PRISM: Fine-Grained Paper-to-Paper Retrieval with Multi-Aspect-Aware Query Optimization

Anonymous ACL submission

Abstract

Scientific paper retrieval, particularly framed as document-to-document retrieval, aims to identify relevant papers in response to a long-form query paper, rather than a short query string. Previous approaches to this task have focused on abstracts, embedding them into dense vectors as surrogates for full documents and calculating similarity across them, although abstracts provide only sparse and high-level summaries. To address this, we propose PRISM, a novel document-to-document retrieval method that introduces multiple, fine-grained representations for both the query and candidate papers. In particular, each query paper is decomposed into multiple aspect-specific views and individually embedded, which are then matched against candidate papers similarity segmented to consider their multifaceted dimensions. Moreover, we present SciFullBench, a novel benchmark in which the complete and segmented context of full papers for both queries and candidates is available. Then, experimental results show that PRISM improves performance by an average of 4.3% over existing retrieval baselines.

1 Introduction

Information Retrieval (IR) is the task of searching for query-relevant documents from a large external corpus, evolving from sparse keyword matching (Sparck Jones, 1972; Robertson et al., 1995) to dense representation-based similarity (Karpukhin et al., 2020; Izacard et al., 2021). Notably, in the era of Large Language Models (LLMs) (Achiam et al., 2023; Team et al., 2023; Dubey et al., 2024; DeepSeek-AI et al., 2025), IR has become increasingly crucial, which enables LLMs to utilize up-to-date external information (Lewis et al., 2020).

In contrast to conventional retrieval tasks, whose queries are short (such as questions or keywords), scientific paper retrieval poses unique challenges. Specifically, queries are long-form, structured documents that encapsulate diverse aspects, ranging from research motivation and proposed methodology to experimental design and empirical findings. Also, relevance in this setting is inherently multi-faceted, as a paper may be considered relevant for various reasons, such as pursuing similar research objectives or employing comparable methodologies. However, primarily due to the limited context lengths of embedding models, prior work has focused on abstract-level representations as a proxy for full papers (Cohan et al., 2020; Yasunaga et al., 2022; Ostendorff et al., 2022; Singh et al., 2023; Zhang et al., 2023a), by fine-tuning models using abstract pairs connected via citation relationships. While effective to some extent, they are suboptimal for capturing deeper contextual relationships between papers, resulting in limited performance, especially on tasks such as identifying complementary work and generating literature reviews, where understanding the full paper beyond surface-level abstracts is essential (Asai et al., 2024; Baek et al., 2024; Chamoun et al., 2024; Jin et al., 2024).

However, it is non-trivial to represent a full scientific paper and retrieve it. On the one hand, full papers sometimes exceed 100K tokens in length, far surpassing the context limits of most embedding

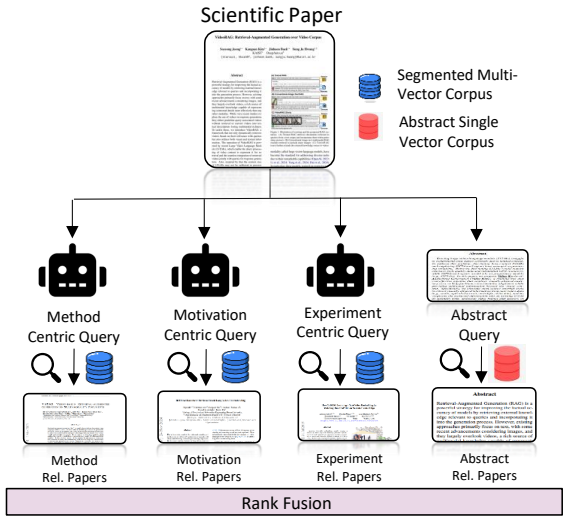

Figure 1: Conceptual illustration of our framework, PRISM.

models. Even when a full paper can be encoded into a single vector, representing its diverse aspects (such as motivation, methods, and experiments) within a single embedding may lead to oversimplification and blur fine-grained distinctions, especially in cases where similarity lies in a specific aspect.

To this end, we propose a novel paper-to-paper retrieval framework, PRISM, which handles full papers from two different angles. Specifically, on the query side, it generates multiple aspect-specific queries, each capturing a distinct perspective of the given paper, and we operationalize this with multi-agent query optimization: LLM agents (each specialized in a certain aspect) independently analyze the paper and formulate their corresponding queries. Also, on the corpus side, it segments candidate documents into section-level representations stored in separate corpora, allowing each aspect-specific query to perform targeted retrieval based on the most relevant parts of the paper. Lastly, the retrieval results from all queries are aggregated at the rank level, producing a single unified ranking.

To validate this, we construct SciFullBench, a new benchmark suite that enables paper-to-paper retrieval with complete papers for both queries and retrieval targets in ML and NLP domains, since existing benchmarks are primarily designed for abstract-based retrieval and thus lack support for our scenario. Results on SciFullBench demonstrate that PRISM outperforms existing abstract-level retrieval approaches and those specific to paper retrieval domains substantially, and is compatible with any (domain-agnostic) embedding models.

2 Related Work

Scientific Paper Retrieval Pioneering work in scientific paper-to-paper retrieval used the numerical statistics with citations or cocitations (Small, 1973; Haruna et al., 2018). Recently, thanks to the capability of neural models, many studies have focused on calculating semantic textual similarities between abstracts of respective documents (Bhagavatula et al., 2018; Ostendorff, 2020), with embedding models specific to this domain. For example, Cohan et al. (2020) and Ostendorff et al. (2022) fine-tune BERT-based models (Devlin et al., 2019) using abstract pairs extracted from the citation graph, and Mysore et al. (2022) further consider the Wasserstein distance between sentence segments within abstract pairs. Moreover, recent studies target multiple tasks (such as paper classification and citation prediction in addition to paper retrieval) in a unified framework by generating their respective representations adaptively (Singh et al., 2023; Zhang et al., 2023a). In contrast, we focus on paper-to-paper retrieval using the full content of papers, representing each paper in multiple ways.

Query Optimization with LLMs Since effective query formulation plays a central role in retrieval performance, a long line of research has explored query optimization, from early relevance feedback techniques (Rocchio Jr, 1971; Salton and Buckley, 1990) to query expansion approaches (Kuzi et al., 2016; Nogueira et al., 2019). More recent methods either leverage LLMs themselves to reformulate queries (Yu et al., 2023; Wang et al., 2023; Gao et al., 2023), or further augment them with external query-relevant information via retrieval (Yu et al.; Shen et al., 2024; Park and Lee, 2024; Lei et al., 2024). There are also studies to disambiguate queries (to capture their underlying semantics) by decomposing them into smaller subqueries (Zheng et al., 2024; Korikov et al., 2024). However, query optimization has less explored in paper retrieval, where each query paper is far longer than others.

3 Method

Paper-to-Paper Retrieval Given a query paper P (with its abstract as $P_{\texttt{abstract}}$), paper retrieval is to return a ranked list of relevant candidate papers from the corpus C. In contrast to existing studies that use $P_{\texttt{abstract}}$, we utilize its complete version P for query formulation and corpus construction.

Aspect-Aware Query Optimization To capture the multifaceted nature of scientific papers, we formulate the query optimization process as transforming the full paper P into a set of aspect-specific queries. Formally, we define a set of query optimization functions as follows: $\mathcal{F} = \{f_{\texttt{R}}, f_{\texttt{M}}, f_{\texttt{E}}\}$, where each function $f_i \in \mathcal{F}$ maps P to a query $q_i = f_i(P)$ that targets a specific aspect of the paper (e.g., research motivation, methods, and experiments). Notably, each function is instantiated with an LLM agent coupled with an aspect-specific template (see Appendix E). Additionally, we include the abstract $P_{\texttt{abstract}}$, since it reflects a general and broad view of the overall content (that can complement fine-grained queries), yielding the final query set: $\mathcal{Q} = \{f_i(P) \mid f_i \in \mathcal{F}\} \cup \{P_{\texttt{abstract}}\}$.

Retrieval with Multi-View Corpora Once the optimized query set $\mathcal{Q}$ is formulated, we perform retrieval individually for each query $q \in \mathcal{Q}$. Specifically, for the abstract-based query $P_{\texttt{abstract}}$, we use

Table 1: Main Results on SCIFULLBENCH, showing macro-averaged means over three different runs.

	IR Method	ICLR-NeurIPS References Recall@200	ICLR-NeurIPS References Recall@300	ICLR-NeurIPS Citations MRR@50	ICLR-NeurIPS Citations Recall@200	ACL-EMNLP References Recall@100	ACL-EMNLP References Recall@200	ACL-EMNLP Citations MRR@50	ACL-EMNLP Citations Recall@50
	Domain-Specific Retriever								
	SciNCL-**A2A**	45.96	51.80	37.82	42.22	28.21	35.58	29.77	19.04
	SPECTER2-Base-**A2A**	44.79	50.41	41.19	43.59	26.88	34.47	32.07	19.72
	SPECTER2-Adapter-MTL CTRL-**A2A**	45.16	51.07	39.18	42.07	27.35	34.44	30.51	18.66
	SciMult-MHAExpert-**A2A**	39.56	45.04	35.71	35.89	24.67	31.70	28.34	16.59
Domain-Agnostic Retriever	**Jina-Embeddings-v2-BASE-EN**								
	A2A (Abstract-to-Abstract)	44.83	49.92	40.29	42.53	26.92	33.57	32.79	19.27
	F2F (Full-to-Full)	45.22	50.60	42.38	44.30	29.15	36.72	35.60	21.56
	A2C (Abstract-to-Chunk)	40.58	45.89	38.77	42.44	25.22	32.44	32.28	20.50
	F2C (Full-to-Chunk)	39.75	45.17	34.87	40.63	25.64	33.11	28.59	19.53
	PRISM with Llama-3.2-3B-Instruct	47.75	53.41	42.79	48.74	29.81	38.55	39.36	23.98
	PRISM with GPT-4o-Mini-2024-0718	**48.18**	**53.96**	**43.49**	**49.93**	**30.61**	**39.07**	**39.86**	**24.70**
	Text-Embedding-3-Small								
	A2A (Abstract-to-Abstract)	45.64	51.22	40.02	43.81	27.78	34.50	33.75	20.73
	F2F (Full-to-Full)	34.94	39.82	39.74	37.34	18.70	24.53	22.19	12.32
	A2C (Abstract-to-Chunk)	39.56	44.55	39.68	41.31	23.34	30.55	31.65	18.88
	F2C (Full-to-Chunk)	35.34	39.94	34.77	37.89	18.98	24.20	17.91	12.12
	PRISM with Llama-3.2-3B-Instruct	47.31	**53.46**	45.16	48.23	**28.82**	**37.29**	37.48	22.97
	PRISM with GPT-4o-Mini-2024-0718	**47.39**	53.33	**46.44**	**49.63**	28.49	37.03	**39.94**	**23.74**

a corpus $\mathcal{C}_{\texttt{abstract}}$ containing candidate abstracts (as they are comparable in length and structure). For all other aspect-specific queries derived from the full paper, we use a segmented version of the full corpus: $\mathcal{C}_{\texttt{chunked}}$, where each paper is split into fixed-length chunks (likely to capture its specific aspect) and indexed using multi-vector representations (Khattab and Zaharia, 2020; Santhanam et al., 2021). Then, each query retrieves top-k relevant segments, which are then mapped back to their source papers, resulting in a ranked list of candidate papers per query, as follows: $\mathcal{R} = \{\mathcal{R}_q \mid q \in \mathcal{Q}\}$.

Rank Fusion We now turn to aggregate these aspect-specific rankings $\mathcal{R}$ into a unified document-level ranking, to ensure that relevance signals from different scientific dimensions are collectively reflected in the final retrieval outcome. In particular, we adopt Reciprocal Rank Fusion (RRF) (Cormack et al., 2009), which combines multiple ranked lists by assigning higher weights to top-ranked items in each list (without requiring score normalization across inconsistent fine-grained similarities), as follows: $\texttt{RRF}(P) = \sum_{q \in \mathcal{Q}} \frac{1}{k + \texttt{rank}_q(P)}$, where $\texttt{rank}_q(P)$ denotes the rank of candidate paper P in the retrieved list $\mathcal{R}_q$, and k is a smoothing constant. In other words, the ranking score from RRF ensures that candidates strongly aligned with at least one aspect can surface in the final retrieval outcome.

4 Experiment

4.1 SciFullBench

Query Formulation To support the task of full paper-to-paper retrieval, we collect papers in ML (NeurIPS and ICLR) and NLP (ACL and EMNLP) venues from OpenReview API[1], ACL-Anthology[2], and SEA (Yu et al., 2024) (see Appendix A for details), since no single source covers all of them. Then, for each query paper within them, we annotate its ground-truth relevant papers based on their neighboring relationships over the academic graph, such as **references** (incoming links) and **citations** (outgoing links). In addition, we parse each document using the AllenAI Science Parse tool[3], and further apply post-processing to remove reference sections, filter citation markers, and eliminate their associated in-context mentions (see Appendix A). Lastly, from the set of papers with at least 10 relevant documents on both the reference and citation criteria, we construct our evaluation suite by randomly sampling 400 query papers per venue.

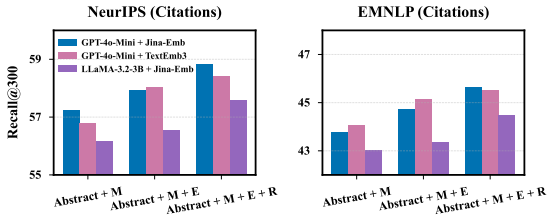

Figure 2: Retrieval performance with varying the number of coordinating query optimization agents in PRISM.

Corpus Construction To construct a comprehensive corpus for retrieval, we collect papers in the category around CS from 2020 to 2025 in arXiv[4], which offers open access to the full content. Also, to avoid exposing citation or reference information, we apply the same parsing and filtering procedures used in query construction. Lastly, we include labeled documents for all query papers in the corpus, resulting in a collection of about 40K papers with both abstract and full-text available. For segmentation of each document into finer units, we use the tokenization from NLTK (Loper and Bird, 2002).

[1] https://openreview.net/
[2] https://aclanthology.org/
[3] https://github.com/allenai/science-parse
[4] https://arxiv.org/

Table 2: Performance between specialized agents generating one query each and a single agent generating all queries.

	QoA + Retriever	ACL-Citations Recall@100	ICLR-Citations Recall@200
Single	GPT-4o-Mini-2024-0718 + JE-v2-Base-EN	33.27	45.62
	GPT-4o-Mini-2024-0718 + TE3-Small	32.89	45.35
	Llama-3.2-3B-Instruct + JE-v2-Base-EN	31.18	43.05
	Llama-3.2-3B-Instruct + TE3-Small	30.77	42.43
Multi	GPT-4o-Mini-2024-0718 + JE-v2-Base-EN	**34.25**	**46.44**
	GPT-4o-Mini-2024-0718 + TE3-Small	**33.48**	**46.34**
	Llama-3.2-3B-Instruct + JE-v2-Base-EN	**33.00**	**45.71**
	Llama-3.2-3B-Instruct + TE3-Small	**32.88**	**45.06**

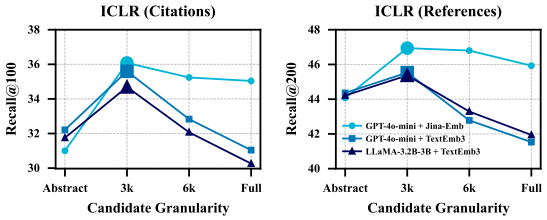

Figure 3: Results with different candidate paper granularities.

4.2 Evaluation Setup

Retrieval Models We compare PRISM against existing retrievers developed for scientific paper retrieval, as follows: **SPECTER2 Base** (Singh et al., 2023); **SciNCL** (Ostendorff et al., 2022); **SciMult-MHAExpert** (Zhang et al., 2023a); **SPECTER2 Adapters + MTL CTRL** (Singh et al., 2023). We also consider off-the-shelf general-purpose embedding models, such as **Jina-Embeddings-V2-Base-EN** (Günther et al., 2023) and **Text-Embedding-3-Small** (OpenAI, 2024c) for baselines and PRISM.

Retrieval Units We consider various retrieval units, where we denote **A** for *Abstract*, **F** for *Full paper*[5], and **C** for *chunked context* (i.e., segmented units of the full paper, each capped at 3K tokens). Using them, we consider four retrieval setups: **A2A** (Abstract-to-Abstract), **F2F** (Full-to-Full), **A2C** (Abstract-to-Chunk), and **F2C** (Full-to-Chunk).

Query Optimizers We instantiate query optimizers using LLMs: the open-weight Llama-3.2 (Meta, 2024) and the proprietary GPT-4o-Mini (OpenAI, 2024a). Please see Appendix B for more details.

4.3 Results and Analysis

Main Results Table 1 shows the main results, where PRISM outperforms all baselines across various configurations. First of all, despite using an off-the-shelf retriever (such as Jina-Embeddings) that underperforms the best task-specific retriever in the conventional A2A setup, PRISM achieves a significant average improvement of 7%. This confirms that optimizing queries via aspect-aware decomposition can yield substantial gains even without additional training or modification of the embedding space. Further, when using the same domain-agonistic retriever, PRISM consistently surpasses the corresponding baselines in the F2C and A2C setups in addition to the A2A setup. These results suggest that simply using longer inputs or chunked candidates is insufficient; yet, structural optimization of queries is essential for effective matching. Finally, PRISM exceeds the performance of the F2F setting by as much as 17%, highlighting the importance of using fine-grained (or aspect-specific) representations on both the queries and candidates.

Aspect Coverage in Query Optimization To assess the impact of covering diverse aspects in query formulation, we study the relationship between the number of participating query optimizer agents and the resulting retrieval performance. As Figure 2 shows, performance improves consistently as more aspect-specific agents contribute to the final query set, indicating the importance of capturing the multifaceted nature of scientific papers for retrieval.

Benefit of Specializing Agents by Aspect Our results in Table 2 confirm the effectiveness of using multiple specialized LLM agents, each dedicated to a single aspect. Specifically, using multiple specialized agents outperforms a single agent tasked with generating multiple queries (without aspect separation), yielding an average gain of 1.66%.

Impact of Candidate Granularity To see how the granularity of candidate representations affects the performance, we perform an analysis. The results in Figure 3 demonstrate the effectiveness of our multi-vector approach: segmenting each candidate document into 3K-token chunks yields the best performance, which outperforms not only abstract-only and full-document single-vector representations but also coarser chunking strategies (such as 6K-token segments), suggesting that finer-grained representations are effective in capturing diverse and localized signals within full-length papers.

5 Conclusion

In this work, we introduced PRISM, a novel scientific paper-to-paper retrieval framework that is composed of aspect-aware query optimization and fine-grained candidate representations. On a battery of tests with our newly constructed benchmark (SciFullBench) designed to support full-context retrieval, PRISM – by decomposing full papers into multiple aspect-specific queries and retrieving over (segmented) candidate corpora – consistently outperforms prior abstract-level and full-context retrieval baselines with off-the-shelf models, showing potential to move beyond surface-level retrieval.

[5]Since full papers often exceed the context length of embedding models, we truncate them up to the maximum length.

Limitations

Although our work explores a novel retrieval approach that leverages the entire context of scientific papers through multi-aspect query optimization, we have yet to explore further specialization or coordination of independent query optimizer agents in our system with auxiliary training procedures. This possibly bounds the performance of our framework as naively deploying off-the shelf individual LLMs may not operate in an optimal manner. Meanwhile, training each agent is also challenging since curating agent-specific labels for supervised finetuning is infeasible (at least within the scope of our work). Therefore, annotating or automatically collecting (labeled) data to enhance agent coordination thus remains an important avenue for future research. In addition, our candidate corpus is constructed by segmenting full documents into multiple vectors, each representing a portion of the original content. Although this approach offers scalability and efficiency for handling long-context candidates in large corpora, it may risk losing important contextual information during the rule-based segmentation process. In this vein, future work could explore (very rapid) content-aware segmentation strategies.

Ethics Statement

Although our PRISM framework improves retrieval performance compared to prior approaches, it still retrieves irrelevant papers at a high rate, potentially conveying incorrect information (such as harmful content) to both the human users and AI agents. Thus, in order to implement a trustworthy automated system, future research may focus on implementing verifiers that accurately filter out irrelevant documents from a pool of retrieved documents.

References

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

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'arcy, and 1 others. 2024. Openscholar: Synthesizing scientific literature with retrieval-augmented lms. *arXiv preprint arXiv:2411.14199*.

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. 2024. Researchagent: Iterative research idea generation over scientific literature with large language models. *arXiv preprint arXiv:2404.07738*.

Chandra Bhagavatula, Sergey Feldman, Russell Power, and Waleed Ammar. 2018. Content-based citation recommendation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 238–251, New Orleans, Louisiana. Association for Computational Linguistics.

Peter Brown and Yaoqi Zhou. 2019. Large expert-curated database for benchmarking document similarity detection in biomedical literature search. *Database*, 2019:baz085.

Eric Chamoun, Michael Schlichtkrull, and Andreas Vlachos. 2024. Automated focused feedback generation for scientific writing assistance. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9742–9763, Bangkok, Thailand. Association for Computational Linguistics.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 179 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv*, abs/2501.12948.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman,

Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 510 others. 2024. The llama 3 herd of models. *ArXiv*, abs/2407.21783.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Ronald L. Graham. 1972. An efficient algorithm for determining the convex hull of a finite planar set. *Info. Proc. Lett.*, 1:132–133.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, and 1 others. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *arXiv preprint arXiv:2310.19923*.

Khalid Haruna, Maizatul Akmar Ismail, Abdullahi Baffa Bichi, Victor Chang, Sutrisna Wibawa, and Tutut Herawan. 2018. A citation-based recommender system for scholarly paper recommendation. In *Computational Science and Its Applications–ICCSA 2018: 18th International Conference, Melbourne, VIC, Australia, July 2-5, 2018, Proceedings, Part I 18*, pages 514–525. Springer.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Yiqiao Jin, Qinlin Zhao, Yiyang Wang, Hao Chen, Kaijie Zhu, Yijia Xiao, and Jindong Wang. 2024. Agentreview: Exploring peer review dynamics with llm agents. *arXiv preprint arXiv:2406.12708*.

Anshul Kanakia, Zhihong Shen, Darrin Eide, and Kuansan Wang. 2019. A scalable hybrid research paper recommender system for microsoft academic. In *The world wide web conference*, pages 2893–2899.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Anton Korikov, George Saad, Ethan Baron, Mustafa Khan, Manav Shah, and Scott Sanner. 2024. Multi-aspect reviewed-item retrieval via llm query decomposition and aspect fusion. In *IR-RAG@SIGIR*, pages 23–33.

Saar Kuzi, Anna Shtok, and Oren Kurland. 2016. Query expansion using word embeddings. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 1929–1932.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Yibin Lei, Yu Cao, Tianyi Zhou, Tao Shen, and Andrew Yates. 2024. Corpus-steered query expansion with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 393–401, St. Julian's, Malta. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

Zoran Medić and Jan Snajder. 2022. Large-scale evaluation of transformer-based article encoders on the task of citation recommendation. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 19–31, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Meta. 2024. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models.

Sheshera Mysore, Arman Cohan, and Tom Hope. 2022. Multi-vector models with textual guidance for fine-grained scientific document similarity. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4453–4470, Seattle, United States. Association for Computational Linguistics.

Sheshera Mysore, Tim O'Gorman, Andrew McCallum, and Hamed Zamani. 2021. CSFCube - a test collection of computer science research articles for faceted query by example. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.

OpenAI. 2022. text-embedding-ada-002. https://platform.openai.com/docs/models/text-embedding-ada-002.

OpenAI. 2024a. Gpt-4o system card. https://cdn.openai.com/gpt-4o-system-card.pdf.

OpenAI. 2024b. Gpt-4o system card. https://cdn.openai.com/gpt-4o-system-card.pdf.

OpenAI. 2024c. text-embedding-3-small. https://platform.openai.com/docs/models/text-embedding-3-small.

OpenAI. 2025a. o3-and-o4-mini-system-card. https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf.

OpenAI. 2025b. o3-mini-system-card-feb10. https://cdn.openai.com/o3-mini-system-card-feb10.pdf.

Malte Ostendorff. 2020. Contextual document similarity for content-based literature recommender systems. *arXiv preprint arXiv:2008.00202*.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood contrastive learning for scientific document representations with citation embeddings. *arXiv preprint arXiv:2202.06671*.

Jeonghyun Park and Hwanhee Lee. 2024. Conversational query reformulation with the guidance of retrieved documents. *arXiv preprint arXiv:2407.12363*.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, and 1 others. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Joseph John Rocchio Jr. 1971. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*.

Gerard Salton and Chris Buckley. 1990. Improving retrieval performance by relevance feedback. *Journal of the American society for information science*, 41(4):288–297.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2021. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *arXiv preprint arXiv:2112.01488*.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Yibin Lei, Tianyi Zhou, Michael Blumenstein, and Daxin Jiang. 2024. Retrieval-augmented retrieval: Large language models are strong zero-shot retriever. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15933–15946, Bangkok, Thailand. Association for Computational Linguistics.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2023. SciRepEval: A multi-format benchmark for scientific document representations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5548–5566, Singapore. Association for Computational Linguistics.

Henry Small. 1973. Co-citation in the scientific literature: a new measure of the relationship between documents. *J. Am. Soc. Inf. Sci.*, 42:676–684.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore. Association for Computational Linguistics.

Siwei Wu, Yizhi Li, Kang Zhu, Ge Zhang, Yiming Liang, Kaijing Ma, Chenghao Xiao, Haoran Zhang, Bohao Yang, Wenhu Chen, Wenhao Huang, Noura Al Moubayed, Jie Fu, and Chenghua Lin. 2024. SciMMIR: Benchmarking scientific multi-modal information retrieval. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12560–12574, Bangkok, Thailand. Association for Computational Linguistics.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland. Association for Computational Linguistics.

Jianxiang Yu, Zichen Ding, Jiaqi Tan, Kangyang Luo, Zhenmin Weng, Chenghua Gong, Long Zeng, RenJing Cui, Chengcheng Han, Qiushi Sun, Zhiyong Wu, Yunshi Lan, and Xiang Li. 2024. Automated peer reviewing in paper SEA: Standardization, evaluation, and analysis. In *Findings of the Association*

for Computational Linguistics: EMNLP 2024, pages 10164–10184, Miami, Florida, USA. Association for Computational Linguistics.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations*.

Wenhao Yu, Zhihan Zhang, Zhenwen Liang, Meng Jiang, and Ashish Sabharwal. Improving language models via plug-and-play retrieval feedback. corr, abs/2305.14002, 2023. doi: 10.48550. *arXiv preprint ARXIV.2305.14002*.

Yu Zhang, Hao Cheng, Zhihong Shen, Xiaodong Liu, Ye-Yi Wang, and Jianfeng Gao. 2023a. Pre-training multi-task contrastive learning models for scientific literature understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12259–12275, Singapore. Association for Computational Linguistics.

Yu Zhang, Bowen Jin, Xiusi Chen, Yanzhen Shen, Yunyi Zhang, Yu Meng, and Jiawei Han. 2023b. Weakly supervised multi-label classification of full-text scientific papers. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3458–3469.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. 2024. Take a step back: Evoking reasoning via abstraction in large language models. In *The Twelfth International Conference on Learning Representations*.

A SciFullBench

A.1 Comparison with Prior Benchmarks

Previous benchmarks for scientific literature search where the full context is available for both query and candidates rarely exist. (Kanakia et al., 2019) introduce expert-annotated paper recommendation benchmark using abstract and citations within Microsoft Academic Graph(MAG). SciDocs (Cohan et al., 2020) and SciRepEval (Singh et al., 2023) reveal an evaluation set to search relevant scientific literature within a pool of 30 candidates per query document with 5 gold labels in the computer science domain, while MDCR (Medić and Snajder, 2022) disclose a benchmark with 60 candidates per query from 19 scientific fields sourced from MAG. RELISH (Brown and Zhou, 2019) also provides expert-annotated gold candidates that are relevant to the respective input documents in the biomedical domain. In addition, (Mysore et al., 2021) formulates a CFSCube benchmark to evaluate fine-grained sentence-wise alignment between abstract passages. SciMMIR (Wu et al., 2024) also presents a multimodal document retrieval evaluation set to evaluate the performance of figure-wise document retrieval frameworks. However, such benchmarks do not include candidates or queries in which their entire lexical content is fully disclosed. Although (Zhang et al., 2023b) intends to evaluate their classifier pipeline using the complete scientific context, it contains five potential candidates per query, which is infeasible to evaluate large-scale literature retrieval frameworks. Conversely, our proposed benchmark consists of fully-disclosed document context, along with massive number of target candidates, adhering to the realistic setting for evaluating scientific literature retrieval.

A.2 Construction Procedure

In this section, we provide more details on the step-by-step construction process of SciFullBench.

Step 1 As mentioned in the main content, we initially crawled research papers from top-tier machine learning venues including ICLR 2024, 2025, NeurIPS 2024, 2023, ACL 2024, 2023, and EMNLP 2024, 2023 for our query documents. We scraped existing publications from the main conference for ACL and EMNLP, including both long- and short- papers from the ACL-Anthology website[6], and used the OpenReview[7] API to obtain submitted research papers for ICLR and NeurIPS. Moreover, for ICLR 2024 and NeurIPS 2023, we included a subset of documents uploaded by authors from SEA (Yu et al., 2024). Afterwards, we obtained metadata from the arxiv database uploaded to the Kaggle[8] website, with available data up to January 2025. Moreover, we constructed our initial temporary raw corpus where its id starts with 20 to 25 which refers to its uploaded date on Arxiv database, limiting our target corpus to fairly recent papers. Also, we only filtered papers that are in the machine learning domain, to those belong in cs.AI, cs.LG, cs.CL, cs.CV, cs.NE, cs.IR, cs.DS, cs.CC, cs.DL, cs.HC, cs.RO, cs.MM, cs.CG, cs.SY since our queries are sampled from ML venues.

Step 2 After devising raw data for both queries and candidates, we subsequently formulate gold candidates using citations to represent contextual proximity between scientific documents. We collect the title of papers that cite our potential query documents using the Semantic Scholar API[9] and check whether there exists a paper with a matched title (case-insensitive) in our arxiv metadata, and exclude any potential query documents with fewer than ten gold candidates that meet such conditions. As for benchmark split with references, we use Allen AI Science Parse to obtain reference information for respective documents, and follow the same process in filtering out documents based on the implemented criteria from above. Using such data, we formulate our raw (query document, gold candidate) pairs where its abstract information is intact and where more than 10 deduplicated candidates exist, organized by splits in respective years belonging to venues ICLR, NeurIPS, ACL and EMNLP. For each split, we aggregate all the filtered (query, candidate) set by year within each venue. For example, we merge all the candidates in the citation split for ICLR 2024 and ICLR 2025 into a single set of ICLR-cited. From this filtered pool, we randomly sampled 500 potential query candidates and formulated a corpus containing gold candidates for each query. This process enables our target corpus to be kept within manageable size. Next, we assembled the original pdfs of the papers in our temporarily formulated target corpus. Since Arxiv APIs do not

[6] https://aclanthology.org/
[7] https://openreview.net/
[8] https://www.kaggle.com/
[9] https://www.semanticscholar.org

Table 3: Query Paper statistics in SCIFULLBENCH benchmark.

	ICLR		**NeurIPS**		**EMNLP**		**ACL**	
	References	Citations	References	Citations	References	Citations	References	Citations
domain	ML		ML		ML/CL		ML/CL	
years included	2024,2025		2023,2024		2023,2024		2023,2024	
# of papers per year	(141,259)	(324,76)	(170, 230)	(353,47)	(226, 174)	(334, 66)	(184,216)	(256,144)
# of tokens per paper	9402.35	8183.37	11184.39	9035.43	5908.36	6226.45	6190.06	6614.6
average # of candidates per sample	21.74	42.25	24.58	40.37	19.64	39.24	18.05	33.89
minimum # of candidates per sample	10	10	10	10	10	10	10	10

Table 4: Corpus statistics for SCIFULLBENCH.

	SCIFULLBENCH
Total # of papers	40,782
Avg. # of tokens per abstract	200.39
Avg. # of tokens per paper	6987.67
Total # of segmented corpus	115004
Avg. # of tokens per segment	2479.90

support large-scale requests, we used the Google Cloud API[10] to access the full paper dump directly and collect the latest updated versions for each paper by matching its unique ids within the Arxiv database.

Step 3 Subsequently, using the AllenAI Science Parse tool, we parsed the query and candidate pdfs collected in step 2 into a json file containing title, abstract, main content list containing dictionaries with header and contents as keys, and list of reference paper information along with its mentions for both queries and candidates. Since we use citation information as the main signal for measuring document similarities, excluding the citations and the entire reference section was critical to validate the fairness of our benchmark. Although Allen AI Science Parse generally separates reference from main content, there were several cases where reference was intermixed within the main content. Thus, we applied an auxiliary filtering algorithm. Since our tool parses pdf into a structured json file, we were able to obtain text information segmented into multiple passages. Hence, prior to reformatting it back to complete text, we eliminated sections or headers that contain at least one reference title (case-insensitive) completely while reformatting into a full paper. However, there were still issues where for some cases the reference section was left vacant. Because our filtering algorithm depends on the set of titles within parsed reference section, it cannot correctly exclude cases intermixed with reference information when such data are inaccessible. Hence, we ruled out any query documents or papers within our target corpus that did not include more than four references in the respective parsed documents. In this way, it resolves problematic circumstances in which we pass on perturbed documents with reference knowledge by considering that it has no issues because the reference title did not exist in the first place. Moreover, due to our filtering heuristics, there remains an issue where numerous documents experience severe loss of their original content, since we excluded any passage or header that had at least 1 reference title. To mitigate such concerns, we did not include any documents in which such problematic passages comprise more than half of the total main-content list. Furthermore, we ensured to remove any residing title(case-insensitive) included in the reference section.

Step 4 Based on the preprocessed full document, we formulate a pseudo-definitive benchmark consisting of (query, gold candidate), target corpus set by random sampling 400 query documents per split in each venue that has 10 gold candidates still existing in the above filtered target corpus. Ultimately, we formulate a total of 3200 query documents, along with large-scale target corpora consisting of all the gold candidates of such queries. Finally, we remove citations and interlinked mentions from the entire set of formatted papers and finalize our benchmark. Since this process does not lead to exclusion of query document from our benchmark, we have called the previous stage a pseudo-definitive benchmark. Citation mentions are removed using the information provided from our parsing tool, and citation patterns are removed using more than 10 different patterns in the Python regular expression. Consequently, we devise a definitive benchmark in which all documents for both query and candidate corpus consist of title, abstracts, full-paper text, and list of segmented content for the corresponding full-paper text.

[10]https://cloud.google.com/apis?hl=ko

B Experiment Details

B.1 Models

We compared our method with retrieval using domain-specific embedding models, as mentioned in the main section, to demonstrate robust improvement over approaches that seek to devise optimal representations of paper abstracts through extensive training. For fair assessment, we mainly compare our method with models that demonstrated SoTA performance in previous citation link prediction and paper retrieval benchmarks. SPECTER2 models with adapters and its multitask control codes have been reported to have achieved SoTA performance on the MDCR benchmark (Medić and Snajder, 2022). We used proximity control code adapter, as it was specialized in acquiring and ranking relevant papers, and also base models uploaded on huggingface [11]. In addition, we experiment with SPECTER2-base models, trained with triplet loss by inducing neural retrievers to favor positive abstract pairs over vice versa given abstract of query documents sampled from discrete citation graphs. SciNCL [12] is also trained in a similar way, where its abstract pairs are derived from the continuous citation embedding space, achieving the strongest performance on the SciDocs benchmark to date. In addition, we compare our method with the SciMult-MHAExpert model implemented with Mixture-of-Experts (Fedus et al., 2022) architecture which compartmentalize internal transformer layers for different tasks in scientific literature tasks, and is reported to outperform previous models on the recommendation benchmark (Kanakia et al., 2019). For SciMult-MHAExpert, we experimented with the model released on the SciMult github repository [13], utilizing an expert model specially trained for link prediction tasks that predicts linked documents given the abstract of the source document.

As for our domain-agnostic embedding models, we particularly chose off-the shelf retrievers with long context window. This was to ensure that we provided fair experimental setting for our baselines, most notably full document-to-full document retrieval setting. Since general full-documents exceed the context window of most embedding models, we specifically chose long-context window embedding models to mitigate unfair penalization on our full document-to-document retrieval baseline.

[11] https://huggingface.co/allenai/specter2
[12] https://huggingface.co/malteos/scincl
[13] https://github.com/yuzhimanhua/SciMult

B.2 Metrics

In our experiments, **Recall@K** is used as our main metric, which measures the ratio of correct candidates within the Top@k retrieved results. This aligns with our objective, where we seek to acquire a more relevant pool of papers using diverse aspect-aware queries. Moreover, we report **Mean Reciprocal Rank(MRR@K)** to further validate the capabilities of our framework.

B.3 Inference Details

We primarily use the Euclidean distance to measure similarities between our adaptively generated queries and candidates, where we acquired candidates with the minimum L2 distance. Since minimizing the L2 distance was objective for most of our baseline domain-specific embedding models such as SciNCL, SPECTER2-Base, SPECTER2-Adapter-MTL CTRL, we matched such settings when we used jina-embeddings-v2-base-en and text-embedding-3-small. However, for SciMult, we measured MIPs (Maximum Inner Product), since it was trained to maximize MIPs likewise in DPR. In addition, we utilize the FAISS (Douze et al., 2024) library, which enables efficient retrieval from large-scale corpora. We implemented an efficient L2 search using FlatL2 and FlatIP for MIPs.

As for the setup when Llama-3.2 models are used as query optimizers, we report the results by fixing the temperature to 0 and fixing the maximum generation token hyperparameter to 2000 and setting the repetition penalty to 1.2. In addition, we used the VLLM (Kwon et al., 2023) library for faster inference of open source models, using single A6000 GPU for inference of open-sourced models. When it comes to experiments using GPT-4o-Mini-2024-0718 and GPT-4o-2024-11-20 as query optimizers, we used temperature 0 and default hyper parameters of the OpenAI client.chat.completions API [14], while completely using the default hyperparameter settings for O3-Mini-2025-0131 and O4-Mini-2025-04-16(temperature is also set to default). Moreover, 60 is used as the hyperparameter k for Reciprocal Rank Fusion. For base agent experiments in Table 2, we equalized the hyper parameters to those used in its respective counterpart comparison groups, and are prompted to generate three different queries in a structured manner, using Python Pydantic BaseModel [15] module.

[14] https://platform.openai.com/docs/guides/
[15] https://docs.pydantic.dev/latest/api

Table 5: Additional report on performance of our PRISM incorporating additional LLM backbone models and neural retrievers.

Model + Retriever	ICLR-References Recall@200	ICLR-Citations Recall@200
Baseline		
SciNCL-A2A	44.55	38.40
SPECTER2-Base-A2A	43.43	39.27
Jina-Embeddings-v2-BASE-EN-A2A	43.35	39.19
Text-Embedding-3-Small-A2A	43.48	40.56
Text-Embedding-Ada-002-A2A	37.81	33.69
Ours		
Llama-3.2-1B-Instruct + JE-v2-Base-EN	44.35	44.51
Llama-3.2-1B-Instruct + TE3-Small	43.63	44.12
GPT-4o-Mini-2024-0718 + TE-Ada-002	40.99	41.67
GPT-4o-2024-11-20 + JE-v2-Base-EN	**47.96**	**47.50**
O3-Mini-2025-01-31 + JE-v2-Base-EN	**49.08**	**46.91**
O3-Mini-2025-01-31 + TE3-Small	**48.51**	**47.26**
O4-Mini-2025-04-16 + TE3-Small	**48.76**	**47.31**

C Supplementary Experiments

In this section, we provide additional experiments and analysis of our work. In Table 10, the original results before taking the macro-average of the result pairs sampled from the venue pairs, ICLR-NeurIPS and ACL-EMNLP. In Table 5, we report additional results using State-of-the-ART LLMs as query optimizers, namely O4-Mini-2025-04-16, (OpenAI, 2025a), O3-Mini-2025-01-31 ,(OpenAI, 2025b), and GPT 4o-2024-11-20 (OpenAI, 2024b). We also provide results using a smaller LLM agent compared to the ones provided in our main results, utilizing LLama-3.2-1B-Instruct (Meta, 2024). Likewise, the results from Table 1 and 10, our framework generally attains enhanced performance when compared to the best performing domain-specific retriever + abstract-to-abstract retrieval setup, while also outperforming the setup in which the same domain-agnostic retrievers are used for abstract-to-abstract retrieval even when using different LLMs as our query optimizers. Note that query optimizers based on LLMs with more robust instruction-following capabilities and stronger reasoning abilities typically excel vice versa. This further indicates the expandability of our approach, where it has the potential to be further improved with the integration of more advanced reasoning models. Moreover, in Table 5, we also experiment with Text-Embedding-Ada-002 (OpenAI, 2022) and demonstrate that our pipeline is capable of achieving robust performance improvement using arbitrary embedding models, again emphasizing the importance of structural optimization of input queries and candidates.

Moreover, in Table 7, the results are reported when abstract-to-abstract retrieval(A2A) is not utilized within PRISM. Despite the fact that the A2A retrieval method is excluded from our framework, robust improvement over baselines can be observed, although not to the extent when abstract-to-abstract retrieval is incorporated. This in turn highlights the effectiveness of our pipeline in generating multiple aspect-aware queries in the absence of complementary abstract to abstract retrieval.

Table 6: Analysis on the impact of incorporating rank fusion to attain unified results of our hybrid retrieval framework compared to using embedding aggregation mechanism in previous multi-vector retrieval approaches on ACL-Citations split.

Retrieval Method	MRR@50	Recall@100	Recall@200
Ours w/o RRF (PRISM w/o A2A)			
Naive Aggregation			
GPT-4o-Mini-2024-0718 + TE3-Small	22.04	3.95	4.55
Late Interaction(MaxSim)			
GPT-4o-Mini-2024-0718 + TE3-Small	36.12	22.46	28.83
Ours w/ RRF (PRISM w/o A2A)			
GPT-4o-Mini-2024-0718 + TE3-Small	41.35	32.49	41.54

In addition, we conduct an ablation study on the effect of RRF (Reciprocal Rank Fusion) in our hybrid retrieval system, compared to embedding-level merging approach in prior multi-vector retrieval approaches. We primarily compare with two traditional approaches in embedding-level merging strategy, the naive aggregation strategy that forcefully computes similarities (in our case L2 distance) between all the sub-vectors of respective query and candidates and naively sums it up to acquire similarity score of original query and documents. Meanwhile, we also present comparison with late interaction strategy, where only the maximum similarity for subquery and its corresponding sub-documents is aggregated to form original query, document similarity. Table 6 illustrates a drastic drop in retrieval performance when queries optimized in various aspects are used as subvector representations of original articles and are used to compute a single similarity value when matched with the segmented corpus of target candidates. This supports the validity of our design choice, where our ranking-merging system is more robust to noise, while allowing candidates that are strongly aligned in one aspect but unaligned otherwise to be retrieved.

Furthermore, we hypothesize that our approach which adaptively generates multiple queries predicated on various aspects of scientific paper would result in retrieval of more contextually diverse papers. We validate our claim through a deeper analysis, typically observing the level of embedding dispersion within the latent vector space, using three notable metrics: **cosine distance**, **centroid distance**, and **convex hull volume**. For cosine distance, we averaged the pairwise cosine similarities(represents semantic distance between vectors)

Table 7: Performance analysis of our pipeline when original abstract to abstract retrieval is not utilized. Likewise Table 5, we chose the best domain-specific retriever + A2A, and two domain-agnostic A2A results as our baseline. Best results for respective setups are highlighted in bold, while second best results are underlined.

Retrieval Method	EMNLP-References Recall@200	ACL-Citations MRR@50
Baseline		
SciNCL-A2A	34.55	31.83
SPECTER2-Base-A2A	33.12	33.77
Jina-Embeddings-v2-BASE-EN-A2A	32.64	34.97
Text-Embedding-3-Small-A2A	33.61	36.42
Ours w/ A2A		
GPT-4o-Mini-2024-0718 + JE-v2-Base-EN	**38.44**	**42.91**
GPT-4o-Mini-2024-0718 + TE3-Small	**36.34**	**42.26**
Llama-3.2-3B-Instruct + JE-v2-Base-EN	**37.90**	**41.32**
Llama-3.2-3B-Instruct + TE3-Small	**36.54**	38.78
Ours w/o A2A		
GPT-4o-Mini-2024-0718 + JE-v2-Base-EN	36.64	42.68
GPT-4o-Mini-2024-0718 + TE3-Small	34.05	41.35
Llama-3.2-3B-Instruct + JE-v2-Base-EN	35.27	40.14
Llama-3.2-3B-Instruct + TE3-Small	34.13	**39.88**

Table 8: Analysis on the diversity of retrieved document candidates within embedding space using three distinct metrics. Results were averaged my measuring diversity within top 200 retrieved candidates for respective queries in each benchmark split.

Retrieval Method	Cosine Distance	Centroid Distance	Convex Hull Volume
NeurIPS-Citations			
Jina-Embeddings-v2-BASE-EN-A2A	0.1924	5.67	101.23
Text-Embedding-3-Small-A2A	0.4198	0.6436	0.1299
Ours			
GPT-4o-Mini-2024-0718 + JE-v2-Base-EN	0.2057	5.87	120.50
GPT-4o-Mini-2024-0718 + TE3-Small	0.4456	0.6626	0.1478
Llama-3.2-3B-Instruct + JE-v2-Base-EN	0.2087	5.91	123.04
Llama-3.2-3B-Instruct + TE3-Small	0.4469	0.6636	0.1467
ACL-References			
Jina-Embeddings-v2-BASE-EN-A2A	0.1869	5.60	96.94
Text-Embedding-3-Small-A2A	0.4145	0.6395	0.1307
Ours			
GPT-4o-Mini-2024-0718 + JE-v2-Base-EN	0.2017	5.83	119.30
GPT-4o-Mini-2024-0718 + TE3-Small	0.4431	0.6607	0.1498
Llama-3.2-3B-Instruct + JE-v2-Base-EN	0.2046	5.87	121.86
Llama-3.2-3B-Instruct + TE3-Small	0.4445	0.6618	0.1476

of retrieved document embeddings and subtracted it from 1, essentially computing the inverse cosine similarity. In addition, we measured the centroid distance, a Euclidean distance from its centroid of k document embeddings, denoting the variance of data points in latent space. Furthermore, we provide Convex Hull (Graham, 1972) Volume as a metric to statistically measure the dispersion of the embedding data points, which computes the volume of the smallest possible bounding convex closure of the data points after projecting the embedding vectors to the three-dimensional vector space. The results recorded in Table 8 support our claim, as our framework typically retrieves documents with more diverse semantics, in contrast to the baseline domain-agnostic retriever + A2A setting. Moreover, we further visualize the distribution of the embedding vectors of Top@300 retrieved documents for four sampled queries on a two-dimensional plane using **t-distributed Stochastic Neighbor Embedding(t-SNE)** (Van der Maaten and Hinton, 2008) and its distribution boundaries via a two-dimensional convex hull area of the projected data points in Figure 4. We simultaneously visualize the distribution of t-SNE data points for both ground truth candidates, baseline retrieved results, and retrieved results from PRISM. The visualization also advocates our hypothesis, where ours typically displays a more dispersed distribution compared to retrieved document embeddings from baselines. In addition, the cases in Figure 4 demonstrate that through improved diversity within the retrieved pool of articles, PRISM is able to retrieve more relevant papers with ground-truth candidates, as coverage of the documents has increased compared to previous approaches.

D Case Study

Please refer to Table 11 and Table 12 for case studies on query and retrieval results.

E Prompts

Please refer to all the prompts that we use to elicit the query optimizations in Figures 5, 6, 7, and 8.

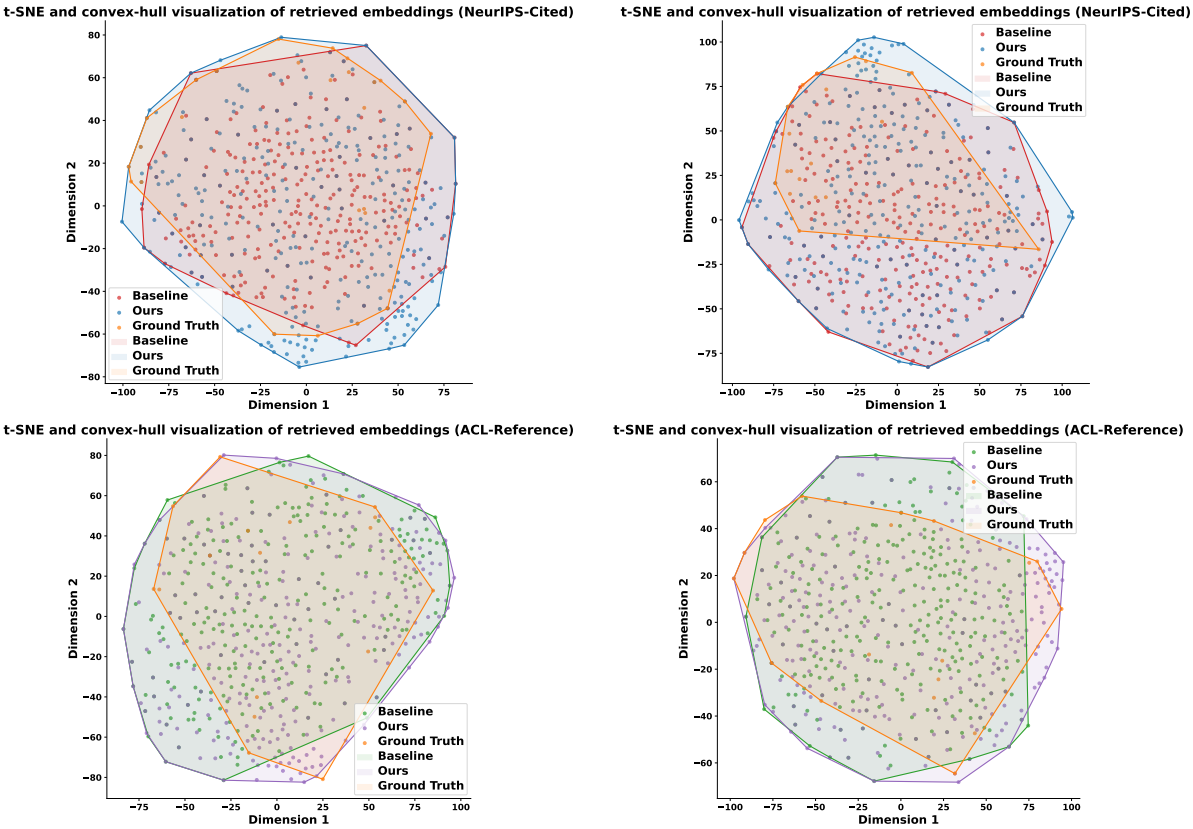

Figure 4: t-SNE and its convex hull boundary visualization of Top@300 retrieved document embeddings across four sampled queries from respective benchmark splits. We analyzed retrieved results when Jina-Embeddings-V2-Base-EN is used as neural embedding models. Baseline setup refers to generic abstract to abstract retrieval, while we use GPT-4o-Mini-2024-0718 as our query optimizers for our setting.

Table 9: Expanded version of Table 2 with additional experiments on EMNLP-Citations, provided with more metrics.

		ACL-Citations		ICLR-Citations		EMNLP-Citations	
	QoA + Retriever	Recall@100	Recall@200	Recall@100	Recall@200	Recall@100	Recall@200
Single	GPT-4o-Mini-0718 + JE-v2-Base-EN	33.27	41.94	35.82	45.62	31.43	40.02
	GPT-4o-Mini-0718 + TE3-Small	32.89	42.65	35.03	45.35	30.63	39.43
	Llama-3.2-3B-Instruct + JE-v2-Base-EN	31.18	39.69	32.98	43.05	28.34	37.04
	Llama-3.2-3B-Instruct + TE3-Small	30.77	40.32	32.33	42.43	28.73	37.37
Multi	GPT-4o-Mini-0718 + JE-v2-Base-EN	34.25	42.79	36.08	46.44	32.00	40.61
	GPT-4o-Mini-0718 + TE3-Small	33.48	43.31	35.60	46.34	30.86	40.69
	Llama-3.2-3B-Instruct + JE-v2-Base-EN	33.00	41.85	35.46	45.71	30.75	39.50
	Llama-3.2-3B-Instruct + TE3-Small	32.88	42.19	34.69	45.06	29.72	39.33

Table 10: Expanded version of our main results on SCIFULLBENCH. Note that results from Table 1 is an average between results from ICLR-NeurIPS and ACL-EMNLP. We provide mean and standard deviation over three iterations.

	ICLR				NeurIPS			
	References		Citations		References		Citations	
IR Method	Recall@200	Recall@300	MRR@50	Recall@200	Recall@200	Recall@300	MRR@50	Recall@200
BASELINE								
SciNCL-A2A	44.55± 0.00	50.51± 0.00	32.33 ± 0.00	38.40 ± 0.00	47.37 ± 0.00	53.08 ± 0.00	43.31 ± 0.00	46.04 ± 0.00
SPECTER2-Base-A2A	43.43 ± 0.00	48.99± 0.00	36.09 ± 0.00	39.27 ± 0.00	46.15 ± 0.00	51.82 ± 0.00	46.28 ± 0.00	47.90 ± 0.00
SPECTER2-Adapter-MTL CTRL-A2A	43.72 ± 0.00	49.98 ± 0.00	34.41 ± 0.00	38.09 ± 0.00	46.60 ± 0.00	52.16 ± 0.00	43.95 ± 0.00	46.04 ± 0.00
SciMult-MHAExpert-A2A	37.13 ± 0.00	42.56 ± 0.00	29.86 ± 0.00	31.48 ± 0.00	41.98 ± 0.00	47.52 ± 0.00	41.56 ± 0.00	40.30 ± 0.00
Jina-Embeddings-v2-BASE-EN-A2A	43.35 ± 0.00	48.38 ± 0.00	33.52 ± 0.00	39.19 ± 0.00	46.30 ± 0.00	51.45 ± 0.00	47.05 ± 0.00	45.87 ± 0.00
Text-Embedding-3-Small-A2A	43.48 ± 0.01	48.94 ± 0.00	35.27 ± 0.00	40.56 ± 0.00	47.79 ± 0.01	53.51 ± 0.00	44.76 ± 0.00	47.05 ± 0.00
Jina-Embeddings-v2-BASE-EN-F2F	44.29 ± 0.00	49.58 ± 0.00	37.38 ± 0.00	42.44 ± 0.00	46.15 ± 0.00	51.62 ± 0.00	47.37 ± 0.00	46.15 ± 0.00
Text-Embedding-3-Small-F2F	35.31 ± 0.01	40.07 ± 0.02	35.02 ± 0.00	34.83 ± 0.00	34.57 ± 0.01	39.56 ± 0.00	44.45 ± 0.00	39.85 ± 0.00
Jina-Embeddings-v2-BASE-EN-A2C	39.07 ± 0.00	44.57 ± 0.00	34.18 ± 0.00	39.31 ± 0.00	42.08 ± 0.00	47.20 ± 0.00	43.36 ± 0.00	45.56 ± 0.00
Text-Embedding-3-Small-A2C	37.41 ± 0.00	42.23 ± 0.01	36.17 ± 0.02	38.04 ± 0.00	41.71 ± 0.01	46.86 ± 0.01	43.18 ± 0.00	44.58 ± 0.02
Jina-Embeddings-v2-BASE-EN-F2C	38.90 ± 0.00	44.50 ± 0.00	29.49 ± 0.00	38.47 ± 0.00	40.60 ± 0.00	45.83 ± 0.00	40.25 ± 0.00	42.79 ± 0.00
Text-Embedding-3-Small-F2C	36.45 ± 0.00	41.43 ± 0.00	31.19 ± 0.00	35.98 ± 0.01	34.22 ± 0.02	38.45 ± 0.01	38.35 ± 0.00	39.80 ± 0.00
OURS (PRISM)								
Jina-Embeddings-v2-BASE-EN + Llama-3.2-3B-Instruct	46.48 ± 0.00	52.04 ± 0.00	39.97 ± 0.00	45.71 ± 0.00	49.03 ± 0.10	54.77 ± 0.22	45.61 ± 0.00	51.77 ± 0.00
Text-Embedding-3-Small + Llama-3.2-3B-Instruct	45.39 ± 0.03	51.70 ± 0.03	42.15 ± 0.00	45.04 ± 0.02	49.23 ± 0.00	55.22 ± 0.00	48.17 ± 0.00	51.42 ± 0.00
Jina-Embeddings-v2-BASE-EN + GPT-4o-Mini-0718	46.88 ± 0.10	52.53 ± 0.25	41.71 ± 0.43	46.45 ± 0.10	49.47 ± 0.05	55.39 ± 0.09	45.27 ± 0.39	53.41 ± 0.08
Text-Embedding-3-Small + GPT-4o-Mini-0718	45.37 ± 0.21	51.43 ± 0.28	43.73 ± 0.22	46.32 ± 0.02	49.40 ± 0.13	55.22 ± 0.07	49.14 ± 0.36	52.94 ± 0.03

	ACL				EMNLP			
	References		Citations		References		Citations	
IR Method	Recall@100	Recall@200	MRR@50	Recall@50	Recall@100	Recall@200	MRR@50	Recall@50
BASELINE								
SciNCL-A2A	29.18 ± 0.00	36.62 ± 0.00	31.83 ± 0.00	19.95 ± 0.00	27.23 ± 0.00	34.55 ± 0.00	27.71 ± 0.00	18.13 ± 0.00
SPECTER2-Base-A2A	28.28 ± 0.00	35.81 ± 0.00	33.77 ± 0.00	21.14 ± 0.00	25.47 ± 0.00	33.12 ± 0.00	30.37 ± 0.00	18.30 ± 0.00
SPECTER2-Adapter-MTL CTRL-A2A	28.42 ± 0.00	35.73 ± 0.00	32.47 ± 0.00	19.84 ± 0.00	26.28 ± 0.00	33.14 ± 0.00	28.54 ± 0.00	17.48 ± 0.00
SciMult-MHAExpert-A2A	25.96 ± 0.00	33.07 ± 0.00	30.30 ± 0.00	17.84 ± 0.00	23.37 ± 0.00	30.32 ± 0.00	26.37 ± 0.00	15.34 ± 0.00
Jina-Embeddings-v2-BASE-EN-A2A	27.75 ± 0.00	34.49 ± 0.00	34.97 ± 0.00	20.19 ± 0.00	26.08 ± 0.00	32.64 ± 0.00	30.61 ± 0.00	18.34 ± 0.00
Text-Embedding-3-Small-A2A	28.71 ± 0.01	35.40 ± 0.00	36.42 ± 0.00	21.97 ± 0.02	26.84 ± 0.00	33.61 ± 0.01	31.08 ± 0.01	19.48 ± 0.00
Jina-Embeddings-v2-BASE-EN-F2F	28.78 ± 0.00	36.46 ± 0.00	38.63 ± 0.00	21.79 ± 0.00	29.51 ± 0.00	36.98 ± 0.00	32.57 ± 0.00	21.32 ± 0.00
Text-Embedding-3-Small-F2F	13.61 ± 0.00	18.56 ± 0.00	15.64 ± 0.00	8.49 ± 0.00	23.78 ± 0.01	30.50 ± 0.02	28.73 ± 0.00	16.14 ± 0.01
Jina-Embeddings-v2-BASE-EN-A2C	25.46 ± 0.00	33.16 ± 0.00	34.85 ± 0.00	22.12 ± 0.00	24.98 ± 0.00	31.72 ± 0.00	29.70 ± 0.00	18.87 ± 0.00
Text-Embedding-3-Small-A2C	24.10 ± 0.00	31.53 ± 0.01	34.58 ± 0.00	19.95 ± 0.00	22.57 ± 0.02	29.57 ± 0.00	28.71 ± 0.00	17.82 ± 0.00
Jina-Embeddings-v2-BASE-EN-F2C	25.43 ± 0.00	33.00 ± 0.00	30.65 ± 0.00	20.07 ± 0.00	25.85 ± 0.00	33.23 ± 0.00	26.52 ± 0.00	18.99 ± 0.00
Text-Embedding-3-Small-F2C	13.59 ± 0.00	17.69 ± 0.00	11.75 ± 0.00	7.64 ± 0.01	24.36 ± 0.01	30.70 ± 0.00	24.06 ± 0.02	16.59 ± 0.00
OURS (PRISM)								
Jina-Embeddings-v2-BASE-EN + Llama-3.2-3B-Instruct	30.50 ± 0.00	39.19 ± 0.00	41.32 ± 0.00	25.31 ± 0.00	29.12 ± 0.00	37.90 ± 0.00	37.40 ± 0.00	22.64 ± 0.00
Text-Embedding-3-Small + Llama-3.2-3B-Instruct	29.69 ± 0.09	38.04 ± 0.10	38.78 ± 0.00	24.10 ± 0.00	27.94 ± 0.02	36.54 ± 0.00	36.18 ± 0.26	21.84 ± 0.05
Jina-Embeddings-v2-BASE-EN + GPT-4o-Mini-0718	31.36 ± 0.15	39.69 ± 0.16	42.91 ± 0.42	25.99 ± 0.04	29.86 ± 0.14	38.44 ± 0.06	36.80 ± 0.22	23.40 ± 0.03
Text-Embedding-3-Small + GPT-4o-Mini-0718	29.33 ± 0.06	37.72 ± 0.11	42.26 ± 0.47	24.97 ± 0.03	27.65 ± 0.30	36.34 ± 0.05	37.61 ± 0.10	22.50 ± 0.10

Table 11: Example of generated queries given input paper document based on method, experiments, and research questions of input document from ACL-Citations in SCIFULLBENCH. Note that due to extensive length of input documents, we only provide the title and abstract information of input paper.

Input Document Meta Data	**[Title]** ThinkSum: Probabilistic reasoning over sets using large language models **[Abstract]** Large language models (LLMs) have a substantial capacity for high-level analogical reasoning: reproducing patterns in linear text that occur in their training data (zero-shot evaluation) or in the provided context (few-shot in-context learning). However, recent studies show that even the more advanced LLMs fail in scenarios that require reasoning over multiple objects or facts and making sequences of logical deductions. We propose a two-stage probabilistic inference paradigm, ThinkSum, which reasons over sets of objects or facts in a structured manner. In the first stage (Think – retrieval of associations), a LLM is queried in parallel over a set of phrases extracted from the prompt or an auxiliary model call. In the second stage (Sum – probabilistic inference or reasoning), the results of these queries are aggregated to make the final prediction. We demonstrate the possibilities and advantages of ThinkSum on the BIG-bench suite of LLM evaluation tasks, achieving improvements over the state of the art using GPT-family models on thirteen difficult tasks, often with far smaller model variants. We also compare and contrast ThinkSum with other proposed modifications to direct prompting of LLMs, such as variants of chain-of-thought prompting. Our results suggest that because the probabilistic inference in ThinkSum is performed outside of calls to the LLM, ThinkSum is less sensitive to prompt design, yields more interpretable predictions, and can be flexibly combined with latent variable models to extract structured knowledge from LLMs. Overall, our proposed paradigm represents a promising approach for enhancing the reasoning capabilities of LLMs.
Method-Focused Query	The methodology of the paper introduces a novel two-stage probabilistic inference paradigm called ThinkSum, designed to enhance the reasoning capabilities of large language models (LLMs) when dealing with complex tasks that require reasoning over multiple objects or facts. The first stage, termed "Think," involves querying the LLM in parallel using a variety of prompts to retrieve associations or generate examples related to the task at hand. This stage employs several techniques, including elementary string manipulations (like order inversion and premise erasure), example generation, list extension, and fact generation, to create a set of candidate phrases or strings that can be used in the subsequent reasoning process. The second stage, "Sum," aggregates the outputs from the Think stage using classical probabilistic reasoning methods. This includes techniques such as mixture aggregation, product aggregation, and likelihood ratios to compute the final answer based on the probabilities derived from the generated strings. The methodology emphasizes the separation of the reasoning process from the LLMś direct output, allowing for more interpretable predictions and reducing sensitivity to prompt design. By leveraging this structured approach, ThinkSum demonstrates improved performance on various reasoning tasks compared to traditional prompting methods, showcasing its potential for broader applications in natural language understanding and reasoning.
Experiment-Focused Query	The experimental setup in this paper focuses on evaluating the ThinkSum paradigm, which enhances reasoning capabilities in large language models (LLMs) through a two-stage probabilistic inference process. The authors utilize the BIG-bench suite, a comprehensive benchmark for LLM evaluation, to assess performance across thirteen challenging tasks. Key datasets include tasks such as "Invented Words," "Odd One Out," and "Logical Deduction," which require the model to reason over multiple objects or facts. Evaluation metrics primarily involve accuracy, with comparisons made against baseline models, including direct prompting and chain-of-thought approaches. The results indicate that ThinkSum consistently outperforms these baselines, achieving state-of-the-art results even with smaller model variants. For instance, in the "Odd One Out" task, ThinkSum maintained near-human accuracy across various model sizes, demonstrating its robustness and effectiveness in probabilistic reasoning. The findings suggest that the structured approach of ThinkSum, which separates fast associative thinking from slower reasoning processes, leads to more interpretable predictions and reduced sensitivity to prompt design. Overall, the experiments validate the proposed methodology, showcasing its potential to enhance LLM performance in complex reasoning tasks.
Research-Question Focused Query	The motivation behind the paper "ThinkSum: Probabilistic reasoning over sets using large language models" stems from the observed limitations of large language models (LLMs) in performing complex reasoning tasks that require sequential logical deductions over multiple objects or facts. While LLMs excel in associative thinking and can generate responses based on patterns in their training data, they struggle with tasks that necessitate deeper reasoning and the manipulation of concepts. This paper addresses the gap by proposing a two-stage probabilistic inference framework called ThinkSum, which enhances the reasoning capabilities of LLMs by structuring the reasoning process into two distinct phases: retrieval of associations (Think) and probabilistic inference (Sum). The primary research questions posed in this work include: How can LLMs be effectively utilized to perform reasoning over sets of objects or facts? What are the advantages of a structured two-stage approach compared to traditional prompting methods? The authors aim to explore whether this structured reasoning can lead to improved performance on complex tasks, particularly in scenarios where LLMs typically falter. The key contributions of this paper are multifaceted. First, it introduces the ThinkSum paradigm, which separates the fast associative thinking from the slower, more deliberate reasoning process, allowing for more robust and interpretable predictions. Second, the paper demonstrates that ThinkSum can achieve state-of-the-art results on thirteen challenging tasks from the BIG-bench suite, often outperforming larger models with smaller variants of the GPT family. Third, it highlights the flexibility of the ThinkSum approach, showing that it can be combined with various probabilistic inference techniques to extract structured knowledge from LLMs. Finally, the authors provide a comparative analysis of ThinkSum against existing methods, such as chain-of-thought prompting, illustrating its advantages in terms of robustness to prompt design and interpretability of results. Overall, this work presents a significant advancement in the field of natural language processing, particularly in enhancing the reasoning capabilities of LLMs through structured probabilistic inference, thereby addressing a critical limitation in their application to complex reasoning tasks.

Table 12: Example of retrieved documents given input paper document from EMNLP-Citations using PRISM. Note that due to extensive length of input documents, we only provide the title and abstract information of input paper, and titles for retrieved documents. Retrieved documents that belong to ground truth candidates are highlighted in blue.

Input Document Meta Data	**[Title]** Stance Detection on Social Media with Background Knowledge **[Abstract]** Identifying users' stances regarding specific targets/topics is a significant route to learning public opinion from social media platforms. Most existing studies of stance detection strive to learn stance information about specific targets from the context, in order to determine the user's stance on the target. However, in real-world scenarios, we usually have a certain understanding of a target when we express our stance on it. In this paper, we investigate stance detection from a novel perspective, where the background knowledge of the targets is taken into account for better stance detection. To be specific, we categorize background knowledge into two categories: episodic knowledge and discourse knowledge, and propose a novel Knowledge-Augmented Stance Detection (KASD) framework. For episodic knowledge, we devise a heuristic retrieval algorithm based on the topic to retrieve the Wikipedia documents relevant to the sample. Further, we construct a prompt for ChatGPT to filter the Wikipedia documents to derive episodic knowledge. For discourse knowledge, we construct a prompt for ChatGPT to paraphrase the hashtags, references, etc., in the sample, thereby injecting discourse knowledge into the sample. Experimental results on four benchmark datasets demonstrate that our KASD achieves state-of-the-art performance in in-target and zero-shot stance detection.
Ground-Truth Document Meta Data	**[Title]** A More Advanced Group Polarization Measurement Approach Based on LLM-Based Agents and Graphs **[Title]** A Survey of Stance Detection on Social Media: New Directions and Perspectives **[Title]** Chain of Stance: Stance Detection with Large Language Models **[Title]** A Challenge Dataset and Effective Models for Conversational Stance Detection **[Title]** Mitigating Biases of Large Language Models in Stance Detection with Counterfactual Augmented Calibration **[Title]** Multi-modal Stance Detection: New Datasets and Model **[Title]** A Logically Consistent Chain-of-Thought Approach for Stance Detection **[Title]** Stance Detection with Collaborative Role-Infused LLM-Based Agents **[Title]** Prompting and Fine-Tuning Open-Sourced Large Language Models for Stance Classification **[Title]** Ladder-of-Thought: Using Knowledge as Steps to Elevate Stance Detection **[Title]** Investigating Chain-of-thought with ChatGPT for Stance Detection on Social Media
Top@30 Retrieved Document Meta Data	**[Title]** A Survey of Stance Detection on Social Media: New Directions and Perspectives **[Title]** Stance Detection with Collaborative Role-Infused LLM-Based Agents **[Title]** Enabling Contextual Soft Moderation on Social Media through Contrastive Textual Deviation **[Title]** Prompting and Fine-Tuning Open-Sourced Large Language Models for Stance Classification **[Title]** A Survey on Stance Detection for Mis- and Disinformation Identification **[Title]** Chain of Stance: Stance Detection with Large Language Models **[Title]** A Challenge Dataset and Effective Models for Conversational Stance Detection **[Title]** Stance Detection on Social Media with Fine-Tuned Large Language Models **[Title]** Stance Detection in Web and Social Media: A Comparative Study **[Title]** Mitigating Biases of Large Language Models in Stance Detection with Counterfactual Augmented Calibration **[Title]** A Benchmark for Cross-Domain Argumentative Stance Classification on Social Media **[Title]** Relative Counterfactual Contrastive Learning for Mitigating Pretrained Stance Bias in Stance Detection **[Title]** Multi-modal Stance Detection: New Datasets and Model **[Title]** TATA: Stance Detection via Topic-Agnostic and Topic-Aware Embeddings **[Title]** DEEM: Dynamic Experienced Expert Modeling for Stance Detection **[Title]** FarExStance: Explainable Stance Detection for Farsi **[Title]** Reinforcement Tuning for Detecting Stances and Debunking Rumors Jointly with Large Language Models **[Title]** Advancing Annotation of Stance in Social Media Posts: A Comparative Analysis of Large Language Models and Crowd Sourcing **[Title]** KCD: Knowledge Walks and Textual Cues Enhanced Political Perspective Detection in News Media **[Title]** Examining the Influence of Political Bias on Large Language Model Performance in Stance Classification **[Title]** Knowledge Graph Augmented Political Perspective Detection in News Media **[Title]** A Logically Consistent Chain-of-Thought Approach for Stance Detection **[Title]** Beneath the Tip of the Iceberg: Current Challenges and New Directions in Sentiment Analysis Research **[Title]** "We Demand Justice!": Towards Social Context Grounding of Political Texts **[Title]** Investigating Chain-of-thought with ChatGPT for Stance Detection on Social Media **[Title]** PAR: Political Actor Representation Learning with Social Context and Expert Knowledge **[Title]** Identifying the Adoption or Rejection of Misinformation Targeting COVID-19 Vaccines in Twitter Discourse **[Title]** Argumentative Stance Prediction: An Exploratory Study on Multimodality and Few-Shot Learning **[Title]** Detect, Investigate, Judge and Determine: A Novel LLM-based Framework for Few-shot Fake News Detection **[Title]** Reading Between the Tweets: Deciphering Ideological Stances of Interconnected Mixed-Ideology Communities

> Role: System

Instruction: You are a specialized research assistant tasked with generating a structured, detailed explanation of a scientific paper based on its **Methodology**. Your goal is to provide a clear yet comprehensive summary that makes it easy to identify relevant papers by emphasizing the methodology of given paper.

IMPORTANT
- Your explanation is going to be used as a query to retrieve similar papers **METHOD** wise.
- Make sure that your explanation can retrieve highly relevant papers easily.
- There are also other agents who are tasked with generating explanation on given paper. Unlike you, they are focused on experiments, and research questions of given paper. You must try to avoid overlap with possible explanations that the other two agents might generate.

Input:
You will be given the full text of a scientific paper. Carefully analyze its content, with a particular focus on the **METHODOLOGY** section, to extract its main approaches.

Key Considerations:
Highlight specific method/approach details, avoiding vague or overly general descriptions. Use precise language to ensure clarity while maintaining depth.

Output Format:
Generate a well-structured, detailed and yet clear paragraph that effectively captures the paper's approach, and key concepts in a concise yet informative manner. Focus on high-level insights rather than excessive detail You must not include title and abstract of given paper in your answer, and try to put it into your own words with high level reasoning after reading the paper.

Figure 5: Prompt for Method-Centric Query Optimizer LLM agent.

> Role: System

You are a specialized research assistant tasked with generating a structured, **detailed explanation of a scientific paper's experimental setup**. Your goal is to clearly outline the datasets, evaluation metrics, baselines, and key experimental findings, making it easy to understand how the paper validates its approach.

IMPORTANT

- Your explanation is going to be used as a query to retrieve similar papers **EXPERIMENT** wise.
- Make sure that your explanation can retrieve highly relevant papers easily.
- There are also other agents who are tasked with generating explanation on given paper. Unlike you, they are focused on methods, and research questions of given paper. You must try to avoid overlap with possible explanations that the other two agents might generate.

Input:

You will be provided with the full text of a scientific paper. Carefully analyze its content, paying particular attention to the **Experiments, Results, and Evaluation sections** to extract the key experimental details.

Key Considerations:

Datasets & Benchmarks: Clearly specify the datasets and benchmarks used for evaluation.
Baselines & Comparisons: Identify what methods or models the paper compares against.
Key Results & Insights: Summarize the main experimental findings without excessive detail.

Output Format:

Generate a clear, well structured and detailed paragraph that highlights the experimental methodology, datasets, evaluation metrics, baselines, and key results. Focus on high-level insights rather than excessive detail. You must not include title and abstract of given paper in your answer, and try to put it into your own words with high level reasoning after reading the paper.

Figure 6: Prompt for Experiment-Centric Query Optimizer LLM agent.

> Role: System

Instruction:You are a specialized research assistant tasked with generating a structured, **detailed explanation of a scientific paper based on its Motivation, Research Questions, and Contributions.** Your goal is to provide a clear yet comprehensive summary that makes it easy to identify relevant papers by emphasizing the core problem, key contributions, and research objectives.

IMPORTANT
- Your explanation is going to be used as a query to retrieve similar papers **RESEARCH QUESTION** wise.
- Make sure that your explanation can retrieve highly relevant papers easily.
- There are also other agents who are tasked with generating explanation on given paper. Unlike you, they are focused on experiments, and methods of given paper. You must try to avoid overlap with possible explanations that the other two agents might generate.

Input:
You will be given the full text of a scientific paper. Carefully analyze its content, with a particular focus on the Introduction and Conclusion, to extract its main contributions, research questions, and motivations.

Key Considerations:
Highlight specific motivations, research questions, and contributions, avoiding vague or overly general descriptions. Use precise language to ensure clarity while maintaining depth.

Output Format: Generate a well-structured, detailed and yet clear paragraph that effectively captures the paper's motivation, problem statement, research questions, and key contributions in a concise yet informative manner. Focus on high-level insights rather than excessive detail You must not include title and abstract of given paper in your answer, and try to put it into your own words with high level reasoning after reading the paper.

Figure 7: Prompt for Research Question-Centric Query Optimizer LLM agent.

> Role: System
Instruction: You are a specialized research assistant tasked with generating a structured, detailed explanation of a scientific paper based three different aspects.

1. Method-Specific Queries:

Generate a structured, detailed explanation of a scientific paper based on its Methodology Your goal is to provide a clear yet comprehensive summary that makes it easy to identify relevant papers by emphasizing the methodology of given paper.

IMPORTANT

- Your explanation is going to be used as a query to retrieve similar papers METHOD wise.
- Make sure that your explanation can retrieve highly relevant papers easily.

Input: You will be given the full text of a scientific paper. Carefully analyze its content, with a particular focus on the METHODOLOGY section, to extract its main approaches.

Key Considerations: Highlight specific method/approach details, avoiding vague or overly general descriptions. Use precise language to ensure clarity while maintaining depth.

Output Format: Generate a well-structured, detailed and yet clear paragraph that effectively captures the paper's approach, and key concepts in a concise yet informative manner. Focus on high-level insights rather than excessive detail You must not include title and abstract of given paper in your answer, and try to put it into your own words with high level reasoning after reading the paper.

2. Experiment-Specific Queries:

Generate a structured, detailed explanation of a scientific paper's experimental setup Your goal is to clearly outline the datasets, evaluation metrics, baselines, and key experimental findings, making it easy to understand how the paper validates its approach.

IMPORTANT

- Your explanation is going to be used as a query to retrieve similar papers EXPERIMENT wise.
- Make sure that your explanation can retrieve highly relevant papers easily.

You will be provided with the full text of a scientific paper. Carefully analyze its content, paying particular attention to the Experiments, Results, and Evaluation sections to extract the key experimental details.

Key Considerations:

Datasets & Benchmarks: Clearly specify the datasets and benchmarks used for evaluation.
Baselines & Comparisons: Identify what methods or models the paper compares against.
Key Results & Insights: Summarize the main experimental findings without excessive detail.

Output Format: Generate a clear, well structured and detailed paragraph that highlights the experimental methodology, datasets, evaluation metrics, baselines, and key results. Focus on high-level insights rather than excessive detail. You must not include title and abstract of given paper in your answer, and try to put it into your own words with high level reasoning after reading the paper.

3. Research Question-Specific Queries:

Generate a structured, detailed explanation of a scientific paper based on its Motivation, Research Questions, and Contributions. Your goal is to provide a clear yet comprehensive summary that makes it easy to identify relevant papers by emphasizing the core problem, key contributions, and research objectives.

IMPORTANT

- Your explanation is going to be used as a query to retrieve similar papers RESEARCH QUESTION wise.
- Make sure that your explanation can retrieve highly relevant papers easily.

Input: You will be given the full text of a scientific paper. Carefully analyze its content, with a particular focus on the Introduction and Conclusion, to extract its main contributions, research questions, and motivations.

Key Considerations: Highlight specific motivations, research questions, and contributions, avoiding vague or overly general descriptions. Use precise language to ensure clarity while maintaining depth.

Output Format: Generate a well-structured, detailed and yet clear paragraph that effectively captures the paper's motivation, problem statement, research questions, and key contributions in a concise yet informative manner. Focus on high-level insights rather than excessive detail You must not include title and abstract of given paper in your answer, and try to put it into your own words with high level reasoning after reading the paper.

Output: Return a structured json file for respective Method-Specific Queries, Experiment-Specific Queries, and Research Question-Specific queries, with the respective keys as "method_query", "experiment_query", and "research_question_query". Each key should contain the generated explanation as a string.

Figure 8: Prompt for Base-Agent Query Optimizers.