The Structure-Content Trade-off in Knowledge Graph Retrieval: A Diagnostic Study of Question Decomposition

Valentin Six[1,2], **Gaël de Chalendar**[1], **Evan Dufraisse**[1]
[1] Université Paris-Saclay, CEA, List, Palaiseau, France
[2] Georgia Institute of Technology, Atlanta, United States
vsix3@gatech.edu, gael.de-chalendar@cea.fr, evan.dufraisse@mbzuai.ac.ae

Abstract

Large Language Models are increasingly combined with knowledge graphs to support multi-hop factual reasoning. A classical strategy for handling complex questions in such settings is question decomposition, where a question is broken into simpler subquestions to guide retrieval. While decomposition can improve relevance, its impact on the structure and connectivity of the retrieved information, as well as the implications for downstream reasoning, remain unclear. In this work, we present a diagnostic study of the effects of question decomposition on knowledge graph retrieval. We use a simple parametric interpolation between retrieval guided by the original question and its subquestions, allowing us to vary retrieval focus in a controlled manner. By softly anchoring subquestion-level retrieval to the original question, we allow structural properties of the retrieved subgraph to change naturally, without post-hoc enforcement of connectivity. Across different multi-hop QA benchmarks, we observe a consistent structure-content trade-off: subquestion-focused retrieval improves content precision but fragments the retrieved graph, whereas question-focused retrieval preserves structural coherence at the cost of relevance. Downstream QA performance peaks at intermediate settings, where sufficient connectivity emerges while maintaining high relevance. These results highlight the importance of jointly considering content and structure when designing retrieval strategies for reasoning over structured knowledge.

Keywords: Retrieval Augmented Generation, Knowledge Graph Question Answering, Question Decomposition

1. Introduction

Large Language Models (LLMs) have shown impressive capabilities across many natural language tasks, including question answering (Kamalloo et al., 2023), summarization (Liu et al., 2024), and machine translation (Zhang et al., 2023). However, their performance remains brittle when reasoning over multiple facts and maintaining factual consistency (Ji et al., 2023; Yang et al., 2024; Huang et al., 2025). Knowledge Graph Question Answering (KGQA) systems address these limitations by grounding generation in structured representations of entities and relations (Yasunaga et al., 2021; Opsahl, 2024). In such systems, the quality of the retrieved subgraph often determines whether an LLM can successfully integrate evidence and produce a correct answer.

Retrieving an appropriate subgraph for a complex question, however, is far from straightforward (Kotiranta et al., 2022; Peng et al., 2023). Unlike unstructured retrieval, knowledge graph retrieval must balance coverage, relevance, and coherence under strict context constraints (Zou, 2020). Retrieving too little information risks missing critical entities or relations, while retrieving too much introduces noise that can overwhelm downstream reasoning (Li et al., 2024a; Dong et al., 2025). As a result, many KGQA systems rely on heuristics to narrow retrieval to question-specific regions of the graph, often prioritizing relevance signals derived from the input question.

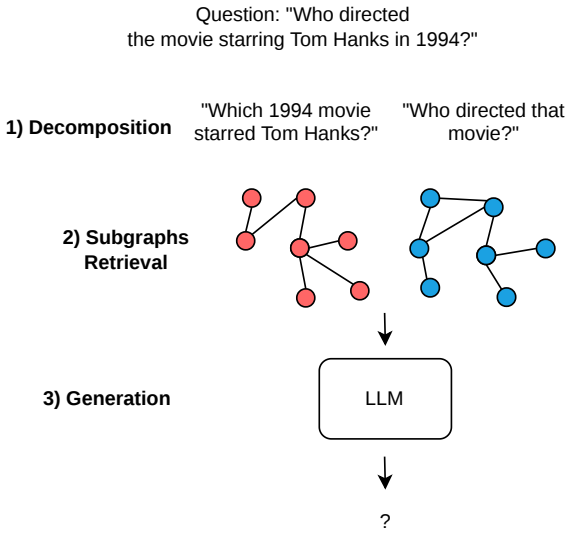

Figure 1: Illustration of the KGQA setting studied in this work. A complex question is decomposed into subquestions, each guiding retrieval of a relevant subgraph. While the retrieved subgraphs are individually informative, they may be structurally disconnected, making evidence integration challenging for downstream question answering.

A widely used strategy for handling complex questions is question decomposition, which breaks

Proceedings of the Knowledge Graphs and Large Language Models Workshop (KG-LLM) @ LREC26, pages 166–175
May 16, 2026.

a complex question into simpler subquestions that can be handled more locally (Perez et al., 2020; Fu et al., 2021). Decomposition is appealing because it can improve relevance, reduce semantic ambiguity, and align retrieval more closely with individual reasoning steps. Simultaneously, effective decomposition introduces strong assumptions, the most notable one being that retrieving locally relevant information for each step is sufficient for global reasoning. In practice, this assumption does not always hold: decomposed questions may retrieve highly specific but weakly related pieces of information or yield evidence that is difficult to reconcile into a coherent reasoning chain (Li et al., 2024b). Despite its conceptual simplicity and frequent use, the conditions under which decomposition helps or harms graph-based reasoning remain poorly understood. An illustration of the problem is shown in Figure 1.

One largely overlooked aspect of this issue is the structure of the retrieved subgraph. Knowledge graphs are not mere collections of relevant facts: their utility for reasoning depends on how retrieved entities are connected (Ding et al., 2024). Connectivity determines whether relations can be composed, whether intermediate entities can serve as bridges, and whether evidence can be integrated into a consistent explanation. However, many retrieval pipelines treat structure as secondary to relevance, evaluating success primarily by the presence of correct entities rather than by how those entities are organized. As a result, retrieval decisions that appear beneficial from a relevance perspective may silently degrade the structural coherence needed for multi-hop reasoning.

When retrieved subgraphs lack connectivity, a common response is to repair structure post hoc, for example, by adding linking nodes or pruning disconnected components (Shin and Lee, 2020). While such interventions can enforce a single connected graph (Wu et al., 2023), they also alter the retrieved evidence itself and introduce potentially irrelevant nodes and edges or exclude relevant information. From an analytical standpoint, this makes it difficult to disentangle whether downstream performance changes arise from improved evidence selection or from graph manipulation. Consequently, post-hoc structural enforcement can obscure the mechanisms by which retrieval strategies influence reasoning, particularly in the presence of question decomposition.

These challenges suggest that the limitations of question decomposition in knowledge graph retrieval are not merely a matter of algorithmic optimization but of understanding how retrieval choices reshape both what information is retrieved and how it is structured. Benchmark-driven evaluations, which collapse retrieval and reasoning effects into a single accuracy score, offer limited insight into this interaction. Instead, a diagnostic perspective is needed: one that isolates retrieval focus as a controllable variable and examines its consequences for subgraph relevance and connectivity independently of downstream modeling choices.

In this work, we adopt such a diagnostic approach to study question decomposition in knowledge graph retrieval [1]. We vary the relative influence of the original question and its subquestions during retrieval, allowing structural properties of the retrieved subgraph to emerge naturally rather than being imposed through post-hoc repair. Through experiments on two multi-hop QA benchmarks, CWQ (Talmor et al., 2018) and WebQSP (Yih et al., 2016), we observe a consistent structure–content trade-off: emphasizing subquestions improves content relevance but tends to fragment the retrieved graph, while emphasizing the original question preserves connectivity at the cost of precision. Downstream QA performance peaks at intermediate settings, where relevance and connectivity are jointly balanced. These findings highlight that effective reasoning over structured knowledge depends not only on retrieving relevant information, but on retrieving it in a form that preserves usable structure.

2. Related Work

Recent work in Knowledge Graph Question Answering (KGQA) has emphasized the role of retrieval in enabling effective reasoning with large language models. Beyond identifying relevant entities, retrieval in KGQA implicitly induces a graph structure that constrains how evidence can be combined. Several approaches have therefore introduced structural constraints directly into the retrieval process. Notably, G-Retriever (He et al., 2024) explicitly enforces connectivity for retrieval by extracting a single connected component, balancing relevance and compactness. This work highlights that structural properties of retrieved subgraphs can significantly influence downstream reasoning behavior.

A related line of work studies the trade-off between relevance, coverage, and graph size in knowledge graph retrieval. Prior analyses have shown that increasing node or edge coverage can improve recall but often introduces irrelevant information, while aggressively pruning the graph risks omitting critical evidence (Dai et al., 2025). These findings highlight that retrieval quality cannot be assessed solely by relevance metrics, but must also account for how retrieved information is organized and constrained.

[1] Corresponding code: https://github.com/cea-list-lasti/kg-retrieval-tradeoff

In parallel, question or query decomposition has emerged as a common strategy for handling complex, multi-hop questions in KGQA and retrieval-augmented generation systems (Chan et al., 2024; Li et al., 2024b). By decomposing a question into simpler subquestions, these approaches aim to improve retrieval precision and modularize reasoning. However, existing work largely evaluates decomposition through end-task performance, treating it as a heuristic to be optimized rather than as a variable whose effects can be systematically analyzed. As a result, little is known about how decomposition reshapes the content and structure of retrieved subgraphs, or how these changes impact the ability of language models to integrate evidence.

Our work connects these different threads by adopting a diagnostic perspective on question decomposition in knowledge graph retrieval. Rather than proposing a new retrieval algorithm, we study how varying the influence of the original question and its subquestions affects both the relevance and connectivity of retrieved subgraphs. This analysis reveals a structure–content trade-off induced by decomposition and clarifies how retrieval structure mediates downstream reasoning performance in KGQA.

3. Retrieval Setup for Studying Question Decomposition

3.1. Question Decomposition

Given an input question Q, we generate a set of atomic subquestions $\{q_1, \ldots, q_n\}$ using a large language model under explicit decomposition constraints. For example, the question *'Who directed the film starring Tom Hanks in 1994?'* decomposes into subquestions about Tom Hanks's filmography and film directors. Subquestions are required to be logically ordered, answerable from the knowledge graph, and to isolate a single semantic aspect of the reasoning process. The prompt enforces that subquestions collectively cover the information needed to answer Q, while avoiding redundancy and minimizing overlap between subquestions. At inference time, subquestions are answered sequentially, and the answer to each subquestion is appended to the subsequent subquestion to provide additional context for retrieval. These constraints are designed to standardize decomposition quality and ensure that differences observed downstream can be attributed to retrieval focus rather than unconstrained or inconsistent decomposition. The full prompt templates used for decomposition are provided in Section 4.

3.2. Parametric Retrieval Focus

To study the role of both the initial question Q and the corresponding subquestions $\{q_1, \ldots, q_n\}$, we introduce a new similarity function that allows us to control the influence of both sides:

$$\begin{aligned}\mathrm{sim}_\alpha(z, z_q, z_Q) = (1-\alpha)\ \cos(z, z_Q) \\ +\ \alpha\ \cos(z, z_q)\end{aligned} \tag{1}$$

We encode each subquestion q (augmented with the answers to preceding subquestions) as z_q, and the initial question Q as z_Q. Each node and edge of the graph is characterized by their textual entity label, which is encoded as z. The coefficient $\alpha \in [0, 1]$ controls retrieval focus: $\alpha = 0$ emphasizes the initial question Q, and $\alpha = 1$ the subquestions $\{q_1, \ldots, q_n\}$.

3.3. Prize Assignment

For each subquestion q, we define a prize assignment $\mathcal{P}_q$ over nodes and edges by selecting the top-k_n nodes and top-k_e edges according to the similarity score:

$$\mathcal{P}_q = \text{top-k}[\,\mathrm{sim}_\alpha(z, z_q, z_Q)\,] \tag{2}$$

We then extract a compact connected subgraph using the Prize-Collecting Steiner Tree (PCST) objective (Bienstock et al., 1993), as used in (He et al., 2024). PCST selects a connected set of nodes and edges that maximizes the total collected prize while paying edge costs, allowing it to retain relevant items while introducing only a few intermediate (Steiner) nodes when needed for connectivity. In our implementation, the hyperparameters k_n (nodes) and k_e (edges) do not set the final subgraph size: they determine which nodes/edges receive non-zero prize (top-k_n and top-k_e), while PCST may include additional nodes along connecting paths. We treat nodes and edges independently at the scoring stage so that relation relevance is not constrained by entity selection, and we then apply PCST to obtain a single connected subgraph per subquestion. We use PCST because it yields a connected subgraph for each subquestion while avoiding post-hoc repair of the merged graph, making the resulting structure–content trade-off directly observable.

3.4. Subgraph Merging

We finally merge the resulting subgraphs into G^*, using structural union without duplication.

$$G^* = \bigcup_{q \in \{q_1, \ldots, q_n\}} G_q \tag{3}$$

Note that after merging the per-subquestion connected subgraphs, the final retrieved graph G^* may

split into multiple connected components, and we refer to this phenomenon as fragmentation. In that sense, fragmentation is not caused by failure of the retrieval objective but by the lack of overlap across subquestions. This distinction is important because it lets us study how decomposition changes global graph organization even when each local retrieval step is structurally coherent.

This formulation enables us to explicitly vary retrieval focus and observe how it affects both the structure of the resulting subgraph, and the relevance of retrieved entities. This allows us to answer our two main research questions:

- How does α shape the relevance and structure of the retrieved content? (**RQ1**)
- How do structure and content influence downstream reasoning performance? (**RQ2**)

4. Experimental Evaluation

For question decomposition, we use DeepSeek-AI (2025) as the language model. We manually evaluate the quality of generated subquestions with different LLM sizes in Table 1: we classify each decomposition in one of four categories (excellent, good, weak, poor). For the rest of the study, we choose the 32-B variant of the model.

Model size	Excellent (%)	Good (%)	Weak (%)	Poor (%)
7B	27	11	27	35
14B	34	11	24.5	30.5
32B	69.5	12.5	11.5	7.5

Table 1: Percentage of decomposition quality per model size. Evaluation was performed on a random sample of 200 questions.

To guide the generation of the subquestions, we use the prompt shown in Figure 2. Additionally, we augment the prompt with some decomposition examples, as shown in Figure 3. In particular, we show diverse examples of both complex and simple questions, in which case the LLM must decide whether to decompose the question or not.

We encode both questions and subquestions using Sentence-Transformers (2025). Given the format of entity labels (in the Freebase style), we choose to encode node and edge textual attributes using the same model.

During retrieval, the parameters k_n and k_e control the number of nodes and edges retrieved for each subquestion and directly influence both relevance and graph size. For small values (e.g., $k_n = k_e = 2$), retrieval is highly selective: resulting subgraphs remain compact and contain little irrelevant noise, but often miss important entities or relations needed for reasoning. Conversely, larger values of k (e.g., $k_n = k_e = 10$) increase the likelihood of retrieving relevant nodes and edges, but also introduce substantial noise and lead to rapidly growing merged graphs that are harder for the model to process. We evaluate graph size and relevance across a range of (k_n, k_e) values and observe that beyond $k_n = 5$, gains in relevance diminish while graph size and noise increase sharply. We choose $k_n = k_e = 5$ in our experiments.

```
You are an expert at decomposing complex
    questions into smaller, atomic
    subquestions.
If the question can't be decomposed into
    smaller questions, leave it as it
    is.
Decompose the following question into a
    list of simpler subquestions that:

- Are atomic (addressing only one piece
    of information at a time)
- Are logically ordered
- Have access to answers from previous
    subquestions
- Cover all necessary aspects of the
    original question
- Can be answered with a single entity
- Lead to the answer in the last
    subquestion

You must strictly format your answer as
    a valid JSON array; do NOT include
    explanations or reasoning.

Now decompose the following question in
    JSON format.

Complex Question:
"Which city is the birthplace of the
    author of the novel '1984'?"

Subquestions:
1. Who is the author of the novel
    '1984'?
2. Where was this author born?
```

Figure 2: Example of decomposition prompt for a complex question

For answer generation, we use LLaMA-2-7B and LLaMA-2-13B (Meta AI, 2023b,a) and we use a generation pipeline similar to that in He et al. (2024). We conduct evaluations on CWQ (Talmor et al., 2018) and WebQSP (Yih et al., 2016), two multi-hop QA benchmarks derived from Freebase (Bollacker et al., 2008). Both datasets consist of questions written in English. We follow Luo et al. (2023) for preprocessed datasets and splits. CWQ includes 34,689 questions, while WebQSP includes 4,737 questions. We measure retrieval quality with four

```
Examples:

Input: What is the capital of the
    country that exports the most honey
    ?
Output: ["Which country exports the most
     honey ?", "What is the capital of
    that country ?"]

Input: What sports team does Michael's
    best friend support ?
Output: ["Who is Michael's best friend
    ?", "What sports team does he
    support ?"]

Input: What fruits grow in the hottest
    countries from the largest continent
     in the world ?
Output: ["What is the largest continent
    in the world ?",
         "What countries are hottest on
             this continent ?",
         "What fruits grow in those
             countries ?"]

Input: How old is Obama ?
Output: ["How old is Obama ?"]

Now decompose the following question in
    JSON format.
```

Figure 3: Example of possible few-shot prompting

metrics: percentage of connected graphs, average graph density, strong matching, and exact matching. Strong matching measures whether the retrieved subgraph contains a node that is semantically very close to the answer (cosine similarity ≥ 0.95), while exact matching measures whether the answer entity itself is retrieved. We report both because they capture complementary aspects of retrieval quality: strong matching reflects semantic relevance, whereas exact matching reflects strict recoverability of the target answer. We measure QA accuracy through Hit@1 metric.

Overall, our experiments address two central questions: **(RQ1)** How does α affect subgraph structure and content? **(RQ2)** How does this trade-off affect reasoning?

5. Results

5.1. The Structure-Content Trade-off (RQ1)

Figure 4 illustrates how the retrieval focus parameter α influences both the structure and content of the retrieved subgraph. Lower α values, which emphasize the original question, yield subgraphs that are denser and more likely to be connected, but whose nodes and edges exhibit weaker semantic alignment with the target answer. As α increases, retrieval becomes increasingly driven by subquestions, resulting in higher content relevance but a rapid loss of global connectivity. This behavior reveals an intrinsic structure–content trade-off induced by question decomposition. More concretely, the loss of connectivity at higher α corresponds to fragmentation of the merged graph G^*: each subquestion yields a connected but locally focused subgraph, yet these components increasingly fail to connect after union.

This trade-off can be understood intuitively. When retrieval is guided exclusively by the original question, the PCST optimization procedure enforces connectivity, at the cost of potentially not retrieving all relevant nodes. While this produces structurally predictable subgraphs, it can dilute relevance by including entities that are only loosely related to specific reasoning steps. As we start to increase the importance of subquestions, the retrieval process isolates individual semantic facets of the reasoning chain, maximizing local relevance but retrieving evidence from disparate regions of the graph. This tendency continues to persist as α gets closer to 1, until we no longer use the original question during retrieval. At that point, the merged subgraph often consists of multiple disconnected components (see Figure 5), even though each component is individually informative.

These observations highlight that question decomposition does not merely affect which information is retrieved, but also how that information will be structurally organized. These results raise a central question: how does this retrieval-induced trade-off between structure and content translate into downstream reasoning performance? Is the LLM QA performance impacted by the degree to which the provided information is tightly structured and connected?

5.2. Impact on Reasoning Performance (RQ2)

In Figure 6, we observe that downstream QA accuracy peaks at intermediate values of α (approximately 0.3–0.7), confirming that neither extreme retrieval regime leads to optimal performance. When α is small, the retrieved subgraphs are well connected but often lack sufficiently precise evidence, limiting the LLM's ability to conclude specific reasoning steps. Alternatively, when α grows towards 1, the retrieved content is more relevant but frequently fragmented into disconnected components. In this case, the LLM must implicitly infer relationships between separate graph fragments, making evidence integration more difficult despite high local

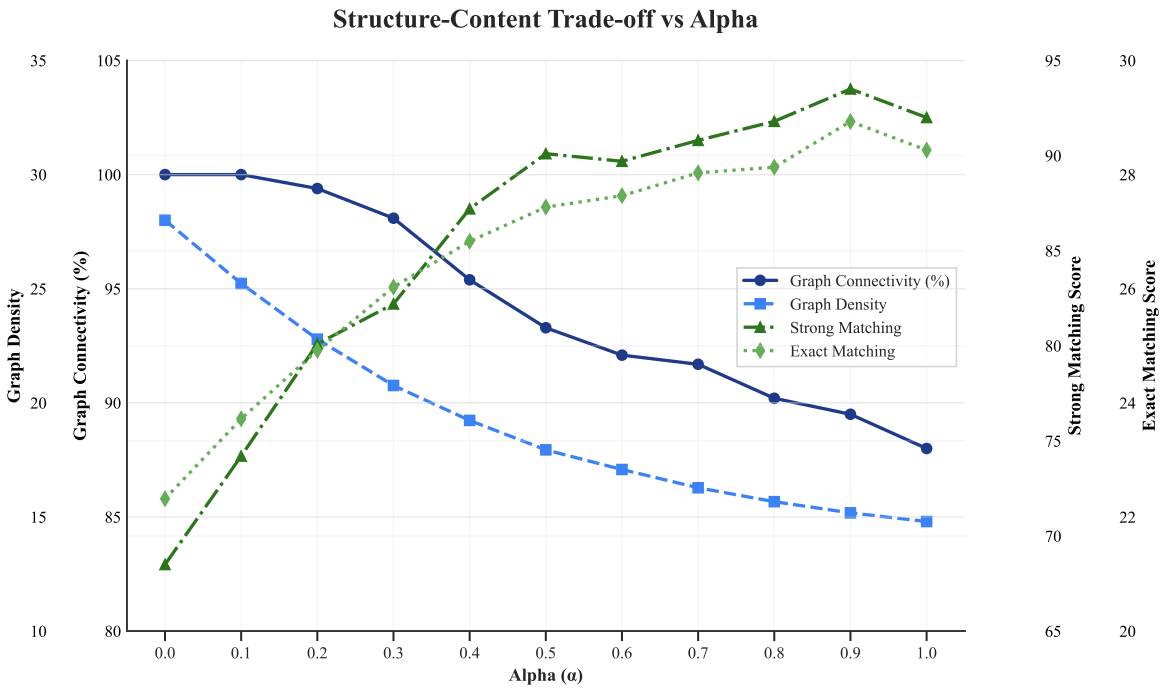

Figure 4: Effect of α on subgraph structure (blue) and content (green). Lower α (focusing on the initial question) increases connectivity and density, but lowers content relevance; higher α (focusing on subquestions) yields opposite observations.

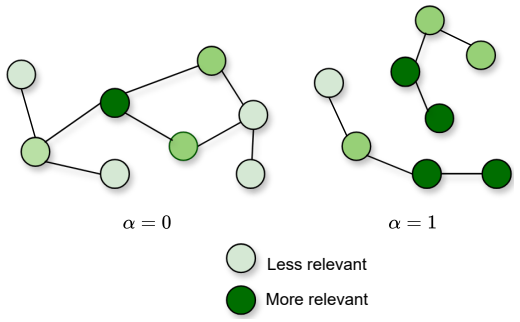

Figure 5: Examples of retrieved subgraphs for different setups: $\alpha = 0$ (focusing on the initial question), and $\alpha = 1$ (focusing on subquestions).

relevance.

These results suggest that providing an LLM with relevant information is not sufficient in itself: the structural features of the retrieved information play a critical role in enabling effective multi-hop reasoning for the LLM. Intermediate values of α naturally balance these factors, yielding subgraphs that remain compact and sufficiently connected while retaining high content precision and thereby supporting improved downstream accuracy.

To further isolate the role of connectivity, we analyze QA performance at $\alpha = 1$ by separating cases where the merged subgraph is connected from those where it is not. We observe that connected subgraphs achieve an average Hit@1 of 55%, compared to 48% for disconnected ones, indicating that fragmentation alone can significantly impair reasoning performance even when retrieved content is relevant. While connectivity does not guarantee correct answers, this gap supports the hypothesis that structural disconnection is a key failure mode of strongly subquestion-focused retrieval.

Finally, we replicate the downstream QA experiments using a larger language model (LLaMA-2-13B). The same qualitative trends are observed: reasoning accuracy again peaks at intermediate α values, and overall performance remains comparable to that obtained with the 7B model. This consistency suggests that the observed structure–content trade-off is not merely an artifact of limited model capacity but reflects a more general interaction between retrieval structure and LLM-based reasoning. Given those implications, an adaptive strategy that sets α based on question complexity, decomposition length, or expected graph fragmentation is a promising direction for future work.

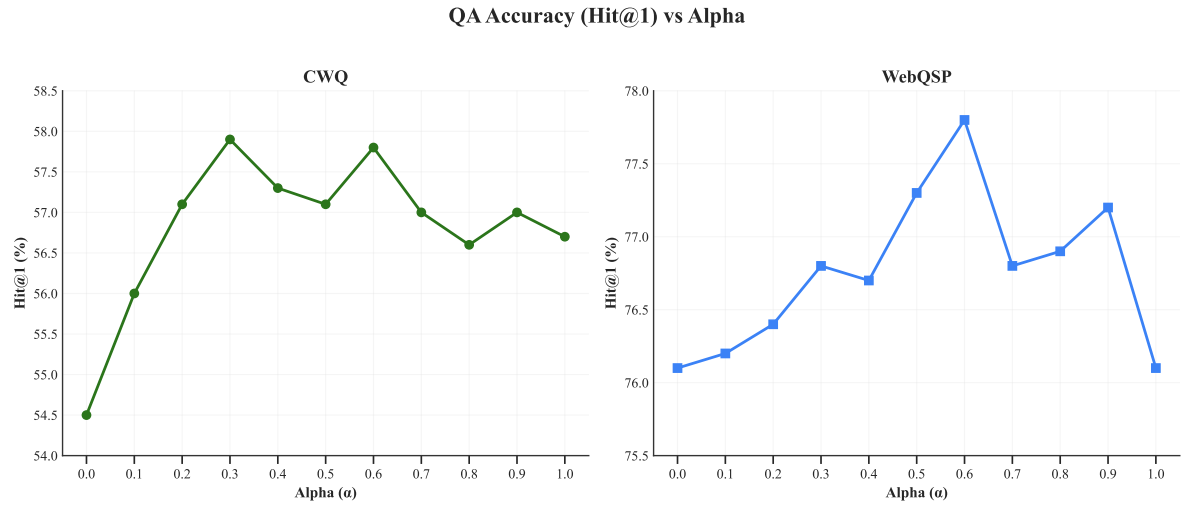

Figure 6: QA accuracy (LLaMa-2-7B) peaks at intermediate α, showing the benefit of balancing structure and content. Note that $\alpha = 0$ corresponds to the setup in (He et al., 2024)

6. Conclusion

This work provides an empirical characterization of how question decomposition shapes knowledge graph retrieval and downstream reasoning. Through a controlled analysis of retrieval focus, we show that decomposition introduces a systematic trade-off between content relevance and structural coherence: subquestion-focused retrieval yields precise but fragmented subgraphs, while question-focused retrieval preserves connectivity at the cost of relevance. Across datasets, downstream QA performance is maximized at intermediate retrieval settings, where relevant evidence remains sufficiently connected to support multi-hop reasoning. These results suggest that the benefits of decomposition cannot be understood purely in terms of relevance but depend critically on how retrieved information is structured.

Our study also opens the door for several directions for future work. Downstream performance is sensitive to the quality of the generated subquestions, motivating the development of more principled ways to evaluate decomposition quality; for example, using large language models as judges to assess coverage or reasoning faithfulness. Moreover, extending the analysis to larger and more capable language models would help confirm that the observed structure-content trade-off reflects a fundamental property of retrieval rather than model-specific limitations. Finally, further work is needed to disentangle the respective contributions of decomposition quality and graph connectivity to downstream performance, potentially through controlled interventions or learned retrieval strategies that adapt structure and content jointly.

7. Ethical Considerations

This work improves the reasoning abilities of large language models by using structured knowledge from textual graphs. While this improves the model's ability to make consistent and transparent predictions, it does not eliminate risks such as the propagation of biases present in the training data or the underlying knowledge graphs. We do not train new language models or use user-generated content. Our experiments are conducted using publicly available datasets. No personal or sensitive data is used. Nevertheless, caution should be exercised when deploying such systems in high-stakes or real-world applications, as flawed reasoning over structured data can result in factually inaccurate outputs.

8. Limitations

Our analysis relies on question decomposition as an intermediate step, and downstream performance is therefore sensitive to the quality of the generated subquestions. While we enforce explicit constraints to standardize decomposition, subquestions may still vary in completeness, specificity, or semantic alignment with the original question. In practice, obtaining high-quality decompositions often requires a sufficiently capable language model, which may limit applicability in settings where only smaller or less reliable models are available. Errors or omissions at the decomposition stage can propagate through retrieval and affect both the relevance and structure of the resulting subgraphs.

In addition, while our results reveal a consistent association between subgraph fragmentation and degraded QA performance, we do not fully disen-

tangle the sources of failure. In particular, incorrect answers may arise either from structural disconnection between retrieved subgraphs or from subquestions that fail to capture essential aspects of the original reasoning chain. Although our diagnostic setup allows us to observe how retrieval focus influences both structure and content, a more fine-grained analysis separating the effects of decomposition quality from those of graph connectivity remains an open direction for future work. Our experiments are conducted on CWQ and WebQSP over Freebase-derived graphs, which do not directly evaluate behavior on larger, noisier, or incomplete knowledge graphs, nor in production retrieval settings. In addition, we intentionally use text-based similarity over node and edge labels to keep retrieval behavior interpretable and to isolate the effect of retrieval focus. Extending the analysis to richer representations, such as relation-aware retrieval or graph-embedding-based scoring, is an important direction for future work.

Acknowledgments

This work has been partially funded by the European Union's Horizon RIA research and innovation program under grant agreement No. 101189679 (ASTIR). This work also benefited from the FactoryIA supercomputer, financially supported by the Ile-de-France Regional Council.

This work was conducted during Valentin's internship at CEA-List.

9. Bibliographical References

Daniel Bienstock, Michel X Goemans, David Simchi-Levi, and David Williamson. 1993. A note on the prize collecting traveling salesman problem. *Mathematical programming*, 59(1):413–420.

Kurt Bollacker et al. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proc. ACM SIGMOD*, pages 1247–1250.

Chi-Min Chan et al. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation. *arXiv preprint arXiv:2404.00610*.

Xinbang Dai et al. 2025. Large language models can better understand knowledge graphs than we thought. *Knowledge-Based Systems*, 312:113060.

Wentao Ding, Jinmao Li, Liangchuan Luo, and Yuzhong Qu. 2024. Enhancing complex question answering over knowledge graphs through evidence pattern retrieval. In *Proceedings of the ACM Web Conference 2024*, pages 2106–2115.

Na Dong, Natthawut Kertkeidkachorn, Xin Liu, and Kiyoaki Shirai. 2025. Refining noisy knowledge graph with large language models. In *Proceedings of the Workshop on Generative AI and Knowledge Graphs (GenAIK)*, pages 78–86.

Ruiliu Fu et al. 2021. Decomposing complex questions makes multi-hop QA easier and more interpretable. In *Findings of EMNLP*, pages 169–180.

Xiaoxin He et al. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907.

Lei Huang et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st annual meeting of the Association for Computational Linguistics (volume 1: long papers)*, pages 5591–5606.

Petri Kotiranta, Marko Junkkari, and Jyrki Nummenmaa. 2022. Performance of graph and relational databases in complex queries. *Applied sciences*, 12(13):6490.

Mufei Li, Siqi Miao, and Pan Li. 2024a. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. *arXiv preprint arXiv:2410.20724*.

Yading Li et al. 2024b. A framework of knowledge graph-enhanced large language model based on question decomposition and atomic retrieval. In *Findings of EMNLP*, pages 11472–11485.

Yixin Liu, Kejian Shi, Katherine S He, Longtian Ye, Alexander R Fabbri, Pengfei Liu, Dragomir Radev, and Arman Cohan. 2024. On learning to summarize with large language models as references. *arXiv preprint arXiv:2305.14239*.

Linhao Luo et al. 2023. Reasoning on graphs: Faithful and interpretable large language model reasoning. *arXiv preprint arXiv:2310.01061*.

Tobias Aanderaa Opsahl. 2024. Fact or fiction? improving fact verification with knowledge graphs through simplified subgraph retrievals. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 307–316.

Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge graphs: Opportunities and challenges. *Artificial intelligence review*, 56(11):13071–13102.

Ethan Perez et al. 2020. Unsupervised question decomposition for question answering. *arXiv preprint arXiv:2002.09758*.

Sangjin Shin and Kyong-Ho Lee. 2020. Processing knowledge graph-based complex questions through question decomposition and recomposition. *Information sciences*, 523:234–244.

Wenqing Wu, Zhenfang Zhu, Jiangtao Qi, Wenling Wang, Guangyuan Zhang, and Peiyu Liu. 2023. A dynamic graph expansion network for multi-hop knowledge base question answering. *Neurocomputing*, 515:37–47.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024. Do large language models latently perform multi-hop reasoning? *URL https://arxiv. org/abs/2402.16837*.

Michihiro Yasunaga et al. 2021. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *Proc. NAACL-HLT*, pages 535–546.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. In *International conference on machine learning*, pages 41092–41110. PMLR.

Xiaohan Zou. 2020. A survey on application of knowledge graph. In *Journal of Physics: Conference Series*, volume 1487, page 012016. IOP Publishing.

10. Language Resource References

DeepSeek-AI. 2025. *DeepSeek-R1-Distill-Qwen-32B*. DeepSeek-AI. Hugging Face. PID https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-32B. Large language model distilled from DeepSeek-R1.

Meta AI. 2023a. *LLaMA 2 13B*. Meta AI. Hugging Face. PID https://huggingface.co/meta-llama/Llama-2-13b-hf. Large language model (13B parameters), Hugging Face version.

Meta AI. 2023b. *LLaMA 2 7B*. Meta AI. Hugging Face. PID https://huggingface.co/meta-llama/Llama-2-7b-hf. Large language model (7B parameters), Hugging Face version.

Sentence-Transformers. 2025. *all-roberta-large-v1*. Sentence-Transformers project. Hugging Face. PID https://huggingface.co/sentence-transformers/all-roberta-large-v1. Neural sentence embedding model based on RoBERTa.

Talmor et al. 2018. *Complex Web Questions (CWQ)*. Tel-Aviv University. PID http://nlp.cs.tau.ac.il/compwebq. Question answering dataset over Freebase.

Yih et al. 2016. *WebQSP*. Microsoft Research. PID https://www.microsoft.com/en-us/research/publication/the-value-of-semantic-parse-labeling-for-knowledge-base-question-answering-2/. Question answering dataset over Freebase.

A. Default Hyperparameters and Experimental Setting

This appendix documents the default runtime configuration used in our codebase. Unless explicitly stated otherwise in the main text, experiments use these defaults. For all reported results, we select checkpoints by validation loss and evaluate the best checkpoint on the test set.

A.1. Command-Line Defaults

The shorthand fields in Table A.1 correspond to CLI arguments (`workers`→`num_workers`, `epochs`→`num_epochs`, `warmup`→`warmup_epochs`, `eval_bs`→`eval_batch_size`, `virt_tok`→`llm_num_virtual_tokens`). When `frozen=True`, LLM backbone weights are frozen and training focuses on graph encoder/projector (or prompt parameters for prompt tuning). If a model path is not explicitly passed, it is auto-resolved from the configured local model map.

A.2. Internal Defaults Used by the Implementation

In decomposition-aware retrieval, node/edge relevance blends original-question and subquestion signals (see Equation 1). Thus, $\alpha = 0$ uses only

Group	Param	Default
General	model_name	graph_llm
General	dataset	cwq
Training	lr	$1e{-5}$
Training	wd	0.05
Training	patience	2
Training	batch_size	2
Training	grad_steps	2
Training	workers	min(8, cpu)
Training	epochs	10
Training	warmup	1
Eval	eval_bs	16
LLM	model	7b
LLM	frozen	True
LLM	virt_tok	10
LLM	max_len	512
LLM	gen_len	32
LLM	max_memory	[80, 80] GiB (per GPU index)
GNN	layers	4
GNN	dim	1024
GNN	heads	4
GNN	dropout	0.0
Pipeline	alpha	0.5

Table 2: Main CLI defaults (simplified).

Component	Param	Default
PCST retrieval	topk, topk_e, cost_e, α	3,5,0.5,0.5
CWQ preprocess	topk, topk_e, cost_e	5,7,0.5
WebQSP preprocess	topk, topk_e, cost_e	3,5,0.5
LoRA (if unfrozen)	r, α, dropout	8,16,0.05
Optimizer	AdamW β	(0.9, 0.95)
Grad clip	max norm	0.1
LR schedule	min LR	5e-6
Generation	temp, top_p	1, 1

Table 3: Internal defaults.

the original question, while $\alpha = 1$ uses only the current subquestion.

A.3. Protocol Notes

For CWQ, preprocessing follows the project setting that uses the first 1000 test examples. Early stopping is based on validation loss with patience $= 2$, and the best validation checkpoint is reloaded for final test evaluation. Generation is deterministic in the default GraphLLM inference path (do_sample=False, temperature=1, top_p=1).