

When Retrieval Succeeds and Fails: Rethinking Retrieval-Augmented Generation for LLMs

Yongjie Wang^a, Yue Yu^a, Kaisong Song^b, Jun Lin^b and Zhiqi Shen^c

^aAlibaba-NTU Global e-Sustainability CorpLab, Nanyang Technological University, Singapore

^bTongyi Lab, Alibaba Group, Hangzhou, China

^cCollege of Computing & Data Science, Nanyang Technological University, Singapore

yongjie.wang@ntu.edu.sg, yuyu0022@e.ntu.edu.sg

{kaisong.sks,linjun.lj}@alibaba-inc.com, zqshen@ntu.edu.sg

Abstract

Large Language Models (LLMs) have enabled a wide range of applications through their powerful capabilities in language understanding and generation. However, as LLMs are trained on static corpora, they face difficulties in addressing rapidly evolving information or domain-specific queries. Retrieval-Augmented Generation (RAG) was developed to overcome this limitation by integrating LLMs with external retrieval mechanisms, allowing them to access up-to-date and contextually relevant knowledge. However, as LLMs themselves continue to advance in scale and capability, the relative advantages of traditional RAG frameworks have become less pronounced and necessary. Here, we present a comprehensive review of RAG, beginning with its overarching objectives and core components. We then analyze the key challenges within RAG, highlighting critical weakness that may limit its effectiveness. Finally, we showcase applications where LLMs alone perform inadequately, but where RAG, when combined with LLMs, can substantially enhance their effectiveness. We hope this work will encourage researchers to reconsider the role of RAG and inspire the development of next-generation RAG systems.

Keywords: Large Language Models, Retrieval-Augmented Generation, Knowledge Graph.

I. Introduction

Large language models (LLMs) demonstrate extraordinary performance across a wide range of applications, including medical diagnosis [51], behavioral agency [38, 43], and emotional assistance [47]. However, relying solely on their static internal knowledge often leads to inaccurate or fabricated outputs in domain-specific or knowledge-intensive tasks, a phenomenon commonly referred to as hallucination [20, 34]. To mitigate these limitations, Retrieval-Augmented Generation (RAG) [26, 33] has been proposed to dynamically integrate external, query-relevant knowledge into the generation process, and thus complements the knowledge implicitly encoded in the parameters of LLMs. Therefore, RAG has attracted significant attention and demonstrated notable improvements across a variety of real-world applications.

The success of RAG largely stems from the inherent in-context learning (ICL) capabilities [9, 36] of LLMs, which enable them to condition their outputs on externally supplied evidence and dynamically adapt their reasoning to new contextual inputs. Therefore, advanced RAG research has increasingly focused on enhancing the quality of information provided to LLMs, enabling them to address more complex and knowledge-intensive tasks [10, 16, 17, 25]. The first line of RAG research focuses on transforming the original queries to facilitate more effective subsequent retrieval [11, 32]; Some other studies aim to fine-tune the embedding model to accurately retrieve the most relevant content [27, 55]; Recently, much of the research has focused on developing more effective knowledge indexing strategies to facilitate improved retrieval. For example, GraphRAG [10] and HippoRAG [16] incorporate knowledge graphs (KGs) to index the external database, thereby supporting cross-document retrieval; KAG [28] integrates the semantic reasoning capabilities of knowledge graphs by traversing along the KG; Agentic RAG [41] combines both LLM agents and

RAG methods for addressing complex reasoning tasks that require multi-round LLM revoking.

In the era of increasingly powerful LLMs such as DeepSeek-R1 [14] and Qwen-3 [53], the necessity of RAG is perceived as less compelling, which in turn diminishes recognition of its advances. Although prior RAG research has produced remarkable advances, it is necessary to reconsider whether RAG still effectively complements increasingly powerful LLMs and to identify the key challenges it faces in the current era. In this perspective article, we review recent literature on RAG, first identifying the weaknesses that hinder the reliability and performance of current RAG systems. We then highlight the irreplaceable benefits of RAG, which continue to complement and enhance modern LLMs. The limitations of RAG primarily lie in several aspects: (1) insufficient analysis of what LLMs have already learned, which is essential for determining when to trigger retrieval; (2) inadequate intent analysis in complex questions, which affects the identification of query keywords; (3) unresolved knowledge conflicts within external databases; and (4) a limited understanding of how in-context learning operates within the retrieval-augmented framework. However, we also identify several scenarios that still require the integration of RAG, including knowledge-intensive applications, personalized or private information access, and real-time knowledge integration. By reevaluating these weaknesses and strengths, we aim to explore how RAG system design should evolve alongside the continuous advancement of base LLMs. We hope this work will reaffirm the role of RAG and inspire future research toward more robust, next-generation RAG systems.

The remainder of this paper is organized as follows. Section II. introduces the RAG mechanism and its key modules. Section III. illustrates the major challenges in RAG and discusses potential avenues for future research. Section IV. outlines the applications in which LLMs can benefit from RAG. Finally, Section V. presents the conclusion.

II. Anatomy of RAG: Architecture and Key Components

In the typical RAG scenario, a user poses a question to LLMs. Owing to intrinsic limitations, LLMs often lack the capacity to provide comprehensive and reliable responses. RAG bridges this gap by retrieving relevant knowledge from external databases. The retrieved content, combined with the user’s query, forms an enriched prompt that enables the LLM to generate more accurate, informative, and contextually grounded answers.

A. *Mission of RAG*

Similar to traditional retrieval systems [37, 40], implementing an ideal retriever that consistently returns tangible content for a given query remains an elusive goal in RAG. For RAG to succeed, retrieval must achieve both comprehensiveness (high recall) and relevance (high precision). In the following sections, we discuss these two objectives separately.

- **High recall.** Recall that RAG is specifically designed to compensate for the limitations of LLMs when addressing knowledge-intensive queries. Therefore, the ultimate goal of the retriever is to ensure that all documents relevant to answering a query—particularly those containing information beyond the LLM’s parametric knowledge—are successfully returned.
- **High precision.** LLMs function as context-conditioned generators, meaning that each retrieved passage serves as a conditioning factor influencing the generation process. That is to say, if the retriever returns noise or irrelevant text, the LLM may incorporate this misleading evidence, producing hallucinations or spurious justifications [6, 8]. In addition, if useful information is surrounded by irrelevant content in the middle, the LLM may perform

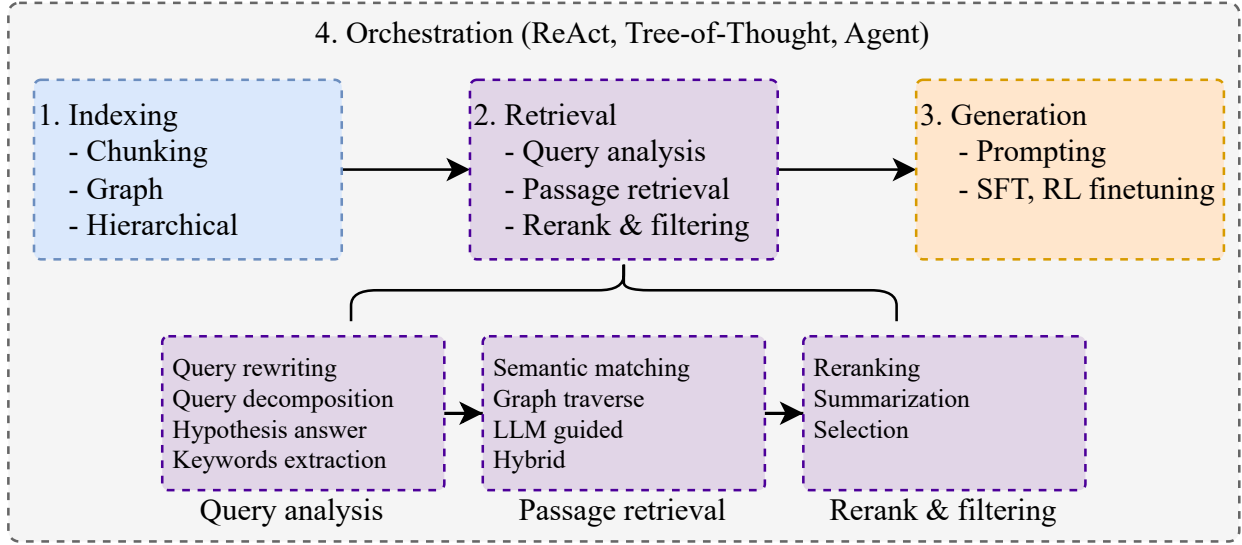


Fig 1 The overall framework of RAG and its four core modules.

worse—essentially becoming “lost in the middle [30].” Therefore, we should minimize the irrelevant content returned.

Generally speaking, retrieval systems face an inherent trade-off between recall and precision. Fetching more documents improves recall but inevitably increases the risk of introducing noise, thereby reducing precision; whereas enhancing precision by suppressing noise may result in the omission of critical information. Achieving an effective balance between recall and precision thus remains a fundamental challenge in retrieval design.

B. Key Modules in RAG

As illustrated in Figure 1, the RAG system can be divided into four structured modules: indexing, retrieval, generation, and orchestration. Each module represents an essential component of the RAG process, responsible for achieving a distinct high-level objective. In the following sections, we describe these four modules in detail.

Indexing Module: At this stage, the indexing module is responsible for organizing and structuring the vast corpus of external sources. A common and straightforward approach is to partition documents into manageable chunks, which are then encoded using representation methods such as BM25 or LLM-based embeddings (e.g., SBERT [39], OpenAI text-embedding-3). This simple strategy facilitates efficient retrieval by comparing the similarity between the query embedding and chunk embeddings, enabling seamless integration with LLMs [26].

Despite its effectiveness for semantic similarity-based retrieval, document chunking suffers from two inherent limitations. First, it fails to maintain global coherence across multiple contexts. Second, embeddings of isolated chunks are insufficient to capture fine-grained relationships such as causal dependencies or hierarchical taxonomies. To address these challenges, knowledge graph (KG) enhanced RAG has been proposed [10, 16, 47]. In this paradigm, external sources are transformed into KGs, where nodes denote entities or concepts and edges encode their relationships. Building on this structure, hierarchical clustering [10] further organizes entities into multi-level communities, thereby capturing both global coherence and layer-wise conceptual structures.

One key challenge of the index module is developing an effective indexing system that can accurately match user queries with the most relevant information. Another challenge involves managing heterogeneous data sources—such as PDF, Markdown, HTML, and Word documents.

Retrieval Module: This module bridges the user query and the indexed database, with the goal of retrieving the most relevant data in response to the query. It can be further divided into three sequential steps: query analysis, passage retrieval, and reranking and filtering.

- **Query analyzer.** This stage focuses on analyzing users’ questions to infer their intent and generate optimized query terms that enhance downstream retrieval performance. The opti-

mized query terms should be both semantically aligned with the user's original query and sufficiently specific to capture the most relevant information. Key techniques include: Question rewriting [31], which involves rephrasing the query from multiple perspectives to improve its expressiveness and coverage; Query decomposition [56], which aims to break down a complex question into simpler sub-questions following the divide-and-query paradigm; Answer inferring [58], which prompts the LLM to generate hypothetical answers in order to guide retrieval toward semantically relevant content; and keyword extraction [47], which identifies salient—often domain-specific—terms from the query and uses them to retrieve specialized or niche knowledge.

- **Passage retrieval.** In the RAG framework, retrieval algorithm is designed in accordance with the indexing system. For chunking-based database, retrieval is typically performed by comparing the similarity between the representation of a query (or its expanded form) and the chunks in the indexed database. Common representation methods include sparse encoders (e.g., BM25) and dense embeddings (e.g., SBERT [39]). More sophisticated multi-stage pipelines hybridize these approaches—for example, by first performing coarse retrieval with BM25 to achieve high recall (e.g., 100+ candidates), followed by semantic retrieval using LLM embeddings to retain the top-K semantically relevant chunks, thus effectively balancing the recall-precision tradeoff.

Advanced indexing systems further facilitate more sophisticated retrieval. For example, knowledge graph-based indexing enables traversal along the graph structure to identify relevant entities and their interrelationships [10]. Moreover, community structures formed through graph clustering offer a global and comprehensive understanding that supports ques-

tion answering requiring cross-document retrieval. Collectively, these techniques establish a hierarchical retrieval ecosystem that dynamically adapts to query complexity, knowledge domain, and computational constraints.

- **Rerank and filtering.** Once relevant passage is retrieved, the rerank and filtering step is designed to refine the results, enhancing precision and filtering out irrelevant noise. Reranking techniques [13] reorder the retrieved content based on the query relevance, retaining only the top relevant results. Since feeding all retrieved documents—including irrelevant content—directly into LLMs can dilute attention and cause information distraction, summarization techniques are often employed to retain only the most informative segments, thereby reducing the context length passed to LLMs [12, 47].

Generation Module: This module instructs LLMs to combine the retrieved data with the user query to produce the final output. Generation performance primarily depends on the LLM’s capability for task understanding and the quality of the retrieved information. Therefore, effective prompt engineering [54] is important to ensure that LLM uses the retrieved documents into its intrinsic generation, although there is no universal principle to guide the prompt design. During responses generation, LLMs could encounter conflicting information—either from multiple retrieved sources or between external evidence and their parametric knowledge, and thus LLM should design to suppress inaccurate content from noisy or irrelevant documents [44, 46]. Correspondingly, some methods [29] fine-tune LLMs specifically for the RAG scenario by training them to better distinguish relevant documents from irrelevant ones.

Orchestration Module: In RAG systems, the above mentioned modules need to be executed sequentially or processed in parallel, depending on system design and task requirements. There-

fore, effective workflow planning is essential for improving system efficiency and achieving the desired outcomes. The orchestration module is designed to manage and coordinate interactions among modules and data flows, determining which components to activate based on the specific requirements of a given query. By orchestrating the process, it enables the system to adapt dynamically to varying scenarios, thereby enhancing versatility and efficiency.

III. Challenges and Future Directions

Despite recent advances, RAG does not always guarantee superior performance compared with responses generated solely by LLMs [4, 6, 52]. In the following, we outline the key challenges in RAG systems and discuss potential solutions to address them.

A. When Should I Retrieve? The Unawareness of LLM Knowledge Boundary

It is important to note that RAG was originally introduced to compensate for knowledge scarcity in domain-specific tasks. However, a major blind spot of most RAG methods lies in their failure to assess what LLMs already know and what they do not. Instead, these methods often directly apply retrieval over large-scale external sources and report performance improvements without examining necessity or relevance. Meanwhile, LLMs such as DeepSeek [14] and Qwen3 [53] have become increasingly powerful, and many fine-tuned models have emerged for specialized domains (e.g., HuatuoGPT-o1 [7] in medical diagnostics and CharacterGLM [57] in role-playing), leaving limited room for RAG approaches to demonstrate their comparative advantages.

We posit that retrieval is not always necessary for all questions [19, 21] and should instead be triggered adaptively based on the model’s capability. A representative example is provided in prior work [21], where, on the Natural Questions dataset, retrieval-triggering reduced API calls

by approximately 40% without any loss in accuracy. Therefore, it is crucial to first determine whether an LLM can answer a given question solely using its internal knowledge. To this end, uncertainty-based methods can be employed to evaluate prediction variability—for instance, semantic uncertainty [23], self-uncertainty [22], prediction confidence [24], and related approaches [15, 49]. RAG is then activated only when the LLM fails to produce a confident or consistent prediction on its own.

B. What to Retrieve? The Ineffectiveness of retrieval method

While RAG excels at fact-oriented questions, it often struggles with complex reasoning tasks (e.g., multi-hop question answering, mathematical reasoning), which require a deep understanding of question intent and the ability to perform step-by-step logical inference.

Traditional RAG systems typically treat the query as a single unit, extracting lexical features through statistical methods or dense embeddings from LLM-based models. However, simply selecting the Top-K similar chunks cannot adequately capture the extrinsic context or nuanced intent underlying complex queries. For example, a query such as “If gravity were 10× stronger, how would architectural design evolve?” might retrieve general physics principles about gravity in reality while overlooking speculative engineering literature. Such retrieved factual data will inevitably lead to an elevated risk of incorrect or hallucinated responses [2].

Knowledge graph-based indexing facilitates advanced reasoning through traversal along connected edges, it still falls short of meeting the demands of complex reasoning tasks. For instance, frequent entities are often densely connected, which substantially expands the search space and introduces noise during traversal. Current retrieval strategies in KG-RAG generally fall into two categories: (i) K-hop neighborhood, which first identify a seed entity through similarity search and

then include its k-hop neighbors, and (ii) LLM-guided search, which leverage language models to evaluate the plausibility of candidate paths. Both approaches exhibit notable limitations—the former often introduces irrelevant entities due to uncontrolled expansion, while the latter is computationally expensive and prone to inconsistency. In addition, for such static graph structures, it is difficult to handle queries involving temporal information, such as identifying the initial, intermediate, and final steps of an execution process. As a result, retrievers struggle to consistently identify the most relevant paths from the KGs for complex reasoning questions.

These retrieval failures can often be attributed to two factors: (1) misinterpretation of the user query, and (2) ineffectiveness of the retrieval method, which together lead to retrieved content that lacks contextual relevance or appropriateness. To facilitate effective retrieval, agent-based frameworks have been introduced to analyze complex reasoning tasks and decompose them into multiple sequential or parallel steps. Within these frameworks, RAG is collaborated with active agents to adaptively retrieve from external knowledge, and thus overcome the challenges of traditional RAG methods, giving rise to the concept of Agentic RAG [41]. Despite the progress in Agentic RAG, future research should continue to focus on achieving a deeper understanding of user intent and developing a unified paradigm capable of adapting to diverse and complex tasks.

C. What Should I Trust? On the Risks of Unverified Data Sources

RAG is designed to mitigate factual errors by incorporating external knowledge during the indexing process. However, most RAG methods explicitly assume that external knowledge is inherently reliable and trustworthy, without additional verification [10, 26]. In real life, factual inaccuracies often present in retrieval databases. For instance, even the widely used medical database PubMed has been shown to contain fraudulent data and publications, raising concerns about their preva-

lence and impact [35]. These issues highlight the importance of curating high-quality knowledge bases—such as PrimeKG [5]—to fulfill the requirements of LLM-based RAG systems. As tool-augmented agents become increasingly prevalent [12], constructing high-quality, retrieval-efficient databases tailored for tool integration emerges as a promising avenue for future investigation.

D. How does RAG Work? Linking Retrieval to Mechanism of In-Context Learning

RAG provides rich, relevant external context that in-context learning (ICL) can leverage to generate higher-quality outputs. However, the mechanisms for resolving conflicts between retrieved evidence and the model’s parametric memory remain unclear, often leading to unpredictable behavior that undermines RAG’s effectiveness. Recent benchmarking studies [18] show that when exposed to either correct or incorrect snippets, LLMs tend to rely heavily on retrieved content regardless of its veracity. This underscores the importance of accurate retrieval; yet, even with perfect sources, LLMs may still produce incorrect responses. Such limitations impose an inherent upper bound on RAG’s achievable performance, and the factors determining this bound remain poorly understood.

To fully exploit the power of RAG, it is essential to understand the mechanisms of ICL [36] for anticipating and governing model behavior in long-context settings with explainability techniques [42, 48, 50]. In particular, we seek to characterize how information flows between the question and retrieved knowledge—for example, which attention heads mediate grounding, how evidence competes with parametric priors, and when positional or recency biases dominate—using tools such as attention rollout/flow, causal tracing and patching, representation probing [49], and token-level attribution.

E. RAG vs Long-context LLM

Long-context LLMs (e.g., GPT-4 [1], Claude 3) are capable of processing substantially longer contexts—ranging from hundreds of thousands to over one million tokens—thereby mitigating the reliance on external retrieval. These models are particularly well-suited to tasks such as reasoning over lengthy documents (e.g., books) or multi-document compilations, where ingesting full texts directly is preferable to retrieval.

However, long-context LLMs also encounter several challenges [3, 45]. First, long-context LLMs must suffer from persistent knowledge cutoffs that limit their access to recent information. Second, the quadratic scaling of attention mechanisms leads to substantially higher inference costs. Third, extending the context window increases the likelihood of introducing irrelevant or noisy information. Lastly, as the context length grows, the availability of high-quality training and evaluation data diminishes rapidly, causing long-context LLMs to perform less effectively in practice than reported in controlled benchmarks.

Both approaches address long-document processing but exhibit distinct advantages. Long-context LLMs are more effective when evidence is evenly distributed across multiple documents, whereas RAG performs better in tasks with sparse evidence, assuming accurate retrieval. Additionally, RAG enables access to up-to-date and private information without retraining. A unified framework that integrates RAG with long-context LLMs can leverage their complementary strengths—precise factual retrieval and holistic cross-document reasoning—yielding greater reliability and robustness than either approach alone.

IV. RAG Applications

Although recent LLMs demonstrate strong performance across a wide range of tasks, their inherent limitations continue to drive research on RAG, especially in the following fields.

- **Knowledge-Intensive Applications.** When tasked with knowledge-intensive applications such as drug dosing or rare disease diagnostics, LLMs often perform sub-optimally. In such cases, RAG is particularly powerful, as it enables access to high-quality domain-specific databases and helps mitigate the limitations of parametric memory. Using the retrieved content, the LLM can ground its responses in authoritative evidence, generate more accurate and trustworthy outputs.
- **Private Knowledge Management.** Another important application of RAG lies in leveraging private or proprietary knowledge sources, such as enterprise documentation and personal notes. LLMs cannot memorize such data due to the wide diversity and restricted accessibility. In these scenarios, RAG enables customized and secure knowledge retrieval, ensuring that LLMs generate responses aligned with organizational or individual needs while preserving data privacy. A typical example is the multi-turn conversation where the LLM acts as an agent interacting with users. The conversational history serves as a retrieval database to provide personalized context, thereby enabling more coherent and engaging dialogue.
- **Real-Time Knowledge Integration.** RAG is also effective in domains where knowledge evolves rapidly, such as news, financial markets, and regulatory updates. By continuously retrieving up-to-date information, LLMs can function as information extractors and summarizers, generating responses for questions that reflect the latest developments.

V. Conclusion

As large language models (LLMs) continue to advance in scale and capability, the relative advantages of traditional RAG frameworks have become less pronounced, reshaping RAG's role in the evolving LLM landscape. This paper provides a systematic review of RAG systems and their core modules, followed by an analysis of key challenges that limit their effectiveness. Addressing these limitations is essential for ensuring that RAG systems remain robust, adaptive, and complementary to next-generation LLMs. Finally, we outline application domains where RAG remains indispensable for mitigating LLM inefficiencies, underscoring the need for close collaboration between RAG and LLMs.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Garima Agrawal, Tharindu Kumarage, Zeyad Alghamdi, and Huan Liu. Mindful-rag: A study of points of failure in retrieval augmented generation. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, pages 607–611. IEEE, 2024.
- [3] Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

- Papers*), pages 3639–3664, Vienna, Austria, July 2025. Association for Computational Linguistics.
- [4] Shuyang Cao, Karthik Radhakrishnan, David Rosenberg, Steven Lu, Pengxiang Cheng, Lu Wang, and Shiyue Zhang. Evaluating the retrieval robustness of large language models. *arXiv preprint arXiv:2505.21870*, 2025.
- [5] Payal Chandak, Kexin Huang, and Marinka Zitnik. Building a knowledge graph to enable precision medicine. *Scientific Data*, 10(1):67, 2023.
- [6] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762, 2024.
- [7] Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. Huatuogpt-o1, towards medical complex reasoning with llms. *arXiv preprint arXiv:2412.18925*, 2024.
- [8] Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 719–729, 2024.
- [9] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.
- [10] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint*

arXiv:2404.16130, 2024.

- [11] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada, July 2023. Association for Computational Linguistics.
- [12] Shanghua Gao, Richard Zhu, Zhenglun Kong, Ayush Noori, Xiaorui Su, Curtis Ginder, Theodoros Tsiligkaridis, and Marinka Zitnik. Txagent: An ai agent for therapeutic reasoning across a universe of tools. *arXiv preprint arXiv:2503.10970*, 2025.
- [13] Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. Re2g: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715, 2022.
- [14] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.
- [15] Xu Guo. Measuring reasoning utility in llms via conditional entropy reduction. *arXiv preprint arXiv:2508.20395*, 2025.
- [16] Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [17] Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models, 2025.

Yongjie Wang, Yue Yu, Kaisong Song Jun Lin, and Zhiqi Shen

- [18] Yukun Huang, Sanxing Chen, Hongyi Cai, and Bhuwan Dhingra. To trust or not to trust? enhancing large language models’ situated faithfulness to external contexts. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [19] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7029–7043, 2024.
- [20] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), March 2023.
- [21] Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.
- [22] Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty. *arXiv preprint arXiv:2502.18581*, 2025.
- [23] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023.
- [24] Joshua Ong Jun Leang, Zheng Zhao, Aryo Pradipta Gema, Sohee Yang, Wai-Chung Kwan, Xuanli He, Wenda Li, Pasquale Minervini, Eleonora Giunchiglia, and Shay B Cohen. Picsar: Probabilistic confidence selection and ranking. *arXiv preprint arXiv:2508.21787*, 2025.
- [25] Jonathan Leung, Yongjie Wang, and Zhiqi Shen. Knowledge retrieval in llm gaming: A shift

- from entity-centric to goal-oriented graphs. *arXiv preprint arXiv:2505.18607*, 2025.
- [26] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.
- [27] Xianming Li and Jing Li. AoE: Angle-optimized embeddings for semantic textual similarity. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1825–1839, Bangkok, Thailand, August 2024. Association for Computational Linguistics.
- [28] Lei Liang, Zhongpu Bo, Zhengke Gui, Zhongshu Zhu, Ling Zhong, Peilong Zhao, Mengshu Sun, Zhiqiang Zhang, Jun Zhou, Wenguang Chen, Wen Zhang, and Huajun Chen. Kag: Boosting llms in professional domains via knowledge augmented generation. In *Companion Proceedings of the ACM on Web Conference 2025, WWW '25*, page 334–343, New York, NY, USA, 2025. Association for Computing Machinery.
- [29] Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. Ra-dit: Retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2023.
- [30] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 11:157–173, 2024.
- [31] Xinbei Ma, Yeyun Gong, Pengcheng He, hai zhao, and Nan Duan. Query rewriting in retrieval-augmented large language models. In *The 2023 Conference on Empirical Meth-*

Yongjie Wang, Yue Yu, Kaisong Song Jun Lin, and Zhiqi Shen

ods in Natural Language Processing, 2023.

- [32] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting in retrieval-augmented large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore, December 2023. Association for Computational Linguistics.
- [33] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, 2023.
- [34] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online, July 2020. Association for Computational Linguistics.
- [35] Consolato Gianluca Nato and Federico Bilotta. Fraud in medical publications. *Anesthesiology Clinics*, 42(4):607–616, 2024.
- [36] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.
- [37] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford infolab, 1999.
- [38] Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and

- Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023.
- [39] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. 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)*, pages 3982–3992, 2019.
- [40] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.
- [41] Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*, 2025.
- [42] Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. Rethinking interpretability in the era of large language models. *arXiv preprint arXiv:2402.01761*, 2024.
- [43] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024.
- [44] Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Retrieval-augmented generation with conflicting evidence. In *Second Conference on Language Modeling*, 2025.
- [45] Minzheng Wang, Longze Chen, Fu Cheng, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, Yunshui Li, Min Yang, Fei Huang, and Yongbin Li. Leave no document behind: Benchmarking long-context LLMs with extended multi-doc QA. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the*

Yongjie Wang, Yue Yu, Kaisong Song Jun Lin, and Zhiqi Shen

- 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5627–5646, Miami, Florida, USA, November 2024. Association for Computational Linguistics.
- [46] Yike Wang, Shangbin Feng, Heng Wang, Weijia Shi, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. Resolving knowledge conflicts in large language models. In *First Conference on Language Modeling*, 2024.
- [47] Yongjie Wang, Jonathan Leung, and Zhiqi Shen. Rolerag: Enhancing llm role-playing via graph guided retrieval. *arXiv preprint arXiv:2505.18541*, 2025.
- [48] Yongjie Wang, Xiaoqi Qiu, Yu Yue, Xu Guo, Zhiwei Zeng, Yuhong Feng, and Zhiqi Shen. A survey on natural language counterfactual generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4798–4818, 2024.
- [49] Yongjie Wang, Yibo Wang, Xin Zhou, and Zhiqi Shen. Response uncertainty and probe modeling: Two sides of the same coin in llm interpretability? *arXiv preprint arXiv:2505.18575*, 2025.
- [50] Yongjie Wang, Tong Zhang, Xu Guo, and Zhiqi Shen. Gradient based feature attribution in explainable ai: A technical review. *arXiv preprint arXiv:2403.10415*, 2024.
- [51] Kevin Wu, Eric Wu, Kevin Wei, Angela Zhang, Allison Casasola, Teresa Nguyen, Sith Riantawan, Patricia Shi, Daniel Ho, and James Zou. An automated framework for assessing how well llms cite relevant medical references. *Nature Communications*, 16(1):3615, 2025.
- [52] Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. How easily do irrelevant inputs skew the responses of large language models? In *First Conference on Language Modeling*, 2024.
- [53] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint*

arXiv:2505.09388, 2025.

- [54] Xiang Zhang, Juntao Cao, Chenyu You, and Dujian Ding. Why prompt design matters and works: A complexity analysis of prompt search space in LLMs. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32525–32555, Vienna, Austria, July 2025. Association for Computational Linguistics.
- [55] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.
- [56] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*, 2023.
- [57] Jinfeng Zhou, Zhuang Chen, Dazhen Wan, Bosi Wen, Yi Song, Jifan Yu, Yongkang Huang, Pei Ke, Guanqun Bi, Libiao Peng, JiaMing Yang, Xiyao Xiao, Sahand Sabour, Xiaohan Zhang, Wenjing Hou, Yijia Zhang, Yuxiao Dong, Hongning Wang, Jie Tang, and Minlie Huang. CharacterGLM: Customizing social characters with large language models. In Franck Dernoncourt, Daniel Preoȃiuc-Pietro, and Anastasia Shimorina, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1457–1476, Miami, Florida, US, November 2024. Association for Computational Linguistics.
- [58] Weichao Zhou, Jiabin Zhang, Hilaf Hasson, Anu Singh, and Wenchao Li. Hyqe: Ranking

Yongjie Wang, Yue Yu, Kaisong Song Jun Lin, and Zhiqi Shen

contexts with hypothetical query embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13014–13032, 2024.