Graph-Based Retrieval-Augmented Generation: Applications, Challenges, Solutions, and Opportunities

Angela Bonifati
Lyon 1 University, CNRS
Liris & IUF
Lyon, France
angela.bonifati@univ-lyon1.fr

Yixiang Fang
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
fangyixiang@cuhk.edu.cn

Sibo Wang
The Chinese University of Hong Kong
Hong Kong SAR, China
swang@se.cuhk.edu.hk

Yingli Zhou
Lyon 1 University, CNRS
Liris
Lyon, France
yingli.zhou@univ-lyon1.fr

ABSTRACT

Graph-Based Retrieval-Augmented Generation (RAG) has proven effective in integrating structured external knowledge into Large Language Models (LLMs), improving their factual accuracy, adaptability, interpretability, and trustworthiness. Accordingly, a large number of graph-based RAG methods have been proposed in recent years by the database, data mining, machine learning, and natural language processing communities. In this tutorial, we first highlight the real-world applications of graph-based RAG and the unique challenges that need to be addressed. We then classify and compare representative graph-based RAG methods from well-known top data management and data mining venues. Afterward, we outline the core pipeline stages of graph-based RAG and analyze them from a data management perspective, including graph modeling, graph repair, index construction and maintenance, and retrieval, rather than treating graph-based RAG solely as an LLM application. Finally, we discuss how graph-based RAG can be extended to multimodal data and identify promising future research directions.

PVLDB Reference Format:
Angela Bonifati, Yixiang Fang, Sibo Wang, and Yingli Zhou. Graph-Based Retrieval-Augmented Generation: Applications, Challenges, Solutions, and Opportunities. PVLDB, 19(12): 4900 - 4905, 2026.
doi:10.14778/3827998.3828147

1 INTRODUCTION

The development of Large Language Models (LLMs) like GPT-5 [1], Qwen 3 [66], and Claude 4.6 [4] has sparked a revolution in the field of artificial intelligence [36, 44, 60, 73]. Despite their remarkable comprehension and generation capabilities, LLMs may still generate incorrect outputs due to a lack of domain-specific knowledge, real-time updated information, and proprietary knowledge, which are outside LLMs' pre-training corpus, known as "hallucination" [46].

To bridge this gap, the *Retrieval Augmented Generation (RAG)* technique [16, 23, 24, 64, 71, 78] has been proposed, which supplements LLM with external knowledge to enhance its factual accuracy and trustworthiness. Consequently, RAG techniques have been widely applied in various fields, especially in domains where LLMs need to generate reliable outputs, such as healthcare [36, 56, 73], finance [44], and education [60]. Moreover, RAG has proven highly useful in many data management tasks, including NL2SQL [13], data cleaning [14, 30, 48], knob tuning [17], DBMS diagnosis [74, 75], and SQL rewrite [32, 53]. Due to the important role of the RAG technique in LLM-based applications, numerous RAG methods have been proposed in the past three years [23]. Among these methods, the state-of-the-art RAG approaches typically use the graph data as the external data (also called graph-based RAG), since they capture the rich semantic information and link relationships between entities. Given a user query Q, the key idea of graph-based RAG methods is to retrieve relevant information (e.g., nodes, edges, subgraphs, or textual data) from the graph and then feed them along with Q as a prompt into LLM to generate answers.

Recently, several mainstream database systems have started supporting graph-based RAG, including PostgreSQL [47], Neo4j [43], and Databricks [10]. At the same time, graph-based RAG has become a core component in modern graph-native agentic systems, such as LangGraph [27] and Chat2Graph [7]. Following the success of graph-based RAG, researchers from database, data mining, machine learning, and natural language processing have proposed many efficient and effective methods [6, 11, 12, 18, 25, 28, 29, 49, 50, 59, 61–63]. These methods differ substantially in graph construction, indexing, and information retrieval. To this end, in this tutorial, we aim to provide a comprehensive review of graph-based RAG from a unified framework. We focus on the main stages of graph-based RAG systems, including graph construction, schema design and repair, index building and maintenance, graph retrieval, extensions to other modalities, and open research challenges. The goal is to help the VLDB community understand graph-based RAG not only as an LLM application, but also as a data management problem involving graph modeling, indexing, maintenance, and query processing.

Recent related tutorials. Recent tutorials have studied machine learning for graph data management and the use of LLMs and data agents in data management [26, 39, 55]. These works mainly examine how learning-based or agentic techniques can support traditional data-management and graph-processing tasks, such as query processing, optimization, analytics, and data-centric automation. In contrast, our focus is on graph-based RAG itself, where graphs serve as the core structure for managing, indexing, and retrieving knowledge to ground LLM outputs. To the best of our knowledge, this graph-based RAG tutorial is a fresh offering for VLDB and is not a revision of prior tutorials.

Proceedings of the VLDB Endowment, Vol. 19, No. 12 ISSN 2150-8097.
doi:10.14778/3827998.3828147

Table 1: Classification of representative graph-based RAG methods. Graph Type denotes the constructed graph structure; Index Component denotes indexed graph elements; Retrieval Primitive denotes the query-derived retrieval signal; Retrieval Granularity denotes the retrieved evidence unit.

Method	Graph Type (Stage ❶)	Index Component (Stage ❷)	Retrieval Primitive (Stage ❸)	Retrieval Granularity (Stage ❹)	Specific QA	Abstract QA
`RAPTOR` [49]	Tree	Tree node	Question vector	Tree node	✓	✓
`SiReRAG` [69]	Tree	Sentence, Entity	Question vector	Tree node, Chunk	✓	✓
`KGP` [62]	Passage Graph	Chunk	Question	Chunk	✓	×
`LinearRAG` [79]	Multi-Partite Graph	Entity, Chunk	Activated entities	Chunk	✓	×
`HippoRAG` [19]	Knowledge Graph	Entity	Entities in question	Chunk	✓	×
`HippoRAG2.0` [20]	Knowledge Graph	Entity, Chunk	Entities in question	Chunk	✓	×
`G-Retriever` [21]	Knowledge Graph	Entity, Relationship	Question vector	Subgraph	✓	×
`ToG` [51]	Knowledge Graph	Entity, Relationship	Question	Subgraph	✓	×
`ToG-2` [40]	Textual Knowledge Graph	Entity, Relationship, Chunk	Entities in question	Path, Chunk	✓	×
`FastToG` [34]	Knowledge Graph	Entity, Relationship	Entities in question	Community, Path	✓	×
`DALK` [29]	Knowledge Graph	Entity	Entities in question	Subgraph	✓	×
`GNN-RAG` [41]	Knowledge Graph	Subgraph, Path	Entities in question	Subgraph	✓	×
`SubgraphRAG` [31]	Knowledge Graph	Entity, Relationship	Entities in question	Subgraph	✓	×
`GRAG` [22]	Knowledge Graph	Entity, Relationship	Question vector	Subgraph	✓	×
`PathRAG` [8]	Knowledge Graph	Path, Relationship	Question vector	Path	✓	✓
`KAG` [33]	Knowledge Graph	Entity, Fact, Chunk	Question	Fact, Chunk, Subgraph	✓	×
`GraphRAG` [12]	Textual Knowledge Graph	Entity, Community	Question vector	Entity, Relationship, Chunk, Community	✓	✓
`LEGO-GraphRAG` [6]	Textual Knowledge Graph	Entity, Relationship, Path, Subgraph	Question	Entity, Relationship, Path, Subgraph	✓	✓
`LightRAG` [18]	Rich Knowledge Graph	Entity, Relationship	keywords in question	Entity, Relationship, Chunk	✓	✓
`ArchRAG` [59]	Rich Knowledge Graph	Entity, Community	Question vector	Entity, Relationship, Community, Chunk	✓	✓
`KET-RAG` [25]	Multi-Partite Graph	Entity, Relationship	Question vector	Entity, Chunk	✓	×
`Clue-RAG` [50]	Multi-Partite Graph	Entity, Chunk, Knowledge unit	Entities in question	Entity, Chunk	✓	×
E^2`GraphRAG` [72]	Tree	Entity, Chunk	Question vector	Chunk	✓	✓
`GFM-RAG` [38]	Knowledge Graph	Node, Relationship, Subgraph	Question vector	Subgraph	✓	✓
`HyperGraphRAG` [37]	Hypergraph	Hyperedge, Entity	Entities in question	Hyperedge, Chunk	✓	✓
`HyperRAG` [35]	Hypergraph	Hyperedge, Entity	Entities in question	Entity, Hyperedge, Chunk	✓	✓
`GraphRAG-R1` [67]	Knowledge Graph	Entity, Relationship	Question vector	Subgraph	✓	×

2 TUTORIAL OUTLINE

This will be a 1.5 hours tutorial. Below is our outline.

2.1 Introduction, Applications, and Challenges (15 minutes)

This part aims to provide the necessary background to the audience. It will consist of an introduction to the research field to highlight the popularity and applications of graph-based RAG. We will first introduce the well-known pipeline of graph-based RAG [46, 77], discuss the interesting applications of graph-based RAG over data management scenarios [52, 53, 74], and finally present the key challenges of graph-based RAG. We will also cover another important aspect, namely a feature-based review of commercial database or data analytics systems that support RAG and graph-based RAG.

Subsequently, we present two key challenges in graph-based RAG: (1) how to build graph-based RAG methods that are both accurate and efficient, and (2) how to develop graph-based RAG methods that can satisfy diverse requirements across different scenarios, data modalities, and application settings. To address the first challenge, recent studies have explored a variety of techniques to optimize different stages of the graph-based RAG pipeline, including lightweight graph construction, hybrid indexing, and efficient online retrieval. For the second challenge, researchers often develop new graph structures, retrieval mechanisms, and system designs by considering the characteristics of different tasks, query types, data modalities, and application requirements [11, 59, 61, 77].

2.2 A Unified Framework and Prototype System of Graph-Based RAG (45 minutes)

In this part, we introduce a unified framework that incorporates all the existing solutions for the graph-based RAG methods [77]. The framework consists of four stages: ❶ *graph building*, ❷ *index construction*, ❸ *operator configuration*, and ❹ *retrieve relevant information and generate response*.

Specifically, given the large corpus $\mathcal{D}$, we first split it into multiple chunks C (line 1). We then sequentially execute operations in the following four stages: ❶ Build the graph $\mathcal{G}$ for input chunks C; ❷ Construct the index based on the graph $\mathcal{G}$ from the previous stage; ❸ Configure the retriever operators for subsequent retrieving stages, and ❹ For the input user question Q, retrieve relevant information from $\mathcal{G}$ using the selected operators and feed them along with the question Q into the LLM to generate the answer. Figure 1 emphasizes this end-to-end workflow and clarifies how information flows from raw corpus chunks to graph construction, indexing, retrieval-operator selection, and final response generation. It also separates the offline stages, where the graph and index are prepared, from the online stage, where a user query is mapped to retrieval operations and grounded evidence is passed to the LLM. Table 1 complements the figure by comparing how representative methods instantiate the main design choices in this workflow. In Table 1, Stage ❶ records the graph type built by each method, Stage ❷ records the graph components stored in the index, and Stage ❸ records both the retrieval primitive derived from the query

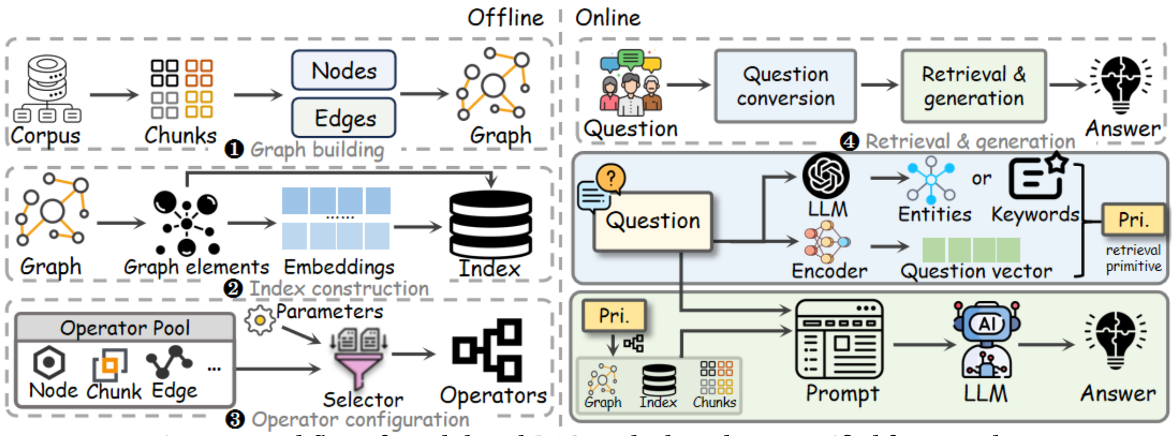

Figure 1: Workflow of graph-based RAG methods under our unified framework.

and the granularity of the retrieved evidence. The last two columns summarize whether each method supports specific fact-oriented QA, abstract QA, or both. Together, Figure 1 provides the process view, while Table 1 provides the method-by-method comparison view. Notice that the framework presented in Figure 1 has been fully implemented in our prototype system, which is publicly available on GitHub: https://github.com/JayLZhou/GraphRAG and has received more than 1,500 GitHub stars.

2.2.1 Graph Construction. In this part, we introduce how the graph is built from raw corpora, and discuss construction methods for different graph structures used in graph-based RAG methods, such as knowledge graphs [19, 29, 62], property graphs [2, 33], hypergraphs [35, 37], and tree-structured organizations [49, 61].

We also briefly discuss schema design and constraints for property graphs, including PG-Schema and PG-Keys [2, 3]. Note that property graphs provide a general modeling abstraction for many graph-based RAG methods, as their heterogeneous entities, relations, and textual attributes can be viewed as property graph elements. We further discuss graph repair or schema validation [5], and show how these techniques can improve graph quality for the later stages of indexing and retrieval.

2.2.2 Index Construction and Maintenance. In this part, we will discuss the indexing stage after graph construction [50, 59], which plays a central role in making graph-based RAG practical for real-world applications. Since graph-based RAG systems often need to retrieve relevant information from large and heterogeneous graph structures, effective indexing is essential for reducing online retrieval cost and improving response efficiency.

Specifically, we will introduce different types of indexes built over graph data, including indexes for entities, relations, paths, communities, and subgraphs [8, 12, 18, 31, 41, 59]. We will explain how these indexing strategies support different retrieval patterns, such as entity-centric retrieval, path-based reasoning, subgraph expansion, and hybrid retrieval over graph and text. We will also discuss the trade-offs among preprocessing cost, storage overhead, index granularity, maintenance complexity, and query latency, and show how different design choices are suitable for different scenarios.

In addition, we will discuss how indexes can be maintained when documents and graphs evolve, including incremental updates, partial re-indexing, and re-materialization [70]. This is particularly important in graph-based RAG settings where the underlying knowledge source may change frequently, and stale indexes may hurt both retrieval quality and answer reliability.

2.2.3 Graph Retrieval. In this part, we introduce the main retrieval strategies in graph-based RAG, which determine how a system identifies, expands, filters, and ranks graph-grounded information for a given user query. We will also explain how different graph-based RAG methods perform retrieval under different graph structures and task settings. Specifically, we will cover several representative retrieval strategies, including entity selection, k-hop path selection, and subgraph retrieval [8, 31, 40, 41, 51]. We will discuss how these strategies differ in retrieval granularity and how they support different reasoning needs, such as local fact lookup, multi-hop reasoning, abstract summarization, and evidence aggregation. We will also explain how different retrieval strategies balance effectiveness and efficiency. In addition, we will also highlight the close connection between retrieval design and index support, showing how online retrieval behavior depends on the underlying graph organization and indexing choices.

2.3 Extensions to Other Modalities (15 minutes)

In this part, we discuss how graph-based RAG can be extended beyond unstructured text to a broader range of data modalities, including tables, images, code, spatial data, and structured or semi-structured documents [61]. We will highlight that, in many real applications, knowledge is distributed across heterogeneous sources rather than plain text alone, which makes multimodal and multi-source graph-based RAG an important research direction.

Specifically, we will introduce representative extension scenarios, such as table-oriented RAG [9], spatial-data RAG [68], and graph-based RAG over semi-structured or enterprise data [54, 61, 65]. We will discuss how graphs can be used to organize cross-modal entities, relations, and contexts, so that evidence from different modalities can be connected and retrieved in a unified manner [49, 61]. In addition, we will introduce several important design questions for these extended settings, including multimodal graph construction, modular system interfaces, evidence serialization, cross-modal retrieval, and provenance tracking across different modalities.

2.4 Future Directions (15 minutes)

In this part, we summarize several important open problems and opportunities in graph-based RAG.

• **Dynamic knowledge sources.** Most existing graph-based RAG methods assume that the external corpus is static. Although EraRAG [70] has made an early attempt to support evolving corpora, it remains an important open problem to maintain graphs and indexes efficiently while preserving retrieval quality as the knowledge source changes over time.

• **Graph quality and construction cost.** The quality of the constructed graph has a major impact on retrieval effectiveness, while graph construction can be expensive. Developing cost-efficient yet high-quality graph construction methods remains a meaningful research direction.

• **Causal-driven GraphRAG.** Inspired by recent causal RAG studies such as CausalRAG and HugRAG [57, 58], future GraphRAG systems could incorporate cause-effect relations to improve retrieval faithfulness, reduce spurious associations, and support more interpretable reasoning [45].

• **Skill-oriented GraphRAG.** Future GraphRAG systems could move beyond factual evidence retrieval by retrieving and composing reusable skills [42, 76], such as tool-use patterns, coding procedures, and multi-step workflows. This would allow models to reuse procedural knowledge for complex tasks instead of relying only on static textual evidence.

• **System support.** Another important direction is to integrate graph-based RAG with graph database systems and support richer retrieval operators and more efficient system implementations.

Learning outcomes: After the tutorial, attendees should be able to: (1) The typical applications and key challenges of graph-based RAG. (2) Explain how graph construction choices affect downstream indexing, retrieval quality, and answer faithfulness. (3) Describe how graph indexes are built and maintained for interactive graph-based RAG workloads. (4) Choose among hop-based, ranking-based, and hybrid graph retrieval patterns for different query types. (5) Identify promising research directions for graph-based RAG systems.

3 TARGET AUDIENCE AND PRE-REQUISITES

The tutorial is aimed at academic and industrial attendees interested in data management, graph databases, knowledge graphs, graph mining, or LLM applications who want a structured, database-centric entry point to graph-based RAG. Newcomers will leave with a clear distinction between graph-based and vanilla RAG, a mental model of the end-to-end pipeline (construction, retrieval, reasoning, serving), and pointers to representative designs such as hierarchical summarization graphs and community-based indexing [11, 49, 59]. Experienced graph or database researchers will find a mapping from graph-based RAG operators to familiar query and indexing problems, a discussion of retrieval settings where community search and cohesive subgraphs matter [15], and a concise overview of representative open-source and commercial systems, including PostgreSQL [47] and Neo4j [43]. We also discuss how graph-based RAG has become a core component of modern graph-native agentic systems, such as LangGraph [27] and Chat2Graph [7], together with optimization questions in the VLDB tradition. Practitioners will also benefit from a feature-oriented view of representative open-source and commercial graph-based RAG systems when evaluating deployment choices. We assume a typical VLDB background: elementary databases, declarative querying intuition, and basic graph terminology (paths, neighborhoods, subgraphs). Prior exposure to LLMs is helpful but not required, and we keep model internals to the minimum needed for interfaces and failure modes. The material should also interest researchers in information retrieval and ML systems and practitioners building retrieval stacks over enterprise knowledge graphs or large document collections.

4 PRESENTERS

The four presenters have complementary expertise in graph data management, knowledge graphs, and graph-based RAG.

Angela Bonifati is a Distinguished Professor of Computer Science at Lyon 1 University and the CNRS LIRIS research laboratory, where she leads the Database Group, and is also an Adjunct Professor at the University of Waterloo. Her research focuses on data management, especially graph databases, knowledge graphs, and data integration, with applications to data science and artificial intelligence. She has co-authored numerous papers in top data management venues, including five Best Paper Awards. She is an ACM Fellow and a member of Academia Europaea, and has received an ERC Advanced Grant, the IEEE TCDE Impact Award, and the VLDB Women in Database Research Award. She is the current President of ACM SIGMOD.

Yixiang Fang is an Associate Professor at the School of Data Science, The Chinese University of Hong Kong, Shenzhen. His research interests mainly focus on the areas of data management, data mining, and artificial intelligence over big data. He has published extensively in top-tier conferences and journals, and his work was awarded the 2021 ACM SIGMOD Research Highlight Award. He has also served as PC members and reviewers for several top conferences and journals (e.g., PVLDB, ICDE, KDD, and VLDBJ).

Sibo Wang is an Associate Professor at The Chinese University of Hong Kong. His research interests mainly focus on big data processing and analytics. He has published extensively in top-tier conferences and journals. He has served as PC members for several top conferences (e.g., SIGMOD, PVLDB and ICDE) and reviewers for top journals (e.g., TKDE and VLDBJ).

Yingli Zhou is a postdoctoral researcher at Lyon 1 University and the CNRS LIRIS research laboratory. His research lies at the intersection of graph data management and large language model (LLM) systems. His work has been published in top-tier venues in data management and data mining, such as SIGMOD and PVLDB.

Note: This tutorial has not been offered elsewhere before.

ACKNOWLEDGMENTS

Angela Bonifati and Yingli Zhou were supported by the European Research Council grant GO-Y nr. 101199575. Yixiang Fang and Sibo Wang were supported by the 1+1+1 CUHK-CUHK(SZ)-GDSTC Joint Collaboration Fund under Grant 2025A0505000045.

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman,

Shyamal Anadkat, et al. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Renzo Angles, Angela Bonifati, David Bronsard, George Fletcher, Pasquale Liuzzo, Oliver Lorenz, and Hannes Voigt. 2023. PG-Schema: Schemas for property graphs. *Proceedings of the VLDB Endowment* 16, 12 (2023), 4439–4442.

[3] Renzo Angles, Angela Bonifati, Nadime Francis, and Hannes Voigt. 2021. PG-Keys: Keys for property graphs. In *Proceedings of the 2021 International Conference on Management of Data*. 2423–2426.

[4] Anthropic. 2026. Introducing Claude Sonnet 4.6. https://www.anthropic.com/news/claude-sonnet-4-6.

[5] Angela Bonifati, Martin Junghanns, Andrea Marchetti, Salvatore Orlando, Claudio Piro, Giorgio Re, and Maxime Tence. 2019. Schema validation and evolution for graph databases. In *Conceptual Modeling*. Springer, 448–456.

[6] Yukun Cao, Zengyi Gao, Zhiyang Li, Xike Xie, S. Kevin Zhou, and Jianliang Xu. 2025. LEGO-GraphRAG: Modularizing Graph-based Retrieval-Augmented Generation for Design Space Exploration. *Proc. VLDB Endow.* 18, 10 (2025), 3269–3283.

[7] Chat2Graph. 2025. About Chat2Graph: Graph Native Agentic System. https://chat2graph.vercel.app/chat2graph/en-us/introduction.

[8] Boyu Chen, Zirui Guo, Zidan Yang, Yuluo Chen, Junze Chen, Zhenghao Liu, Chuan Shi, and Cheng Yang. 2025. PathRAG: Pruning Graph-based Retrieval Augmented Generation with Relational Paths. *arXiv preprint arXiv:2502.14902* (2025).

[9] Si-An Chen, Lesly Miculicich, Julian M Eisenschlos, Zifeng Wang, Zilong Wang, Yanfei Chen, Yasuhisa Fujii, Hsuan-Tien Lin, Chen-Yu Lee, and Tomas Pfister. 2024. Tablerag: Million-token table understanding with language models. *Advances in Neural Information Processing Systems* 37 (2024), 74899–74921.

[10] Databricks. 2024. Building, Improving, and Deploying Knowledge Graph RAG Systems on Databricks. https://neo4j.com/docs/neo4j-graphrag-python/current/.

[11] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv preprint arXiv:2404.16130* (2024).

[12] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130* (2024).

[13] Ju Fan, Zihui Gu, Songyue Zhang, Yuxin Zhang, Zui Chen, Lei Cao, Guoliang Li, Samuel Madden, Xiaoyong Du, and Nan Tang. 2024. Combining small language models and large language models for zero-shot nl2sql. *Proceedings of the VLDB Endowment* 17, 11 (2024), 2750–2763.

[14] Meihao Fan, Xiaoyue Han, Ju Fan, Chengliang Chai, Nan Tang, Guoliang Li, and Xiaoyong Du. 2024. Cost-effective in-context learning for entity resolution: A design space exploration. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 3696–3709.

[15] Yixiang Fang, Xin Huang, Lu Qin, Ying Zhang, Wenjie Zhang, Reynold Cheng, and Xuemin Lin. 2020. A survey of community search over big graphs. *VLDB J.* 29, 1 (2020), 353–392.

[16] Yunfan Gao, Xiao Ma, Jiafeng Lin, Jiawei Zhao, and Bing Zhang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* (2023).

[17] Victor Giannankouris and Immanuel Trummer. 2024. {\lambda}-Tune: Harnessing Large Language Models for Automated Database System Tuning. *arXiv preprint arXiv:2411.03500* (2024).

[18] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. LightRAG: Simple and Fast Retrieval-Augmented Generation. *arXiv preprint arXiv:2410.05779* (2024).

[19] Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models. *Advances in Neural Information Processing Systems* 37 (2024).

[20] Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. In *Proceedings of the 42nd International Conference on Machine Learning*.

[21] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. *Advances in Neural Information Processing Systems* 37 (2024).

[22] Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2024. GRAG: Graph Retrieval-Augmented Generation. *arXiv preprint arXiv:2405.16506* (2024).

[23] Yucheng Hu and Yuxing Lu. 2024. Rag and rau: A survey on retrieval-augmented language model in natural language processing. *arXiv preprint arXiv:2404.19543* (2024).

[24] Yizheng Huang and Jimmy Huang. 2024. A Survey on Retrieval-Augmented Text Generation for Large Language Models. *arXiv preprint arXiv:2404.10981* (2024).

[25] Yiqian Huang, Shiqi Zhang, and Xiaokui Xiao. 2025. KET-RAG: A Cost-Efficient Multi-Granular Indexing Framework for Graph-RAG. *arXiv preprint arXiv:2502.09304* (2025).

[26] Yuxin Jin, Ying Zhang, Hanchen Wang, and Wenjie Zhang. 2026. Data-Centric Foundations of Agentic AI. Accepted tutorial at IEEE ICDE 2026, https://icde2026.github.io/tutorials.html.

[27] LangGraph. 2024. LangGraph. https://www.langchain.com/langgraph.

[28] Kwun Hang Lau, Fangyuan Zhang, Boyu Ruan, Yingli Zhou, Qintian Guo, Ruiyuan Zhang, and Xiaofang Zhou. 2026. Breaking the Static Graph: Context-Aware Traversal for Robust Retrieval-Augmented Generation. *arXiv preprint arXiv:2602.01965* (2026).

[29] Dawei Li, Shu Yang, Zhen Tan, Jae Young Baik, Sukwon Yun, Joseph Lee, Aaron Chacko, Bojian Hou, Duy Duong-Tran, Ying Ding, et al. 2024. DALK: Dynamic Co-Augmentation of LLMs and KG to Answer Alzheimer's Disease Questions with Scientific Literature. *arXiv preprint arXiv:2405.04819* (2024).

[30] Lan Li, Liri Fang, and Vetle I Torvik. 2024. AutoDCWorkflow: LLM-based Data Cleaning Workflow Auto-Generation and Benchmark. *arXiv preprint arXiv:2412.06724* (2024).

[31] Mufei Li, Siqi Miao, and Pan Li. 2025. Simple is Effective: The Roles of Graphs and Large Language Models in Knowledge-Graph-Based Retrieval-Augmented Generation. In *International Conference on Learning Representations*.

[32] Zhaodonghui Li, Haitao Yuan, Huiming Wang, Gao Cong, and Lidong Bing. 2024. LLM-R2: A Large Language Model Enhanced Rule-based Rewrite System for Boosting Query Efficiency. *arXiv preprint arXiv:2404.12872* (2024).

[33] Lei Liang, Mengshu Sun, Zhengke Gui, Zhongshu Zhu, Zhouyu Jiang, Ling Zhong, Yuan Qu, Peilong Zhao, Zhongpu Bo, Jin Yang, Huaidong Xiong, Lin Yuan, Jun Xu, Zaoyang Wang, Zhiqiang Zhang, Wen Zhang, Huajun Chen, Wenguang Chen, and Jun Zhou. 2024. KAG: Boosting LLMs in Professional Domains via Knowledge Augmented Generation. *arXiv preprint arXiv:2409.13731* (2024).

[34] Xujian Liang and Zhaoquan Gu. 2025. Fast Think-on-Graph: Wider, Deeper and Faster Reasoning of Large Language Model on Knowledge Graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 24558–24566.

[35] Wen-Sheng Lien, Yu-Kai Chan, Hao-Lung Hsiao, Bo-Kai Ruan, Meng-Fen Chiang, Chien-An Chen, Yi-Ren Yeh, and Hong-Han Shuai. 2026. HyperRAG: Reasoning N-ary Facts over Hypergraphs for Retrieval Augmented Generation. *arXiv preprint arXiv:2602.14470* (2026).

[36] Lei Liu, Xiaoyan Yang, Junchi Lei, Xiaoyang Liu, Yue Shen, Zhiqiang Zhang, Peng Wei, Jinjie Gu, Zhixuan Chu, Zhan Qin, et al. 2024. A Survey on Medical Large Language Models: Technology, Application, Trustworthiness, and Future Directions. *arXiv preprint arXiv:2406.03712* (2024).

[37] Haoran Luo, Haihong E, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin Kuang, Meina Song, Zhonghong Ou, and Luu Anh Tuan. 2025. HyperGraphRAG: Retrieval-Augmented Generation via Hypergraph-Structured Knowledge Representation. *arXiv preprint arXiv:2503.21322* (2025).

[38] Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. 2025. GFM-RAG: Graph Foundation Model for Retrieval Augmented Generation. *arXiv preprint arXiv:2502.01113* (2025).

[39] Yuyu Luo, Guoliang Li, Ju Fan, and Nan Tang. 2026. Data Agents: Levels, State of the Art, and Open Problems. Accepted tutorial at ACM SIGMOD 2026, https://2026.sigmod.org/sigmod_tutorials.shtml.

[40] Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. 2025. Think-on-Graph 2.0: Deep and Faithful Large Language Model Reasoning with Knowledge-guided Retrieval Augmented Generation. In *International Conference on Learning Representations*.

[41] Costas Mavromatis and George Karypis. 2024. GNN-RAG: Graph Neural Retrieval for Large Language Model Reasoning. *arXiv preprint arXiv:2405.20139* (2024).

[42] Xiangcheng Meng, Shu Wang, and Yixiang Fang. 2026. SkillRAE: Agent Skill-Based Context Compilation for Retrieval-Augmented Execution. *arXiv preprint arXiv:2605.10114* (2026).

[43] neo4j rag. 2024. Neo4j GraphRAG. https://neo4j.com/docs/neo4j-graphrag-python/current/.

[44] Yuqi Nie, Yaxuan Kong, Xiaowen Dong, John M Mulvey, H Vincent Poor, Qingsong Wen, and Stefan Zohren. 2024. A Survey of Large Language Models for Financial Applications: Progress, Prospects and Challenges. *arXiv preprint arXiv:2406.11903* (2024).

[45] Amedeo Pachera, Mattia Palmiotto, Angela Bonifati, and Andrea Mauri. 2025. What If: Causal Analysis with Graph Databases. *Proc. VLDB Endow.* 18, 11 (2025), 4009–4016.

[46] Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921* (2024).

[47] PostgreSQL. 2024. Azure Database for PostgreSQL Blog. https://techcommunity.microsoft.com/blog/adforpostgresql/introducing-the-graphrag-solution-for-azure-database-for-postgresql/4299871.

[48] Yichen Qian, Yongyi He, Rong Zhu, Jintao Huang, Zhijian Ma, Haibin Wang, Yaohua Wang, Xiuyu Sun, Defu Lian, Bolin Ding, et al. 2024. UniDM: A Unified

Framework for Data Manipulation with Large Language Models. *Proceedings of Machine Learning and Systems* 6 (2024), 465–482.

[49] Parth Sarthi, Sarah Abdullah, Adam Tunnell, Jack Huang, Shreyas Singh, Allen Zou, and Chris Newman. 2024. RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval. *arXiv preprint arXiv:2401.18059* (2024).

[50] Yaodong Su, Yixiang Fang, Yingli Zhou, Quanqing Xu, and Chuanhui Yang. 2025. Clue-RAG: Towards Accurate and Cost-Efficient Graph-based RAG via Multi-Partite Graph and Query-Driven Iterative Retrieval. *arXiv preprint arXiv:2507.08445* (2025).

[51] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In *International Conference on Learning Representations*.

[52] Wenwen Sun, Zhicheng Pan, Zirui Hu, Yu Liu, Chengcheng Yang, Rong Zhang, and Xuan Zhou. 2025. Rabbit: Retrieval-Augmented Generation Enables Better Automatic Database Knob Tuning. In *ICDE*. IEEE, 3807–3820.

[53] Zhaoyan Sun, Xuanhe Zhou, and Guoliang Li. 2024. R-Bot: An LLM-based Query Rewrite System. *arXiv preprint arXiv:2412.01661* (2024).

[54] Zirui Tang, Boyu Niu, Xuanhe Zhou, Boxiu Li, Wei Zhou, Jiannan Wang, Guoliang Li, Xinyi Zhang, and Fan Wu. 2025. St-raptor: Llm-powered semi-structured table question answering. *Proceedings of the ACM on Management of Data* 3, 6 (2025), 1–27.

[55] Hanchen Wang, Ying Zhang, and Wenjie Zhang. 2025. Machine Learning for Graph Data Management and Query Processing. *Proceedings of the VLDB Endowment* 18, 12 (2025), 5499–5503. doi:10.14778/3750601.3750702

[56] Jinqiang Wang, Huansheng Ning, Yi Peng, Qikai Wei, Daniel Tesfai, Wenwei Mao, Tao Zhu, and Runhe Huang. 2024. A survey on large language models from general purpose to medical applications: Datasets, methodologies, and evaluations. *arXiv preprint arXiv:2406.10303* (2024).

[57] Nengbo Wang, Xiaotian Han, Jagdip Singh, Jing Ma, and Vipin Chaudhary. 2025. CausalRAG: Integrating Causal Graphs into Retrieval-Augmented Generation. In *ACL (Findings) (Findings of ACL, Vol. ACL 2025)*. Association for Computational Linguistics, 22680–22693.

[58] Nengbo Wang, Tuo Liang, Vikash Singh, Chaoda Song, Van Yang, Yu Yin, Jing Ma, Jagdip Singh, and Vipin Chaudhary. 2026. HugRAG: Hierarchical Causal Knowledge Graph Design for RAG. *CoRR* abs/2602.05143 (2026).

[59] Shu Wang, Yixiang Fang, Yingli Zhou, Xilin Liu, and Yuchi Ma. 2026. Archrag: Attributed community-based hierarchical retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 15868–15876.

[60] Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S Yu, and Qingsong Wen. 2024. Large language models for education: A survey and outlook. *arXiv preprint arXiv:2403.18105* (2024).

[61] Shu Wang, Yingli Zhou, and Yixiang Fang. 2025. BookRAG: A Hierarchical Structure-aware Index-based Approach for Retrieval-Augmented Generation on Complex Documents. *arXiv preprint arXiv:2512.03413* (2025).

[62] Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. 2024. Knowledge graph prompting for multi-document question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 19206–19214.

[63] Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Menolascina, and Vicente Grau. 2024. Medical graph RAG: Towards safe medical large language model via graph retrieval-augmented generation. *arXiv preprint arXiv:2408.04187* (2024).

[64] Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, et al. 2024. Retrieval-augmented generation for natural language processing: A survey. *arXiv preprint arXiv:2407.13193* (2024).

[65] Bangrui Xu, Qihang Yao, Zirui Tang, Xuanhe Zhou, Yeye He, Shihan Yu, Qianqian Xu, Bin Wang, Guoliang Li, Conghui He, et al. 2026. MoDora: Tree-Based Semi-Structured Document Analysis System. *Proceedings of the ACM on Management of Data* 3, 6 (2026), 1–27.

[66] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Dayiheng Liu, Fan Zhou, Fei Huang, Haoran Wei, et al. 2025. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388* (2025).

[67] Chuanyue Yu, Kuo Zhao, Yuhan Li, Heng Chang, Mingjian Feng, Xiangzhe Jiang, Yufei Sun, Jia Li, Yuzhi Zhang, Jianxin Li, and Ziwei Zhang. 2025. GraphRAG-R1: Graph Retrieval-Augmented Generation with Process-Constrained Reinforcement Learning. *arXiv preprint arXiv:2507.23581* (2025).

[68] Dazhou Yu, Riyang Bao, Ruiyu Ning, Jinghong Peng, Gengchen Mai, and Liang Zhao. 2025. Spatial-rag: Spatial retrieval augmented generation for real-world geospatial reasoning questions. *arXiv preprint arXiv:2502.18470* (2025).

[69] Nan Zhang, Prafulla Kumar Choubey, Alexander Fabbri, Gabriel Bernadett-Shapiro, Rui Zhang, Prasenjit Mitra, Caiming Xiong, and Chien-Sheng Wu. 2025. SiReRAG: Indexing Similar and Related Information for Multihop Reasoning. In *International Conference on Learning Representations*.

[70] Yang Zhang, Runyuan He, Jiaming Liu, Chao Zhao, Yingli Zhou, and Yixiang Fang. 2025. EraRAG: Efficient Retrieval-Augmented Generation with Evolving Corpora. *arXiv preprint arXiv:2506.20963* (2025).

[71] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473* (2024).

[72] Yibo Zhao, Jiapeng Zhu, Ye Guo, Kangkang He, and Xiang Li. 2025. E^2GraphRAG: Streamlining Graph-based RAG for High Efficiency and Effectiveness. *arXiv preprint arXiv:2505.24226* (2025).

[73] Yanxin Zheng, Wensheng Gan, Zefeng Chen, Zhenlian Qi, Qian Liang, and Philip S Yu. 2024. Large language models for medicine: a survey. *International Journal of Machine Learning and Cybernetics* (2024), 1–26.

[74] Yihang Zheng, Bo Li, Zhenghao Lin, Yi Luo, Xuanhe Zhou, Chen Lin, Jinsong Su, Guoliang Li, and Shifu Li. 2024. Revolutionizing Database Q&A with Large Language Models: Comprehensive Benchmark and Evaluation. *arXiv preprint arXiv:2409.04475* (2024).

[75] Xuanhe Zhou, Guoliang Li, Zhaoyan Sun, Zhiyuan Liu, Weize Chen, Jianming Wu, Jiesi Liu, Ruohang Feng, and Guoyang Zeng. 2024. D-bot: Database diagnosis system using large language models. *Proceedings of the VLDB Endowment* 17, 10 (2024), 2514–2527.

[76] Yingli Zhou, Wang Shu, Yaodong Su, Wenchuan Du, Yixiang Fang, and Xuemin Lin. 2026. A comprehensive survey on agent skills: Taxonomy, techniques, and applications. *arXiv preprint arXiv:2605.07358* (2026).

[77] Yingli Zhou, Yaodong Su, Youran Sun, Shu Wang, Taotao Wang, Runyuan He, Yongwei Zhang, Sicong Liang, Xilin Liu, Yuchi Ma, and Yixiang Fang. 2025. In-depth Analysis of Graph-based RAG in a Unified Framework. *arXiv preprint arXiv:2503.04338* (2025).

[78] Yingli Zhou, Zixuan Wang, and Yixiang Fang. 2026. HAMMER: An Automatic RAG Tuning System via Hierarchical Memory-Guided Monte Carlo Tree Search. *Proceedings of the ACM on Management of Data* 4, 3 (SIGMOD (2026), 1–29.

[79] Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. 2025. LinearRAG: Linear Graph Retrieval Augmented Generation on Large-scale Corpora. *arXiv preprint arXiv:2510.10114* (2025).