Graph Retrieval-Augmented Generation for Large Language Models: A Survey

Tyler Thomas Procko, Omar Ochoa
Department of Electrical Engineering and Computer Science
Embry-Riddle Aeronautical University
Daytona Beach, United States of America
prockot@my.erau.edu, ochoao@erau.edu

***Abstract*—Large Language Models (LLMs) demonstrate general knowledge, but they suffer when specifically needed knowledge is not present in their training set. Two approaches to ameliorating this, without re-training, are 1) prompt engineering and 2) Retrieval-Augmented Generation (RAG). RAG is a form of prompt engineering, insofar as relevant lexical snippets retrieved from RAG corpora are vectorized and aggregated with prompts. However, RAG documents are often noisy, i.e., while relevant to a given prompt, they can contain much other information that obfuscates the desired snippet. If the purpose of pre-training a LLM on massive and general corpora is to engender a generally applicable model, RAG is not: it is a means of LLM optimization, and as such, RAG document selection must be precise, not general. For expert tasks, it is imperative that a RAG corpus be as noise-free as possible, in much the same way a good prompt should be free of irrelevant text. Knowledge Graphs (KGs) provide a concise means of representing domain knowledge free of noisy information. This paper surveys work incorporating KGs with LLM RAG, intending to equip scientists with a better understanding of this novel research area for future work.**

Keywords—LLM, GPT, fine-tuning, knowledge graphs, RAG

I. Introduction

Large Language Models (LLMs) are at the forefront of large-scale generative artificial intelligence constructs being employed in various fields [1]. In applications requiring deep learning techniques, the traditional method of training models uniquely fit for specific use cases has been replaced with the "pre-train, fine-tune and predict" paradigm, which sees the use of large, "off the shelf" models capable of accurate predictions over a large domain knowledge space [2, 3].

Despite the general ability of LLMs, they suffer insofar as they may lack up-to-date information, or niche information, relevant to a given prompt, because re-training a LLM is both time-consuming and labor-intensive. Appropriate prompt engineering, e.g., providing more context, can ameliorate these issues, but it places a burden on the prompter. Another approach is to augment a LLM prompt with a search of external resources for relevant lexical fragments, referred to as Retrieval-Augmented Generation (RAG) [4].

With relevant domain documents, RAG improves LLM accuracy; but RAG documents can be noisy, i.e., they may contain information not exactly relevant to a given prompt. Increased document noise results in decreased accuracy from LLMs employing RAG [5]. The ideal RAG corpus is as specific to a given problem (prompting) set as possible.

Knowledge Graphs (KGs) are a form of knowledge representation comprised of nodes and edges, or entities and the relations between them. KGs may be predicated upon a rigorous ontology, e.g., through the Web Ontology Language (OWL), or grown rather organically with simple lexicons. KGs are commonly created for domain-specific use cases, providing concise and understandable sources of subject matter expert knowledge [6]. KGs may provide non-noisy RAG [7, 8, 9].

Pan et al. present a roadmap for the unification of KGs and LLMs, identifying three archetypes of interaction: KG-enhanced LLMs, LLM-augmented KGs and Synergized LLMs + KGs [10]. Using KGs for RAG is a practice in KG-enhanced LLMs. This research area is relatively novel and no aggregate of its works is extant. The present paper fills this knowledge void.

II. Background

LLMs are generally applicable to most application areas, but require optimization for specific tasks, or when particular constraints need to be met. There are three primary means of optimizing a LLM: fine-tuning, RAG and prompt engineering. These techniques aim to increase model steerability and can be used in tandem for greater effect [11] (see Fig 1).

- Fine-tuning: model trained on task-specific dataset or rule set addressing, e.g., safety, fairness, etc. [12, 13, 14]
- Retrieval-Augmented Generation (RAG): prompt augmented with relevant snippets from external documents [4]
- Prompt engineering: steering a model with task-specific instructions, e.g., Chain-of-Thought (CoT) prompting [15], Reasoning + Action (ReAct) [16], etc.

There are various named prompt engineering techniques [17]. Every prompt engineering technique, at the most fundamental level, amounts to providing a LLM with more context: be it literally, as with few-shot prompting, or through self-recorded logic, e.g., as with CoT prompting. RAG is itself a prompt engineering technique, as snippets induced through RAG are combined with a given input prompt to form a vector; but RAG is distinguished from prompt engineering insofar as it explicitly depends on a search of documents external to the prompt. All modes of information have been researched as RAG input: text, image, video, audio, code and structured knowledge, e.g., databases and graphs [18, 19, 20].

The underpinning of prompt engineering and RAG is the providence of relevant contextual information for a LLM to utilize. Increased document noise significantly worsens RAG efficacy [5], so structured, concise knowledge forms, e.g., knowledge graphs, are posited to be useful as RAG input [7, 8].

A. Knowledge Graphs and Textual Graphs

Traditional KG embedding approaches, which map KG entities and relations into a low-dimensional vector space for various NLP tasks, suffer from data sparsity because of the limited context of KG triples. To address this, some have incorporated

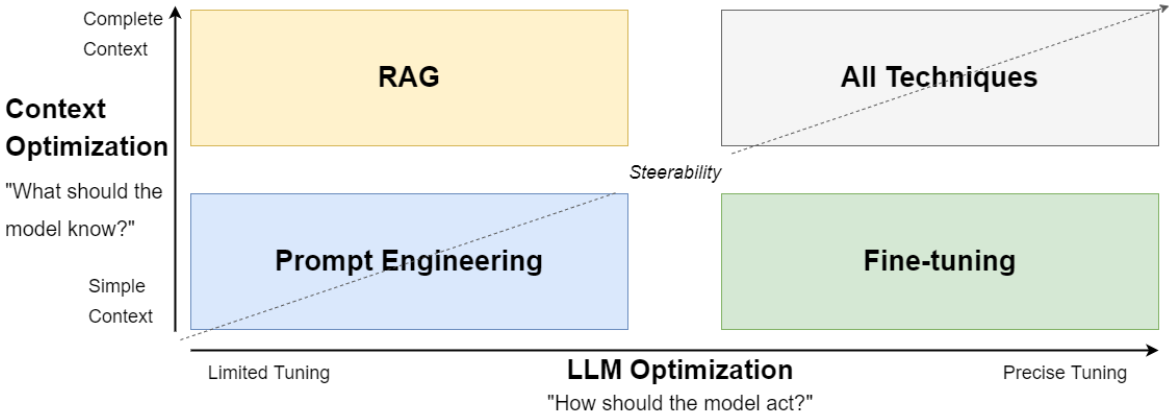

Fig. 1 LLM optimization techniques and their relationships to model steerability (adapted from [11]*).*

textual entity descriptions for richer context [21]. Such a construct is referred to as a text, or textual, graph. A textual graph is a graph whose nodes and edges are attended with textual attributes. Where the focus of a KG is on structured relationships, with text providing only supplementary context, a textual graph integrates text and structure as equally important context.

These two graph forms are distinguished here because they are similar but not precisely the same. Textual graphs typically have an underlying structure, so KGs are central to the discussion. In the present paper, both KGs and textual graphs are considered within the context of Graph RAG, but the term KG is used in reference to both.

III. Graph RAG

Much of the literature in the Graph RAG research landscape pertains to problems of Question-Answering (QA), which was traditionally addressed by KGs interfaced with intention and entity recognition algorithms. Knowledge graph question answering (KGQA) is useful for factual QA but suffers on open-ended questions. The combination of LLMs with KGQA frameworks is a commonly researched application of RAG, although the seminal RAG method is not always explicitly cited [22, 23].

While RAG is useful for providing additional context to a prompted LLM through small, relevant snippets, it suffers insofar as it does not consider broader contextual semantics, e.g., correlations among documents within corpora [24, 25]. Graph RAG approaches aim to address this limitation of naïve RAG by incorporating broader semantics across documents within corpora, which is made possible through searches over KGs or sub-graphs generated therefrom. At the core of Graph RAG research is the problem of relevant sub-graph identification from larger KGs, which is computationally infeasible (NP-hard) if done exhaustively [24].

He et al. present G-Retriever, a QA framework that enables users to "chat with their graph", useful for common sense reasoning, visual scene understanding and general knowledge graph reasoning [26]. Where extant RAG approaches are designed for simpler information, e.g., text, G-Retriever includes a new RAG approach applicable to general textual graphs. Their sub-graph construction solution is the Prize-Collecting Steiner Tree algorithm. They also present the GraphQA benchmark for evaluation, demonstrating that G-Retriever is scalable to larger graphs, applicable to tasks from several domains and lessens LLM hallucination.

Edge et al. present a Graph RAG approach to global question answering, framing the problem as one of query-focused summarization (QFS) [25]. They create a KG using an LLM in a series of steps, culminating in LLM-generated summaries of graph communities, enabling QFS of large corpora for better global QA.

A paper published in association with the business social media platform, LinkedIn, reports on the efficacy of a KG-enabled RAG system for customer service QA [27]. Traditional RAG-based LLM customer service approaches suffer from retrieval inaccuracy because tickets are treated as plain text, resulting in a loss of structure. The authors construct a KG from historical customer service issue tickets; this KG is then parsed into sub-graphs for RAG input.

Sanmartín presents the knowledge graph retrieval-augmented generation (KG-RAG) framework to perform KGQA [28]. Operating under the analogy of the "extended mind", where humans augment their cognitive ability with outside resources, e.g., books, he incorporates the novel Chain of Explorations (CoE) algorithm for Graph RAG, which enables an LLM to explore KG nodes and relationships for better QA.

Hu et al. introduce the Graph Retrieval-Augmented Generation (GRAG) computational framework [24]. Their GRAG approach comprises four stages: (1) generating embeddings of k-hop ego-graphs from a main textual graph, (2) retrieving relevant textual subgraphs, (3) applying soft pruning to reduce irrelevant entities, and (4) utilizing graph neural networks (GNNs) and LLMs to generate responses with both hard and soft prompts. GRAG surpassed G-Retriever in accuracy. The authors also discuss a notable specialization of RAG, in the dichotomy of hard and soft prompting [24].

- Hard Prompts: Explicit, structured text inputs given to a LLM to guide its generation
- Soft Prompts: Implicit embeddings in the LLM's input vector space, which the model uses to guide its generation without explicit instructions

Xu et al. present a QA system for the very specific domain of the Chinese silk weaving craft, Nanjing Yunjin [29]. Their system addresses the limitations of an extant KG-based QA system, namely, the need for constant updates to the KG and the

TABLE I. Graph RAG papers surveyed.

Paper	KG(s) Used	LLM(s) Evaluated	Dataset	Evaluation Technique	Domain
G-Retriever [25]	Unspecified	LLaMA2	GraphQA (ExplaGraphs, SceneGraphs, WebQSP)	Performance, Efficiency	Any
KG-RAG [27]	Generated*	GPT-4	Complex WebQuestions	EM, F_1, Accuracy, Hallucination	Any
GRAG [23]	Unspecified	LLaMA2	ExplaGraphs, Web QuestionsSP	F_1, Hit@1, Recall; Accuracy	Any
TRACE [8]	Generated*	LLaMA3	HotPotQA, 2WikiMultiHopQA, MuSiQue	EM, F_1	Any
Triple-Aware Reasoning [7]	ConceptNet	GPT-3.5, GPT-4	CommonsenseQA, OpenBookQA	Accuracy	Any
RAKG [9]	ConceptNet	Unspecified	CommonsenseQA, OpenBookQA	Accuracy	Any
GNN-RAG [31]	Freebase	LLaMA2	WebQuestionsSP, ComplexWebQuestion	Hit, Hit@1, F_1	Any
LPKG [32]	Wikidata15k	GPT-3.5, LLaMA3, CodeQwen1.5	HotPotQA, 2WikiMultihopQA, Bamboogle, MuSiQue, CLQA-Wiki	EM	Any
[24]	Generated*	GPT-4	Podcast transcripts and MultiHop-RAG	Head-to-head LLM evaluation	Any
[28]	Generated*	LLaMA2	100 domain questions	Accuracy	Anthropology
[26]	Generated*	GPT-4	"Golden" dataset of queries, tickets and solutions	Comparison to "golden" baseline using BLEU, METEOR, ROUGE	Customer Service
[29]	Generated*	GPT-4	Small task set	Comparison to baseline using human experts; correctness, usability, relevance, time; context recall, context precision	Manufacturing
[30]	Generated*	MechGPT	Small evaluation set curated for the article	Various qualitative techniques	Materials Science

limited scope of predefined entities and relations, by cascading user queries first through a typical KG QA pipeline, replete with intention and entity recognition, which may return a response; if the KG QA pipeline is unable to return a response, the problem is shifted to a RAG QA pipeline, which will return a response from a LLM augmented with Nanjing Yunjin documents.

Bahr et al. present a KG-enabled RAG system to be used in a QA chatbot for failure mode and effects analysis (FMEA) in product manufacturing [30]. An ontology of FMEA observations is presented, which is used to serialize KG triples. Vector embeddings are then created from the FMEA KG and incorporated into LLM RAG.

Fang, Meng and Macdonald present TRACE, which employs a KG generator to create a KG from retrieved documents, from which reasoning chains are made, enhancing the multi-hop reasoning abilities of RAG-enabled LLMs [8]. TRACE is similar in approach to the CoT prompting technique but demonstrates an improvement because KG triples are grounded in RAG documents. The authors note that noise is an issue of RAG. Also citing the issue of noisy RAG, Zhang and Shafiq discuss Triple-Aware Reasoning, which combines KGs and RAG for QA [7]. Their system incorporates a three-layer filtering mechanism to ensure that triples from the ConceptNet KG selected for RAG do not introduce noise into prompts: first during retrieval, again according to specific relationships and lastly with a designed prompt to filter incorporated triples again.

Sha et al. present the retrieval-augmented knowledge graph (RAKG) model [9]. RAKG operates by first extracting the most relevant sub-graphs using a density matrix, then fusing embedded questions and KG entities using a graph convolutional network and LLM.

Buehler provides an evaluation of MechGPT, a LLM trained on materials domain knowledge. Through a mixed-methods evaluation of MechGPT, he ultimately asserts that a fine-tuned LLM in combination with KG-enabled RAG results in better accuracy and scientific replicability [31].

Mavromatis and Karypis present GNN-RAG [32]. A graph neural network first reasons over a KG subgraph to retrieve answer candidates for questions. Then, the shortest paths in the KG that link questions to answers are extracted to reasoning paths. These paths are serialized in text and used as RAG input.

Wang et al. present a framework for LLM planning derived from KGs called Learning to Plan from Knowledge Graphs (LPKG) [33]. The LPKG approach begins by constructing planning data from KGs; such data is then used to fine-tune LLMs. They compare their method against direct prompting, CoT prompting, naïve RAG and ReAct prompting, among other baselines.

IV. Discussion

It is apparent, in synthesizing the literature, that there are two main perspectives within Graph RAG research:

1. Using extant KGs for RAG
2. Generating KGs from document corpora for RAG

The former perspective sees the leveraging of pre-existing KGs, e.g., ConceptNet, as RAG input; the latter typically involves the creation of task-specific KGs, either with traditional relation extraction approaches or with an LLM, to be used as RAG input. In the surveyed papers, both perspectives demonstrate efficacy, but the LLM-based generation of task-specific KGs is more prevalent and engenders greater adaptability to a wider array of application areas.

Evaluations of the surveyed Graph RAG approaches vary, but typically involve an open QA dataset and some typical metric, e.g., accuracy and F_1. The WebQuestionsSP, Complex WebQuestions, HotPotQA, CommonsenseQA and OpenBookQA datasets were commonly used to evaluate the surveyed Graph RAG approaches. Most of the approaches surveyed were agnostic to any particular area of interest, with only four specifying distinct focus areas: anthropology, customer service, manufacturing and materials science.

A commonly cited reason for incorporating RAG into LLM workflows is to increase LLM accuracy while reducing LLM hallucination. The underlying structure of KGs used for RAG is beneficial because the identification of relevant sub-graphs directly addresses the hallucination issue. Others have framed

RAG as a hierarchical document search problem, allowing for more efficient retrieval through global document organization [34].

Regarding limitations, the present paper is restricted in terms of replicability, because the survey was not conducted in a systematic fashion, to the extent that doing so is even possible [35]. Moreover, given the extreme novelty of the Graph RAG research area, and the consistent publication stream, it is not feasible to address all applicable research. At most, the present paper provides a concise "snapshot" of current research into Graph RAG.

V. Conclusion

RAG is useful to the LLM prompter, and LLM-enabled applications or enterprises, as it mitigates difficulty in explicit prompting work by shifting the burden of context providence to search algorithms run over large corpora. The selection of a RAG corpus appropriately relevant to a given task is important to the reliable fulfillment of the task. Furthermore, the level of noise, or irrelevant information, within a RAG corpus is inversely proportional to the level of RAG usefulness. So, a cogent RAG corpus is beneficial. Research indicates that KGs provide a basis for less noisy RAG in their cogent structuring of knowledge. The present paper provides researchers with a synthesis of Graph RAG approaches and applications. This area of work is promising insofar as it is a means of effectively reducing RAG noise which in turn reduces LLM response uncertainty.

VI. References

[1] T. T. Procko, T. Elvira and O. Ochoa, "Dawn of the Dialogue: AI's Leap from Lab to Living Room," *Frontiers in Artificial Intelligence,* vol. 7, 2024.

[2] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys,* vol. 55, no. 9, pp. 1-35, 2023.

[3] J. Zhang, H. Jiaxing, S. Jin and S. Lu, "Vision-language models for vision tasks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2024.

[4] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Advances in Neural Information Processing Systems,* vol. 33, pp. 9459-9474, 2020.

[5] J. Chen, H. Lin, X. Han and L. Sun, "Benchmarking large language models in retrieval-augmented generation," *Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 38, no. 16, pp. 17754-17762, 2024.

[6] B. Abu-Salih, "Domain-specific knowledge graphs: A survey," *Journal of Network and Computer Applications,* vol. 185, 2021.

[7] H. Zhang and M. Omair Shafiq, "Triple-Aware Reasoning: A Retrieval-Augmented Generation Approach for Enhancing Question-Answering Tasks with Knowledge Graphs and Large Language Models," *The 37th Canadian Conference on Artificial Intelligence,* 2024.

[8] J. Fang, Z. Meng and C. Macdonald, "TRACE the Evidence: Constructing Knowledge-Grounded Reasoning Chains for Retrieval-Augmented Generation," *arXiv preprint arXiv:2406.11460,* 2024.

[9] Y. Sha, Y. Feng, M. He, S. Liu and Y. Ji, "Retrieval-Augmented Knowledge Graph Reasoning for Commonsense Question Answering," *Mathematics* , vol. 11, no. 15, 2023.

[10] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang and X. Wu, "Unifying large language models and knowledge graphs: A roadmap," *IEEE Transactions on Knowledge and Data Engineering,* 2024.

[11] "Optimizing LLMs for accuracy," OpenAI, [Online]. Available: https://platform.openai.com/docs/guides/optimizing-llm-accuracy. [Accessed June 2024].

[12] O. Honovich, T. Scialom, O. Levy and T. Schick, "Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor," *arXiv preprint arXiv:2212.09689,* 2022.

[13] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in Neural Information Processing Systems,* vol. 36, 2024.

[14] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens and A. Askell, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems,* vol. 35, 2022.

[15] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems,* vol. 35, pp. 24824-24837, 2022.

[16] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan and Y. Cao, "React: Synergizing reasoning and acting in language models," *arXiv preprint arXiv:2210.03629,* 2022.

[17] P. Sahoo, A. K. Sing, S. Saha, V. Jain, S. Mondal and A. Chadha, "A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications," *arXiv preprint arXiv:2402.07927,* 2024.

[18] R. Zhao, H. Chen, W. Wang, F. Jiao, X. Long Do, C. Qin, B. Ding, X. Guo, M. Li, X. Li and S. Joty, "Retrieving multimodal information for augmented generation: A survey," *arXiv preprint arXiv:2303.10868,* 2023.

[19] Z. Hu, A. Iscen, C. Sun, Z. Wang, K.-W. Chang, Y. Sun, C. Schmid, D. A. Ross and A. Fathi, "Reveal: Retrieval-augmented visual-language pre-training with multi-source multimodal knowledge memory," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* pp. 23369-23379, 2023.

[20] W. Yu, "Retrieval-augmented generation across heterogeneous knowledge," *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies: student research workshop,* pp. 52-58, 2022.

[21] L. Hu, M. Zhang, S. Li, J. Shi, C. Shi, C. Yang and Z. Liu, "Text-graph enhanced knowledge graph representation learning," *Frontiers in Artificial Intelligence,* vol. 4, 2021.

[22] Y. Wu, N. Hu, G. Qi, S. Bi, J. Ren, A. Xie and W. Song, "Retrieve-rewrite-answer: A kg-to-text enhanced LLMs framework for knowledge graph question answering," *arXiv preprint arXiv:2309.11206,* 2023.

[23] P. Sen, S. Mavadia and A. Saffari, "Knowledge graph-augmented language models for complex question answering," *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE),* pp. 1-8, 2023.

[24] Y. Hu, Z. Lei, Z. Zhang, B. Pan, C. Ling and L. Zhao, "GRAG: Graph Retrieval-Augmented Generation," *arXiv preprint arXiv:2405.16506,* 2024.

[25] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt and J. Larson, "From local to global: A graph rag approach to query-focused summarization," *arXiv preprint arXiv:2404.16130,* 2024.

[26] X. He, Y. Tian, Y. Sun, N. V. Chawla, T. Laurent, Y. LeCun, X. Bresson and B. Hooi, "G-retriever: Retrieval-augmented generation for textual graph understanding and question answering," *arXiv preprint arXiv:2402.07630,* 2024.

[27] Z. Xu, M. Jerome Cruz, M. Guevara, T. Wang, M. Deshpande, X. Wang and Z. Li, "Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering," *arXiv preprint arXiv:2404.17723,* 2024.

[28] D. Sanmartín, "KG-RAG: Bridging the Gap Between Knowledge and Creativity," *arXiv preprint arXiv:2405.12035,* 2024.

[29] L. Xu, L. Lu, M. Liu, C. Song and L. Wu, "Nanjing Yunjin intelligent question-answering system based on knowledge graphs and retrieval augmented generation technology," *Heritage Science,* vol. 12, no. 1, 2024.

[30] L. Bahr, C. Wehner, J. Wewerka, J. Bittencourt, U. Schmid and R. Daub, "Knowledge Graph Enhanced Retrieval-Augmented Generation for Failure Mode and Effects Analysis," *arXiv preprint arXiv:2406.18114,* 2024.

[31] M. J. Buehler, "Generative retrieval-augmented ontologic graph and multiagent strategies for interpretive large language model-based materials design," *ACS Engineering Au,* vol. 4, no. 2, pp. 241-277, 2024.

[32] C. Mavromatis and G. Karypis, "GNN-RAG: Graph Neural Retrieval for Large Language Model Reasoning," *arXiv preprint arXiv:2405.20139,* 2024.

[33] J. Wang, M. Chen, B. Hu, D. Yang, Z. Liu, Y. Shen, P. Wei, Z. Zhang, J. Gu, J. Zhou, J. Z. Pan, W. Zhang and H. Chen, "Learning to Plan for Retrieval-Augmented Large Language Models from Knowledge Graphs," *arXiv preprint arXiv:2406.14282,* 2024.

[34] K. Goel and M. Chandak, "HIRO: Hierarchical Information Retrieval Optimization," *arXiv preprint arXiv:2406.09979,* 2024.

[35] S. K. Boell and D. Cecez-Kecmanovic, "On being 'systematic' in literature reviews," *Formulating Research Methods for Information Systems,* pp. 48-78, 2016.