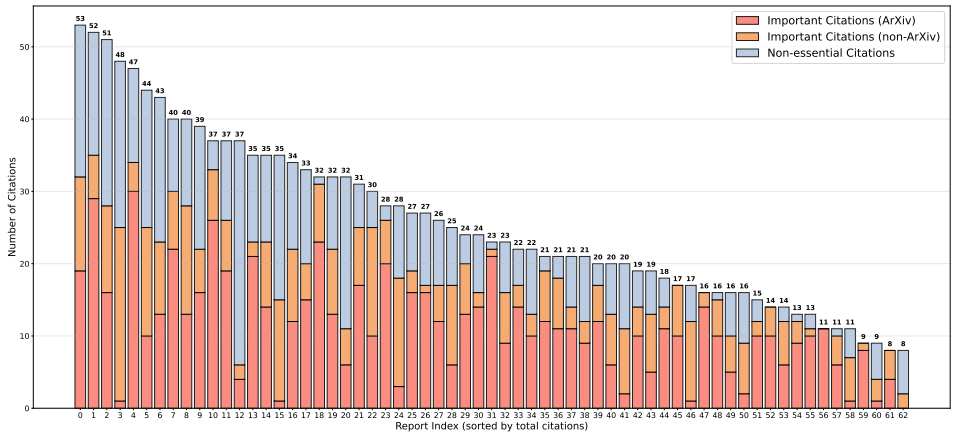

Figure 3: Citation importance breakdown in DeepScholar-Bench. Each bar corresponds to a single human exemplar related work section, sorted by the total number of citations. Bars are color-coded to indicate *important ArXiv citations* (red), *important non-ArXiv citations* (orange), and *non-essential citations* (blue).

The plot highlights two consistent trends. First, many exemplar related work sections contain a large number of non-essential citations. While such references may be useful for narrative flow or broader context, they are not indispensable. Non-essential citations can be somewhat subjective, depending on how authors choose to frame the story of their paper. In contrast, the important citations represent the "must-have" references i.e., the foundational works in the field that are necessary for situating the contribution. Second, we observe that the red segments (important ArXiv citations) are well distributed across exemplars, indicating that ArXiv is a reliable and sufficiently broad source for recovering many of the essential references.

```
query_in = "Carefully read the {title}, {abstract} and {
    related_work_section} of an academic paper. \
    Then consider the cited paper in question, given the title {
    cited_paper_title}, the {cited_paper_authors} and a snippet of
    its content, {cited_paper_content}.\
    Is the cited paper in question an important reference?\
    An important reference reflects a notable prior work that
    provides key information, which a good related works section for
    this paper must include.\
    A non-important reference is one that could be omitted or
    substituted with a different related work.\
    A non-important reference may be a tangential reference, an
    unimportant reference.\
    Alternatively, a non-important reference may be a relevant
    reference that reflects an important topic area, but the
    particular reference could be omitted or substituted with a
    different related work."



res = citations_df.sem_filter(query_in)


```

Figure 4: LOTUS program for Finding Important References

6.4.2 Ablation Study on Verifiability

In the main paper (Section 6.3.3, we reported verifiability metrics results using a sliding window of size $w = 1$ when computing claim coverage. That is, for each claim sentence, we considered a

citation to be valid if any of the references in the same sentence or within one sentence before or after sufficiently supported the claim. Here, we extend this analysis to study the effect of varying the window size. Specifically, we report the citation coverage achieved by different systems when the window size ranges from $w = 0$ (same-sentence only) up to $w = 5$ (five sentences before or after).

As shown in Figure 5, increasing the window size consistently improves citation coverage across all baselines. This is expected: the larger the window, the higher the probability that one of the cited references in the $[-w, +w]$ neighborhood of a claim provides sufficient support. However, we also note that very large window sizes are less desirable in practice, as they often correspond to references being far from the claims they are intended to support, reducing readability and making it harder for readers to verify the connection between claims and citations. Moreover, from Table 5, we see that real academic writing tends to be densely cited, with at least one citation on average per sentence in the human exemplars. Overall, the results of our ablation study highlight the trade-off between stricter precision ($w = 0$) and more lenient recall-oriented settings ($w \geq 1$).

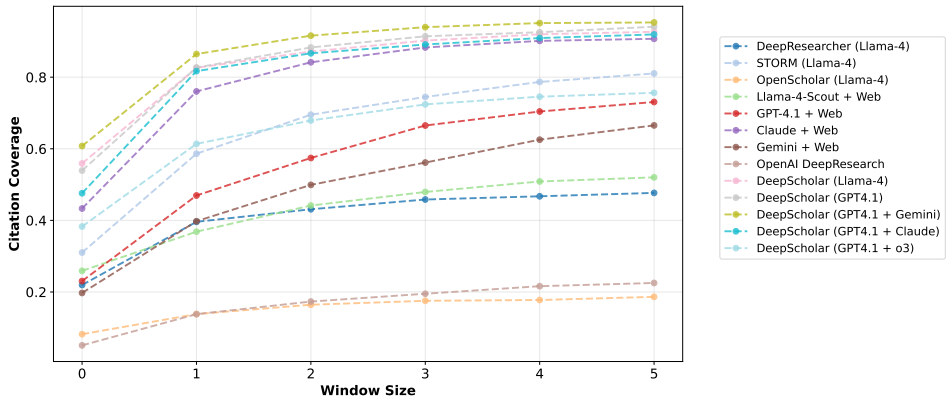

Figure 5: Ablation study on citation coverage with different window sizes. For each claim, we measure whether any citation within a sliding window of $[-w, +w]$ sentences supports it.

6.4.3 Document Importance Across Human Exemplars

In this section, we illustrate the distribution of document importance, measured by the number of citations of references in the human-written exemplars in DeepScholar-Bench. Figure 6 reports two histograms: (a) the distribution of citation counts across all references, and (b) the distribution restricted to references that appear on ArXiv. We plot the logarithm of citation counts, with values obtained from the OpenAlex API OpenAlex (2025), an open and widely used scholarly database that provides citation-level metadata. While citation counts in OpenAlex may not exactly match those from other sources such as Google Scholar, the relative counts are consistent, making it a reliable open-source alternative.

As shown in Figure 6, the distribution is highly skewed due to a small number of papers with exceptionally large citation counts (e.g., over 10k citations). These outliers inflate the mean citation values, resulting in relatively high averages compared to typical references (478.3 citations across all references and 647.6 for ArXiv-only references). In contrast, the median values are lower (31 for all references and 36 for ArXiv-only). This skew highlights the challenge of using citation counts as a proxy for importance, as the median citation count of references, among different human-written exemplars exhibits high variance.

6.5 Extended Description of DeepScholar-base

DeepScholar-base operates through three main stages: retrieval, filtering, and final generation (Figure 7).

Retrieval In this stage, an LLM generates Q search queries conditioned on the input abstract and summaries of prior retrievals. Each query is submitted to the configured search API (ArXiv, tavily, etc.) to obtain up to $search\_K$ relevant papers within the specified date range. The code and prompt used for this step are provided in Figure 8 and Box 4 respectively. This process is repeated N times.

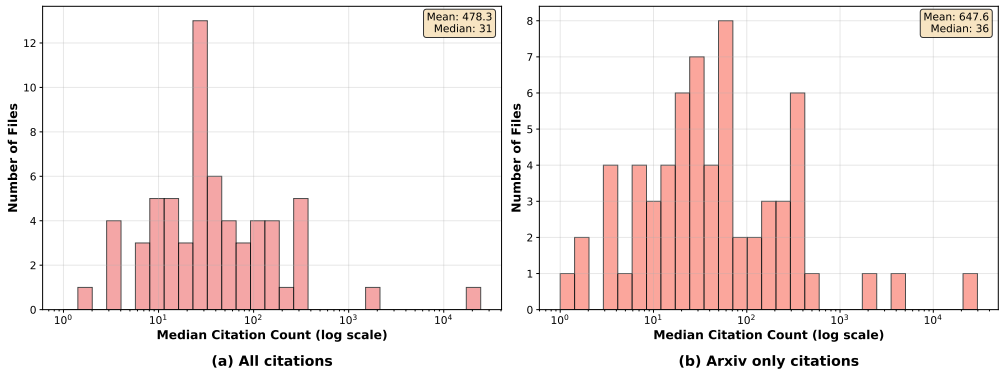

Figure 6: Distribution of citation counts (Document Importance) for references in human-written exemplars. Figure (a) shows all references, while Panel (b) restricts to ArXiv references only. Citation counts are plotted on a logarithmic scale.

Box 1: Prompt for Knowledge Synthesis- Organization

You are an intelligent, rigorous, and fair evaluator of scholarly writing quality and relevance. You will receive the title and abstract of a research paper, together with two candidate related-work sections (A and B) written for that paper. Do not consider the formatting of the text (e.g., LaTeX, markdown, etc.). Only consider the content.

Task: Decide which section—A or B—exhibits better organization and coherence.
How to judge (organization only) Ignore breadth of coverage, citation accuracy, and analytic depth.
Assess:
Logical structure – Clear introduction, grouping of related themes, and smooth progression of ideas.
Paragraph cohesion – Each paragraph develops a single topic and flows naturally to the next.
Clarity & readability – Minimal redundancy or contradictions; transitions guide the reader.
Signposting – Helpful headings, topic sentences, or discourse markers (if provided).

Pick the section that is easier to follow and better structured—no ties.

Paper under assessment: `[TITLE + ABSTRACT GO HERE]`
Candidate related-work section A `[RELATED WORK A TEXT GOES HERE]`
Candidate related-work section B `[RELATED WORK B TEXT GOES HERE]`

Output your answer as a JSON dictionary in the following format:
```
{"decision": "A" or "B", "explanation": "One sentence clearly explaining
the key differences between the two options and why the selected one is
preferred."}
```
Only output the dictionary, do not output any other text.

Filtering Retrieved results are refined using two semantic operators from LOTUS Patel et al. (2025); noa (2025b): `Sem-Filter` and `Sem-TopK`, which together select the top K most relevant papers. The code is given in Figure 9.

Final Generation The filtered set of papers is then aggregated via a `Sem-Agg` query to produce the final output. The corresponding code for this step is shown in Figure 10 with prompt in Box 5.

Unless otherwise specified, the pipeline parameters are set to $Q = 2$, $search\_K = 50$, $N = 2$, and $K = 30$.

6.6 Overview of Experimental Setup

6.6.1 Experimental Setup.

For each benchmarked method, we control the retrieval corpus by allowing each system to access the Web only through the ArXiv API arxiv (2025). We additionally avoid possible information

Box 2: Prompt for Reference Relevance Judgment

You are an intelligent, rigorous, and fair evaluator of scholarly writing quality and citation relevance. You will receive the title and abstract of a research paper under assessment, the ground-truth related-work section written by human experts, and the title and abstract of a candidate reference paper. Do not consider formatting (e.g., LaTeX, markdown, etc.). Only consider the content.

Task: Determine whether the candidate reference paper is relevant to the related-work section.
How to judge • Consider the main research topic and themes described in the related-work section.
• If the reference discusses similar ideas, prior work, or background, mark it as relevant (1).
• If the reference is off-topic or unrelated in scope, mark it as not relevant (0).
• Remember: You are only seeing the title and abstract of the reference, so the full content might be more relevant than it appears.

Paper under assessment: `[PAPER TITLE GOES HERE] [PAPER ABSTRACT GOES HERE]`
Ground-truth related-work section: `[RELATED WORK TEXT GOES HERE]`
Candidate reference paper: `[REFERENCE TITLE GOES HERE] [REFERENCE ABSTRACT GOES HERE]`

Return only the score in this format:
`### final score: <0 or 1>`

Box 3: Prompt for Attribution Validation

You are an intelligent and fair evaluator. Your task is to verify whether a given reference can support the provided claim.

Task: Given a claim and its associated set of references, determine whether the references sufficiently support all aspects of the claim.
CLAIM: `[CLAIM TEXT GOES HERE]`
REFERENCES: `[REFERENCE TEXT GOES HERE]`

Judgment Criteria: • If the references support the claim, return `1`.
• If the references do not support the claim, return `0`.
• Do not explain your answer or include any additional commentary.

Output Format:
`Answer: 1` or `Answer: 0`

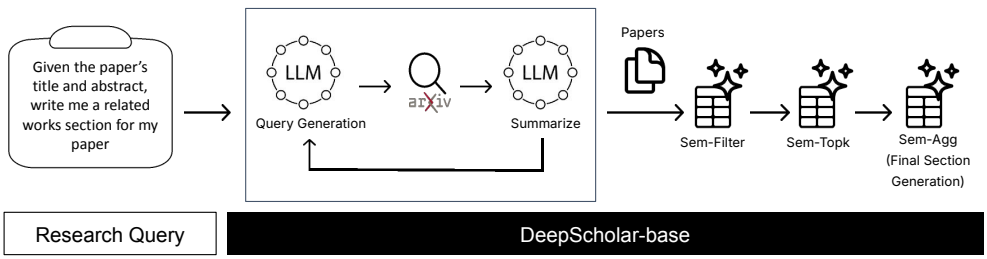

Figure 7: Overview of DeepScholar-base. The system iteratively writes queries and performs web search, before passing the search results through series of semantic operators using the LOTUS system for LLM-based data-processing, including a filtering step to discard irrelevant sources, a top-k ranking step to re-rank the most relevant sources, and a final aggregation step to generate the final report from all remaining sources.

leakage during search by filtering out any search results that were published after the query paper's publication date. We report results using GPT-4.1-2025-04-14 OpenAI (2025b) as the judge for Nugget Coverage, and a GPT-4o-2024-08-06 OpenAI (2025b) judge for Organization, Relevance Rate, Reference Coverage, Citation Precision and Claim Coverage. We report the Organization score as a win rate including ties, we report the strict all score for Nugget Coverage, and we report Claim

```python
from lotus import web_search

class Query(BaseModel):
    queries: list[str]

# Generate the Queries
queries = get_completion(
    lm,
    query_generation_instruction.format(number_of_queries=num_queries),
    f"Topic: {topic}, Background: {background}",
    response_format=Query,
).queries

# Search. corpus = ArXiv/Tavily etc.
paper_dfs = []
for query in queries:
    paper_dfs.append(web_search(corpus, query, search_K))

papers = pd.concat(paper_dfs)

```

Figure 8: Retrieval stage: query generation and batched search.

```python
instruction = (
    "given the article's abstract: {snippet}, "
    "is the article relevant to the specific interests in the user's query: {user_query}."
)

res_df = docs_df.sem_filter(
    instruction.format(user_query=topic, snippet="{snippet}"),
    strategy="cot"
)

res_df = res_df.sem_topk(
    instruction.format(user_query=topic, snippet="{snippet}"),
    strategy="cot", k=K
)

```

Figure 9: `Sem-Filter` and `Sem-TopK` code for Filtering Step in DeepScholar-base

Coverage with a window size of $w = 1$. For all Retrieval Quality metrics, we consider the retrieved set of each given report as the set of any valid ArXiv links found within the report. To measure Document Importance, we use the OpenAlex OpenAlex (2025) API to recover citation information. For each metric, we report an average over all reports.

6.6.2 Baselines

We briefly overview all of the baseline systems we evaluate, and we provide further details in the Appendix Section 6.7.

6.6.3 Open-source Research Systems

We evaluate three state-of-the-art open-source systems, DeepResearcher Zheng et al. (2025), STORM Shao et al. (2024) and OpenScholar Asai et al. (2024). For each, we run these systems using the Llama-4-Scout-17B-16E-Instruct model noa (2025c), which we serve with 4 A100 GPUs using vLLM Kwon et al. (2023).

```
agg_instruction = section_writer_instructions.format(
    topic=topic,
    section_instructions=section_instructions,
    existing_content=existing_content,
    context="{context}",
)

res: pd.DataFrame = res_df.sem_agg(
    agg_instruction, suffix="summary", group_by=group_by
)

```

Figure 10: `Sem-Agg` for final generation in DeepScholar-base

Box 4: Prompt to generate ArXiv Search Queries

You are an expert technical writer generating targeted search queries to retrieve the most relevant arXiv papers for a technical report section.
<Report topic>
{{topic}}
</Report topic>
<Background>
{{background}}
</Background>
<Task>
Generate {number_of_queries} distinct arXiv search queries to comprehensively cover the section topic.
Today's date is date.
Guidelines for queries: 1. Each query should use 1–10 keywords, focusing on a single, specific concept related to the topic.
2. Ensure queries explore different or complementary aspects of the topic to maximize coverage.
3. Use terminology and phrasing likely to match arXiv paper titles or abstracts.
4. Avoid overly broad or generic queries; be as precise as possible.
5. Queries should cover all the key aspects of the topic. Background information may be used to inform the queries.
6. DO NOT create a complex query using AND/OR etc. Keep it simple
The goal is to maximize the relevance and diversity of retrieved papers.
</Task>

DeepResearcher Zheng et al. (2025) leverages trained agents to navigate, browse and synthesize information from the web. To train an agent, this work uses end-to-end reinforcement learning and trains Qwen2.5-7B-Instruct Qwen et al. (2025). In our benchmarks, we evaluate DeepResearcher using both the released, trained model from the authors, and using Llama-4-Scout-17B-16E-Instruct model noa (2025c) as the core LLM. We report the better performing baseline of these two, which we find in our experiments to be the Llama-4-Scout-17B-16E-Instruct backbone.

STORM Shao et al. (2024) studies the problem of how to apply LLMs to write grounded, organized long-form articles (e.g., Wikipedia articles) from scratch. The system involves a pre-writing stage that discovers diverse research perspectives on a topic by stimulating conversations between multiple agents and leveraging web documents.

OpenScholar Asai et al. (2024) builds a specialized retrieval-augmented LLM system for literature synthesis and scientific queries. This method includes a trained retriever from the pre-indexed peS2o Soldaini et al. (2024) corpus, consisting of 45 million open-access academic papers up until October 2024, as an initial retrieval source before using web search. In our experiments, we benchmark the system using this pre-indexed corpus and limit web search to the ArXiv API.

6.6.4 Search AI's

We evaluate the following models: Llama-4-Scout-17B-16E-Instruct noa (2025c), GPT-4.1-2025-04-14 OpenAI (2025b), o3-2025-04-16 OpenAI (2025b), Claude-opus-4-20250514 Anthropic (2025b),

Box 5: Sem-Agg Instruction for final generation and summarization

You are an expert technical writer crafting one section of a technical report.
<User Query>
{topic}
</User Query>
<Section instructions>
{section_instructions}
</Section instructions>
<Existing section content (if populated)>
{existing_content}
</Existing section content>
<Source material>
{context}
</Source material>
<Citation Guidelines>
- Use [X] format where X is the {citation_number}
- Place citations immediately after the sentence or paragraph they are referencing (e.g., information from context [3]. Further details discussed in contexts [2][7].).
- If urls are given in existing section content, rewrite them exactly if using information related to the url.
- Make sure to provide citations whenever you are using information from the source material. This is a MUST.
- Cite as many sources as possible.
- Make sure to retain the citation numbers from the input context. - Provide in-line citations only. You do not need a reference section at the end.
<Citation Guidelines>
<Guidelines for writing>
1. If the existing section content is populated, write a new section that enhances the existing section content with the new information. If not, write a new section from scratch.
2. Provide groundings in the source material for all facts stated.
3. When using information from a given source, make sure to cite the source.
4. If a table or list would enhance understanding of a key point, and if so, include one.
5. Make sure to follow the user query strictly.
</Guidelines for writing>
<Writing style>
1. Content Requirements:
- Ground all facts in the source material and provide citations.
- Maintain an academic, technical focus throughout. No marketing language
- Address potential counter-arguments where relevant.
2. Structure and Formatting:
- Use Markdown formatting.
- Begin with ## for section title (Markdown format) and other headings as needed.
- Strict 1500-2000 word limit
- Use simple, clear language appropriate for academic writing.
</Writing style>
<Quality checks>
- Exactly 1500-2000 words (excluding title and sources)
- No preamble prior to creating the section content
- Cite as many sources as possible.
</Quality checks>

and Gemini-2.5-pro Gemini (2025b). We augment each with search capabilities to ArXiv arxiv (2025), and use the popular ODS framework Alzubi et al. (2025) to allow the LLM to make tool calls to the search API.

6.6.5 Commercial Systems.

We focus our evalution of commercial generative research synthesis systems on OpenAI's o3-deep-research OpenAI (2025b), which provides a public API allowing for our evaluation.

6.6.6 DeepScholar-base

Similar to our evaluation of search AI's we evaluate DeepScholar-base with the following models: Llama-4-Scout-17B-16E-Instruct noa (2025c), GPT-4.1-2025-04-14 OpenAI (2025b), o3-2025-04-16 OpenAI (2025b), Claude-opus-4-20250514 Anthropic (2025b), and Gemini-2.5-pro Gemini (2025b). For each of these baselines, we also use the same or a weaker model, either Llama-4 or GPT-4.1, to perform semantic filtering and top-k operators. We limit the method to two round of search, each with at most 2 queries.

6.7 Extended Description of Baselines

We provide an extended description of each benchmarked method, including relevant implementation details, and parameters used in our evaluation.

6.7.1 DeepResearcher

The DeepResearcher pipeline follows a structured tool-augmented reasoning framework designed for iterative web-based information retrieval. The system mandates explicit reasoning before any tool invocation, with reasoning encapsulated in `<think>` tags to ensure interpretability and control. After reasoning, the model generates a JSON-formatted request specifying the "web search" tool and its query. These queries are executed via the Lotus Search API, which we replaced with an ArXiv-specific search interface to provide a controlled retrieval API for our evaluation. Retrieved results are returned in a structured format containing the title, URL, and snippet, and are stored in memory for reference across subsequent reasoning steps. This iterative process continues until the model determines that sufficient evidence has been gathered, after which a synthesized final response is produced.

For our experiments, we used Llama-4-Scout-17B-16E-Instruct as the base model, replacing the originally proposed DeepResearcher-7b, since it demonstrated consistently better retrieval-augmented reasoning performance in our experiments. The prompt was slightly modified to align with LLama-4 prompt style as detailed in Box 6. The retrieval depth was set to 10 sources per query, which is the default in the system and provides a balanced trade-off between coverage and efficiency. We restricted each query to a single rollout with a maximum of 10 steps, following the DeepResearcher defaults; this limit is generous as most rollouts converge in fewer than three steps, but it ensures the system has headroom for more complex queries. The default web search API was replaced with ArXiv search to comply with our benchmark settings.

6.7.2 Openscholar

The OpenScholar pipeline follows a four-stage process: initial retrieval, response and feedback generation, iterative refinement, and citation verification. In the first stage, text segments are retrieved from a fixed index using a contriever model, which encodes texts and retrieves passages based on semantic similarity. These passages are reranked and used to generate an initial draft response, where citations are aligned with the supporting passages. The second stage introduces feedback generation, where the model produces up to three feedback statements highlighting potential improvements in the draft, such as missing content or organization issues; if additional evidence is required, retrieval queries are issued. The third stage iteratively refines the response by conditioning on the previous draft, retrieved passages, and newly added evidence, yielding improved responses at each step until feedback has been fully incorporated. Finally, citation verification ensures that all citation-worthy statements are adequately grounded in the retrieved sources, inserting additional citations where necessary without removing content.

Box 6: Revised DeepResearcher System Prompt optimized to work with Llama-4-Scout-17B-16E-Instruct

```
\#\# Background information
* Today is {strftime("%Y-%m-%d", gmtime())}
* You are Deep AI Research Assistant
The question I give you is a complex question that requires a *deep research* to answer.

I will provide you with two tools to help you answer the question:
* A web search tool to help you perform google search. Tool call format:
\begin{verbatim}
{{"name": "web_search", "arguments": {{"query": ["<query1>","<query2>","<query3>"]}}}}

* A webpage browsing tool to help you get new page content. Tool call format:
{{"name": "browse_webpage", "arguments": {{"url_list": ["<url1>","<url2>","<url3>"]}}}}
```

You don't have to answer the question now, but you should first think about the research plan or what to search next.
Your output format should be one of the following two formats:

```
<think>
YOUR THINKING PROCESS
</think>
<answer>
YOUR ANSWER AFTER GETTING ENOUGH INFORMATION
</answer>
```

or

```
<think>
YOUR THINKING PROCESS
</think>
<tool_call>
YOUR TOOL CALL WITH CORRECT FORMAT
</tool_call>
```

You should always follow the above two formats strictly. You will be heavily penalized if you do not follow the format strictly. Only output the final answer (in words, numbers or phrase) inside the <answer></answer> tag, without any explanations or extra information. If this is a yes-or-no question, you should only answer yes or no.

For consistency with other baselines, we employ the Llama-4-Scout-17B-16E-Instruct model for generation. The retrieval pipeline initially collects 100 text segments from peS2o_v3 using the default pes2o_contriever model. The reranker used is `OpenScholar_Reranker`, also kept at its default setting. To align parameterization across baselines, we increase the number of sources used in generation (`top_n`) from 10 to 30. Furthermore, the default search API is replaced with the arXiv API, to provided a controlled retreival corpus and API in our experiments.

6.7.3 Search AI

Search AIs are implemented using the open-source OpenDeepSearch (ODS) framework that enables deep web search and retrieval. In particular, the ODS ReAct Agent (instantiated from the `smolagents.ToolCallingAgent`) is used along with the search agent as an external tool. At each reasoning step, the ReAct agent can either invoke the search agent through a `web_search` action or decide to produce a `final_answer`. The search agent interfaces with the search API to fetch relevant academic articles given a query, after which an LLM generates concise summaries of the retrieved content. To maintain consistency with the benchmark setting, the standard search API was replaced with the arXiv API. The regular search agent fails when tasked with full abstract queries; hence the ReAct-based agent was employed, which generates shorter, more effective searchable queries. The agent keeps track of retrieved results across turns, allowing references to past evidence during the reasoning process. After a maximum of 5 iterations, the agent is compelled to conclude with a final response, ensuring bounded computational steps.

For the parameterization of the Search AIs, we set the search agent to retrieve 30 results per query, which is more generous than the default in order to establish fair comparability with other baselines. The reranker parameter was left at infinity, aligning with its default configuration, to avoid prematurely constraining candidate results. The maximum iteration limit was fixed at 5, consistent with the default setup of the ODS framework, providing sufficient exploration without excessive search depth. The ReAct prompt was slightly modified to tailor to the specific use of the ArXiv search API, as presented in Box 7, 8 and 9.

6.7.4 STORM

The STORM pipeline follows a structured multi-stage process to generate comprehensive, Wikipedia-style articles from a given topic. First, related Wikipedia articles are retrieved and their TOCs are clustered to identify candidate perspectives, which act as anchors for exploration. This is followed by Simulated Multi-turn Conversations where an LLM plays both the question-asking and answering roles, querying a retrieval module and synthesizing evidence-based responses. Parallel to this, the model generates a draft outline purely from its parametric knowledge in the Draft Outline Generation stage. The outline is then refined by grounding it with retrieved evidence and conversation outputs. In the final step, each section is drafted with explicit inline citations drawing on both parametric knowledge and retrieved references. All the sections are concatenated together to form the final result.

For parameter settings, we used STORM's default configurations wherever possible to preserve fidelity to its design: a maximum of 3 turns per perspective, 3 perspectives, and up to 3 search queries per turn. For search, we considered the top 15 results for each query, ensuring a reasonable breadth without overwhelming the pipeline. To make STORM comparable with other baselines, we raised the number of collected references per section title to 30 (more generous than the default), as this allows for richer evidence integration during drafting. Importantly, we replaced the original search API with arXiv search to control the retrieval API for our benchmark settings. Finally, we use Llama-4-Scout-17B-16E-Instruct as the base model.

6.7.5 OpenAI's DeepResearch

We use OpenAI's DeepResearch system based on the `o3-deep-research` model with a custom MCP to only search ArXiv and return $n = 30$ results per query. To prevent the model from getting search results after the given paper was uploaded, the MCP used a custom endpoint to set the latest date that it should retrieve. All other settings were set to default values.

6.8 Extended Experimental Analysis and Ablations

6.9 Main Results

Table 1 provides detailed summary of each method's performance scores on all metrics across three key dimensions, knowledge synthesis, retrieval quality and verifiability. We also provide metadata statistics characterizing the generated reports of each benchmarked method in Table 5, as well as statistics related to our evaluation metrics in Table 6. Overall, our evaluation demonstrates two key findings, which we discuss in detail below: first, existing generative research synthesis systems demonstrate significant headway for improvement, and second, DeepScholar-base provides a strong baseline for generative research synthesis.

6.9.1 Generative Research Synthesis Systems Demonstrate Large Room for Improvement.

From Table 1, we see that no method is able to achieve a score greater than .19 across all metrics. Moreover, on several key metrics, including nugget coverage, reference coverage and document importance, each evaluated method's performance remains well below .45. This reflects the inherent difficulty of the generative research task provided by DeepScholar-bench, which requires systems to navigate the live web, reasoning about relevance and importance of documents to perform retrieve sources and then surface key information into a coherent report that answers the query.

We now analyze each evaluated dimension, comparing performance of the open-source research systems, search AI's and commercial systems to the human-written exemplars. On knowledge synthesis, we see that OpenAI DeepResearch offers the best performance compared to all other prior

Table 4: Ablation Study Comparing The Effect of Different Retrieval APIs.

	Knowledge Synthesis		Retrieval Quality			Verifiability	
	Org	Nug. Cov.	Rel. Rate	Ref Cov.	Doc Imp.	Cite-P	Claim Cov ($w = 1$)
DeepScholar-base (GPT-4.1, Claude)							
arxiv.org Retrieval	.786	.370	.586	.167	.007	.936	.817
parallel.ai Retrieval	.865	.444	.675	.160	.017	.846	.781
taviliy.com Retrieval	.929	.327	.550	.070	.015	.711	.578
Oracle Retrieval (arxiv.org)	.782	.487	.686	1.000	1.000	.955	.899
Oracle Retrieval (All)	.778	.528	.680	1.000	.822	.941	.828
DeepScholar-base (Llama-4)							
arxiv.org Retrieval	.254	.262	.421	.103	.008	.648	.826
parallel.ai Retrieval	.246	.265	.559	.114	.015	.223	.543
taviliy.com Retrieval	.111	.229	.532	.030	.016	.442	.676
Oracle Retrieval (arxiv.org)	.202	.316	.681	1.000	1.000	.658	.868
Oracle Retrieval (All)	.198	.350	.693	1.000	.822	.796	.890

methods on both Organization, with a score of .857, and Nugget Coverage, with a score of .392. OpenAI DeepResearch, as well as the o3, Claude and Gemini search AI's achieve relatively high Organization scores compared to human-written exemplars. However, on Nugget Coverage all prior methods scores below .40. This demonstrates that while existing systems, especially those using state-of-the-art models, can generate well-organized and coherent summaries, they still struggle to extract and surface key facts, a crucial capability for synthesis tasks.

Turning our attention to the retrieval quality performance of prior methods, we once again find significant room for improvement. Once again, OpenAI DeepResearch offers the strongest performance among the other benchmarked prior methods on Relevance Rate, Reference Coverage and Document Importance, but still far from saturates performance. While it's Relevance Rate shows strong performance, exceeding that of the human exemplars with a score of .629, it's Reference Coverage and Document Importance scores remain exceedingly low: .187 and .124 respectively. This demonstrates that while state-of-the-art generative research synthesis systems are capable of retrieving relevant sources, they still struggle to find a comprehensive set of notable sources and fall short compared to the ability of human experts.

Lastly, we analyze the verifiability performance of prior methods, we see that OpenAI DeepResearch is outperformed on both Citation Precision and Claim Coverage by the search AI's with GPT4.1, o3, Claude and Gemini models. The Claude search AI offers the highest Citation Precision, a score of .701 and Claim Coverage, a score of .760. Meanwhile, OpenAI's DeepResearch as well as the all other prior methods are unable to achieve a Citation Precision score beyond .5 and a Claim Coverage score beyond .6. We also note that the human-written exemplars appear to exhibit rather low Citation Precision and Claim Coverage scores, however these scores are not comparable to the metric measured for the LLM-based systems since our method for measuring verifiability metrics likely under-estimate the actual verifiability of human writing[1]. Overall, we see that prior LLM-based systems exhibit significant headroom for improvement.

6.9.2 DeepScholar-base Provides a Strong Baseline for Generative Research Synthesis.

We compare the performance of DeepScholar-base to the commercial OpenAI DeepResearch system, search AI's and open-source research systems, finding that DeepScholar-base provides competitive performance against each group of prior methods, offering a strong baseline for generative research synthesis.

First, we see that DeepScholar-base (GPT-4.1, o3) is competitive with OpenAI DeepResearch, achieving a similar or higher Organization, Nugget Coverage, Relevance Rate, Reference Coverage, Citation Precision and Claim Coverage scores. Notably, DeepScholar-base offers significantly higher

verifiability than OpenAI DeepResearch, with $1.5 - 2.3\times$ higher Citation Precision and $4.4 - 6.3\times$ higher Claim Coverage. However, DeepScholar-base's document importance scores still remain especially low compared to OpenAI DeepResearch, representing significant room for improvement.

Next, we compare the performance of DeepScholar-base to the search-AI's, finding that for each evaluated model, DeepScholar-base consistently offers improved performance compared to the corresponding search AI. Specifically, averaged across all 5 baselines with different models for the Search AI's and DeepScholar-base method, DeepScholar-base offers $1.28\times$ higher Organization, $1.29\times$ higher Nugget Coverage, $1.06\times$ higher Relevance Rate, $2.03\times$ higher Reference Coverage $1.64\times$ higher Citation Precision, $1.62\times$ higher Citation Recall.

Lastly, we compare DeepScholar-base (Llama-4) to the open-source research systems, all run with the Llama-4 model as well. We see that the prior open-source research systems exhibit trade-offs among the Knowledge Synthesis, Retrieval Quality and Verifiability dimensions. Specifically, OpenScholar achieves the highest Knowledge Synthesis scores, on both Organization and Nugget Coverage, DeepResearcher achieves the highest Relevance Rate and Reference Coverage on Retrieval Quality, with all systems attaining only very low Document Importance scores, and on Verifiability, DeepResearcher offers the highest Citation-Precision while STORM offers the highest Claim Coverage. In comparison, DeepScholar-base offers strong performance across each dimension. Specifically, Compared to the best-performing prior open-source methods for each metric, DeepScholar-base offers competitive Knowledge Synthesis performance, $1.09\times$ higher Relevance Rates and $2.18\times$ higher Reference Coverage for retrieval, and $2.08\times$ higher Citation Precision and $1.41\times$ higher Claim Coverage scores for verifiability.

Overall, the strong *relative* performance of DeepScholar-base likely reflects the efficiency of the data-processing semantic operators Patel et al. (2025) that DeepScholar-base uses to perform LLM-based filtering, ranking and summarization of sources to generate its report. Notably, DeepScholar-base still demonstrates significant room for improvement and far from saturates DeepScholar-Bench, especially on key Knowledge Synthesis and Retrieval Quality metrics, including Nugget Coverage, Reference Coverage, Document Importance, which represent opportunities for future work.

6.10 Understanding Opportunities for Improvement.

In order to further analyze performance and existing opportunities for improvement, we conduct an ablation study, where we consider different retrievers as well as two oracle retriever settings. Table 4 shows these results for two evaluated DeepScholar-base methods, DeepScholar-base (GPT-4.1, Claude) and DeepScholar-base (Llama-4). The table show the performance of either using three different retrieval APIs, including arxiv.org, the default used in our main results, parallel.ai and tavily.com. In addition the table shows to oracle settings for either DeepScholar-base method: the Oracle Retrieval (arxiv.org) setting, provides the system with the ArXiv references from the human-written exemplars labeled as "Important" following our methodology for evaluating Reference Coverage. The Oracle Retrieval (All) setting, provides the system with *any* from the human-written exemplars labeled as "Important" following the same methodology, including both references form ArXiv and once that are found elsewhere.

Overall,the results shown in Table 4 demonstrate that the performance limitations of existing systems for generative research synthesis lie in both their retrieval capabilities to find and select high-quality references sets, as well as their synthesis abilities to surface key facts and extract insights given the retrieved documents.

First, we see that given either oracle retrieval setting, the DeepScholar-base (GPT-4.1, Claude) method nearly saturates performance on Retrieval Quality and Verifiability metrics, whereas the same method using the arxiv.org, parallel.ai or taviliy.com retrievers obtain lower scores on each of these metrics. This finding demonstrates a significant opportunity to improve performance of generative research systems through improvements to the retrieval method. Specifically, existing systems struggle to find a diverse and holistic set of notable sources, reflected by their especially low Reference Coverage and Document Importance scores.

Additionally, we also see that oracle retrieval settings for either DeepScholar-base method attain higher Nugget Coverage, improving the score by up to $1.62\times$ compared to the rspective arxiv.org, parallel.ai or tavily.ai retreival settings. However, we note that the oracle retreival methods still far from saturate Nugget Coverage, with the DeepScholar-base (GPT-4.1, Claude) Oracle Retrieval (All)

Table 5: Report Statistics.

	Report Length			Citations	
	Chars	Words	Sentences	# Unique Refs	# Inline Citations
Human-written Exemplars					
Human-written Exemplars	4381	497	28	23	27
Open Source Research Systems					
DeepResearcher (Llama-4)	2573	319	35	8	7
STORM (Llama-4)	2766	381	31	18	21
OpenScholar (Llama-4)	3513	483	26	9	19
Search AI's					
Search AI (Llama-4-Scout)	1968	258	16	9	5
Search AI (GPT-4.1)	3168	404	16	10	61
Search AI (o3)	3844	501	24	11	16
Search AI (Claude)	3977	499	27	13	8
Search AI (Gemini)	2810	395	19	6	8
Commercial Systems					
OpenAI DeepResearch	6577	864	74	17	6
DeepScholar Baseline					
DeepScholar-base (Llama-4)	3864	402	58	21	19
DeepScholar-base (GPT-4.1)	14470	1492	167	19	56
DeepScholar-base (GPT-4.1, o3)	5905	642	69	16	20
DeepScholar-base (GPT-4.1, Claude)	12287	1332	136	23	35
DeepScholar-base (GPT-4.1, Gemini)	6118	663	81	22	28

Table 6: Statistics Related to Evaluation Metrics.

	Avg. value over human-written exemplars	Relevant Metric
# Important References from ArXiv.org	11.47	Ref. Cov.
Median number of citations per reference from ArXiv.org	647.5	Doc. Imp.

attaining a modest score of .528. This demonstrates that even with high retrieval quality, existing LLM systems still struggle to effectively surface important facts and synthesize important insights.

6.11 Human Agreement Study

Finally, we study how well our LLM-based evaluation aligns with human judgments, a critical question to validate effectiveness of our fully automated evaluation approach. Overall, we find that each of the metrics we introduce for the DeepScholar-bench task exhibit high agreement between LLM-based judgments and human annotations. We collect over 200 expert annotations, and Table 2 shows the agreement score between human and LLM labelers for organization pairwise comparisons, nugget importance labels, graded relevance scores and reference importance labels. Overall, the results demonstrate above 70% agreement scores across each, reflecting the effectiveness of the DeepScholar-bench evaluation approach.

Box 7: Revised ODS ReAct Agent prompt for only web search tool calling

You are an expert assistant who can solve any task using tool calls. You will be given a task to solve as best you can. To do so, you have been given access to some tools. Never use facts without verification and only cite the sources returned by the tool.
The tool call you write is an action: after the tool is executed, you will get the result of the tool call as an "observation". This Action/Observation can repeat N times, you should take several steps when needed. You can use the result of the previous action as input for the next action. The observation will always be a string containing the search results.
To provide the final answer to the task, use an action blob with "name": "final_answer" tool. It is the only way to complete the task, else you will be stuck on a loop. So your final output should look like this: Action:

```
{
  "name": "final_answer",
  "arguments": {"answer": "insert your final answer here"}
}
```

Here are a few examples using notional tools: —

```
Task: "What historical event happened closest in time to the invention of the telephone:
the American Civil War or the establishment of the Eiffel Tower?"
Action:
{
  "name": "web_search",
  "arguments": {"query": "year of telephone invention"}
}
Observation: "The telephone was invented in 1876."
Action:
{
  "name": "web_search",
  "arguments": {"query": "year American Civil War ended"}
}
Observation: "The American Civil War ended in 1865."
Action:
{
  "name": "web_search",
  "arguments": {"query": "year Eiffel Tower established"}
}
Observation: "The Eiffel Tower was completed in 1889."
Action:
{
  "name": "final_answer",
  "arguments": {"answer": "The historical event closest in time to the invention of the telephone is the
   Civil War (11 years apart)."}
}
---
Task: "Which country has a higher population density: Japan or India?"
Action:
{
  "name": "web_search",
  "arguments": {"query": "population and area of Japan"}
}
Observation: "Japan has a population of 125 million and an area of 377,975 square kilometers."
Action:
{
  "name": "web_search",
  "arguments": {"query": "population and area of India"}
}
Observation: "India has a population of 1.38 billion and an area of 3,287,263 square kilometers."
Action:
{
  "name": "final_answer",
  "arguments": {"answer": "India has a higher population density (419.6 people/km²) than Japan (330.7 pe
}
```

Box 8: Prompt for ODS(continued)

```
---
Task: "Which country hosted the first FIFA World Cup, and in what year?"

Action:
{
  "name": "web_search",
  "arguments": {"query": "country hosted first FIFA World Cup"}
}
Observation: "Uruguay hosted the first FIFA World Cup."

Action:
{
  "name": "web_search",
  "arguments": {"query": "year of first FIFA World Cup"}
}
Observation: "The first FIFA World Cup was held in 1930."

Action:
{
  "name": "final_answer",
  "arguments": {"answer": "Uruguay hosted the first FIFA World Cup in 1930."}
}

---
Task: "Who invented the light bulb, and what company did he later establish?"

Action:
{
  "name": "web_search",
  "arguments": {"query": "inventor of the light bulb"}
}
Observation: "Thomas Edison invented the light bulb."

Action:
{
  "name": "web_search",
  "arguments": {"query": "company founded by Thomas Edison"}
}
Observation: "Thomas Edison founded General Electric."

Action:
{
  "name": "final_answer",
  "arguments": {"answer": "Thomas Edison invented the light bulb and later established General Electric.
}

---
Task: "Which Shakespeare play contains the line \"All the world's a stage,\" and how many years ago was
today is 2024?"

Action:
{
  "name": "web_search",
  "arguments": {"query": "Shakespeare play All the world's a stage"}
}
Observation: "The line is from \"As You Like It.\""

Action:
{
  "name": "web_search",
  "arguments": {"query": "year As You Like It first performed"}
}
Observation: "\"As You Like It\" was first performed in 1603."
```

Box 9: Prompt for ODS(continued)

```
Action:
{
  "name": "calculate",
  "arguments": {"expression": "2024 - 1603"}
}
Observation: "421 years."

Action:
{
  "name": "final_answer",
  "arguments": {"answer": "\"As You Like It\" contains the line \"All the world's a stage\" and was firs
  in 1603."}
}
```

Above examples were using notional tools that might not exist for you. You only have access to these tools:

```
{%- for tool in tools.values() %}
- {{ tool.name }}: {{ tool.description }}
    Takes inputs: {{tool.inputs}}
    Returns an output of type: {{tool.output_type}}
{%- endfor %}

{%- if managed_agents and managed_agents.values() | list %}
```

Here are the rules you should always follow to solve your task: 1. ALWAYS provide a tool call, else you will fail. 2. Always use the right arguments for the tools. Never use variable names as the action arguments, use the value instead. 3. Call a tool only when needed: do not call the search agent if you do not need information, try to solve the task yourself. If no tool call is needed, use final_answer tool to return your answer. 4. Never re-do a tool call that you previously did with the exact same parameters. 5. Always cite sources using [X] format where X is the citation number. 6. Place citations immediately after the sentence or paragraph they are referencing. 7. Make sure to provide citations whenever using information from the source material. 8. Cite as many sources as possible. 9. Create a reference section at the end of your final answer.
Now Begin! If you solve the task correctly, you will receive a reward of $1,000,000.