A Navigational Approach for Comprehensive RAG via Traversal over Proposition Graphs

Anonymous ACL submission

Abstract

Standard RAG pipelines based on chunking excel at simple factual retrieval but fail on complex multi-hop queries due to a lack of structural connectivity. Conversely, initial strategies that interleave retrieval with reasoning often lack global corpus awareness, while Knowledge Graph (KG)-based RAG performs strongly on complex multi-hop tasks but suffers on fact-oriented single-hop queries. To bridge this gap, we propose a novel RAG framework: ToPG (Traversal over Proposition Graphs). ToPG models its knowledge base as a heterogeneous graph of propositions, entities, and passages, effectively combining the granular fact density of propositions with graph connectivity. We leverage this structure using iterative Suggestion-Selection cycles, where the Suggestion phase enables a query-aware traversal of the graph, and the Selection phase provides LLM feedback to prune irrelevant propositions and seed the next iteration. Evaluated on three distinct QA tasks (Simple, Complex, and Abstract QA), ToPG demonstrates strong performance across both accuracy- and quality-based metrics. Overall, ToPG shows that query-aware graph traversal combined with factual granularity is a critical component for efficient structured RAG systems. ToPG is available at `anonymous-link`[1].

1 Introduction

Retrieval-Augmented Generation (RAG) has become the dominant paradigm for grounding Large Language Models (LLMs). RAG directly addresses the limitations of static parametric memory, mitigating hallucinations (Huang et al., 2025) and improving recall, particularly for long-tail knowledge (Kandpal et al., 2023). The standard RAG pipeline relies on Dense Passage Retrieval (DPR) over chunked documents (Karpukhin et al., 2020).

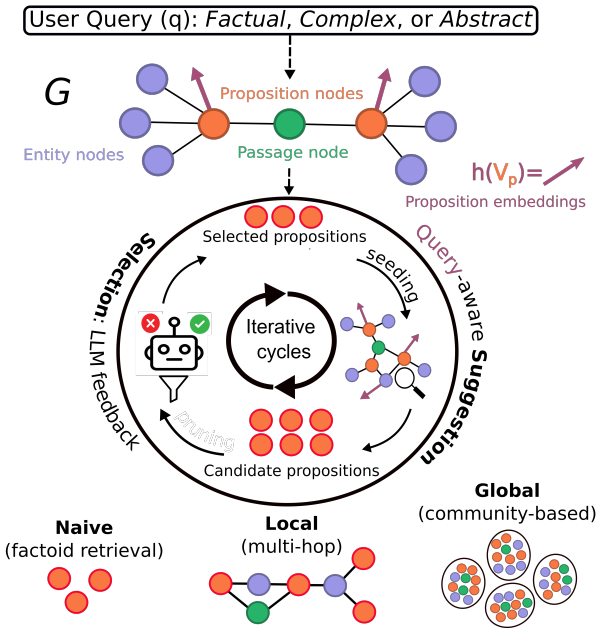

Figure 1: The ToPG framework. The system operates on a heterogeneous graph where propositions connect entities and passages. ToPG navigates this graph using iterative Suggestion-Selection cycles, allowing for three operational modes: `Naive` (factoid retrieval), `Local` (multi-hop inference), and `Global` (community-based search).

While large embedding models (e.g., NV-Embed-v2 (Lee et al., 2025)) have achieved state-of-the-art performance on the MTEB benchmark (Muennighoff et al., 2023), retrieval granularity represents a critical, often overlooked line for improvements. Coarse-grained passages often contain irrelevant or distracting information that degrades LLM generation (Shi et al., 2023). Conversely, proposition-level retrieval (decomposing text into decontextualized atomic facts) has proven superior for direct, single-hop QA and fact checking (Chen et al., 2024b; Min et al., 2023).

Real-world complex queries often require multi-hop reasoning that necessitates connecting disparate pieces of evidence across documents. While

[1]The code will be made publicly available following the anonymity period.

iterative retrieval and Chain-of-Thought (CoT) approaches (Trivedi et al., 2023) commonly operationalize this via successive local searches, they inherently lack a global, structured view of the corpus. To bridge this structural gap, structure-augmented RAG strategies have integrated Knowledge Graphs (KGs) (Zhang et al., 2025). These methods explicitly model entities and relationships to support both multi-hop inference and broader, abstract queries (Edge et al., 2025).

Despite their structural advantages, current approaches face fundamental challenges. First, they lead to information loss as standard KGs enforce triples (s,p,o) representations, compressing complex text into binary relations. Second, a practical challenge exists in navigating the graph. Current strategies are broadly polarized between methods relying on purely topological heuristics (e.g., neighbours, random walks, etc.) and thus inherently ignoring edge semantics, supervised GNNs (Mavromatis and Karypis, 2025), or LLM-driven exploration (Sun et al., 2024). To this end, we introduce ToPG (Traversal Over Proposition Graphs), a novel RAG framework that combines the granularity of propositions with *query-aware graph traversal* (Figure 1).

Unlike traditional KGs, we model the knowledge base as a heterogeneous graph of entities, propositions, and passages. This structure retains the semantic richness of atomic facts while enabling the topological connectivity of a graph. To leverage this structure, we propose a graph exploration method based on *Suggestion-Selection cycles*. The Suggestion phase leverages both query similarity and graph topology to efficiently suggest new relevant propositions. The subsequent Selection phase acts as a feedback mechanism, using in-context LLM-based interpretation to prune irrelevant suggestions and seed the next iteration with high-quality evidence. To address diverse QA requirements, ToPG supports three complexity levels: `Naive` proposition retrieval, `Local` multi-hop inference, and `Global` community-based abstract QA.

2 Methods

2.1 Graph Construction

We represent the knowledge base as a heterogeneous graph $G = (V, E)$ where the node set $V = V_p \cup V_e \cup V_P$ comprises three disjoint types of nodes: atomic factual statements (propositions V_p), named entities that appear within propositions (entities V_e), and document segments that provide the source context for propositions (passages V_P). The edge set $E = E_{p\leftrightarrow e} \cup E_{p\leftrightarrow P}$ contains two types of undirected edges, connecting each proposition to its associated entities and to the passage from which it originates.

Given a document chunk, we apply an LLM-based in-context learning function (in a few-shot setting) to sequentially extract named entities V_e and propositions V_p. Each extracted proposition and entity node is encoded using an encoder $h(\cdot)$. Entity reconciliation is performed using cosine similarity thresholding. Details in Appendix A.

In the resulting graph, each proposition effectively acts as a hyperedge linking multiple entities while being grounded in textual evidence[2]. Passage nodes (V_P) serve a structural role by connecting propositions originating from the same passage, thereby enforcing local neighborhood coherence. Unlike classical *entity-centric* KGs, our representation is explicitly *proposition-centric*: propositions are modeled as first-class nodes, enabling richer reasoning over factual, compositional, and multi-hop relations.

2.2 Graph Navigation: Suggestion–Selection Cycles

Suggestion-Selection Retrieval. We propose to navigate the graph G through Suggestion-Selection cycles. Conceptually, the suggestion step defines a function:

$$S_{\text{new}} = \text{Suggest}_k(q, G, s_{\text{old}}) \quad (1)$$

Given a query q, a graph G and a set of already collected proposition nodes s_{old}, proposes k new potentially relevant nodes S_{new}.

An effective suggestion mechanism should account for both the semantic relevance of nodes to the query q, and the connectivity of nodes to the seed set s_{old} in G. Therefore, the suggestion process should ideally be both query and graph aware.

The Selection phase defines a function that prune irrelevant propositions from the pool S_{new}:

$$s_{\text{new}} = \text{Select}(q, S_{\text{new}}) \quad (2)$$

It acts as feedback from the LLM and seeds the next iteration of Suggest by performing LLM-based relevance pruning: $\text{PROMPT}_{\text{Select}}(q, S_{\text{new}})$. By modulating the query q and collected propositions s_{old}

[2]Reciprocally, entities also create hyperedges between propositions.

during iterative Suggestion-Selection cycles, we can adapt the exploration behavior over G to different question types (see section 2.3).

Query and Graph Aware Suggestions We introduce a retrieval strategy based on a query-aware Personalized PageRank (PPR) (Haveliwala, 2002). An intuitive example is available in Appendix B.1. We propose to determine new candidate propositions as

$$S_{\text{new}} = \text{top}_k\big(\text{PPR}(M, s_{old})\big), \tag{3}$$

where the transition matrix M combines structural and semantic information:

$$\begin{aligned} M &= \text{QueryAwareTransiton}(q, G, \lambda) \\ &= \lambda T_s + (1-\lambda) T_n. \end{aligned} \tag{4}$$

The parameter λ controls the balance between structural and semantic guidance in M. The structural component T_s encodes the topology of G: the higher the connectivity between two propositions through shared entities or passages, the greater the probability of transition between them. Thus, T_s captures connectivity to the seed nodes, but, is independent of the current query q. In contrast, the semantic component T_n maintains the same adjacency pattern as T_s, but weights each potential transition (i, j) according to the similarity between node j and the query q, making nodes similar to the query more attractive. Therefore, random walks are biased toward proposition nodes that are not only structurally connected to the current context s_{old}, but, also semantically relevant to the question. Intuitively, the resulting transition matrix M encourages exploration along paths that remain consistent with the graph structure, while biased toward semantically relevant regions. Then, setting $\lambda = 1$, gives a purely graph-based and query independent Suggest function.

More formally, $T_s \in \mathbb{R}^{n \times n}$ is the degree-normalized transition matrix derived from the proposition–entity–passage connectivity:

$$T_s = \tilde{A}_{p \to eP}\, \tilde{A}_{eP \to p}, \tag{5}$$

where $\tilde{A}_{p \to eP}$ and $\tilde{A}_{eP \to p}$ denote the normalized transition matrices between propositions and entities/passages from the graph.

Then, to build T_n, we compute the query-based similarities $c = \text{cosine}(h(q), h(V_p))$ and apply temperature scaling and thresholding:

$$\tilde{c}_i = \begin{cases} \exp(c_i/\tau), & \text{if } c_i \geq \theta, \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where τ is the temperature (default 0.1) and θ is the cosine threshold (default 0.4). The semantic transition matrix is then defined as

$$T_n(i, j) = \frac{\tilde{c}_j\, \mathbf{1}_{T_s(i,j)>0}}{\sum_k \tilde{c}_k\, \mathbf{1}_{T_s(i,k)>0}}. \tag{7}$$

In both T_s and T_n, self-connections are also canceled (eg. $T_s(i, i) = 0$).

Subgraph Extraction As traversing the full graph G is computationally expensive, we first extract a local subgraph G^* using a Random Walk with Restart (hereafter named GExtract) around the set of seeds with a target size l. This process mitigates hub bias and constrains the exploration space.

2.3 `Naive`, `Local` and `Global` modes

We propose three search modes: `Naive` for simple factual queries, `Local` for complex (eg., multi-hop) queries, and `Global` for abstract questions.

2.3.1 `Naive`

Using a retrieval encoder h, a simple top-k retrieval based on cosine similarity $S_{\text{new}} = \text{top}_k\big(\text{cosine}(h(q), h(V_p))\big)$ can be seen as a naive suggestion process: a retrieval over propositions that ignores both the graph G and previously collected nodes s_{old}. We define this as $\text{SuggestNaive}_k(q, G, \emptyset)$. `Naive` mode uses propositions retrieved from SuggestNaive as context to answer the question, bypassing the Selection step.

2.3.2 `Local`

In the `Local` mode, the graph G is explored through Suggestion-Selection cycles guided by LLM feedback up to a maximum number of iterations (`max-iter`). A step-by-step example is provided in Figure 2-`Local`. Starting from an initial query q_{start}, it completes a local set of propositions $s_{\text{loc}} = \{u_1, u_2, ..., u_m\}$ that represent the evolving context for answering the query. The initial propositions are collected with SuggestNaive and the irrelevant ones are pruned by Select (step 1). This represents the initial *seeding* step in G.

Then, at each iteration, new candidate propositions (S_{new}) are proposed using SuggestLocal, conditioned on the current query and the current context (s_{pool}). Candidates are then pruned by Select. The retained propositions are added to s_{loc} and will seed the next iteration. In step 2, the first iteration yields nodes u_1 and u_2.

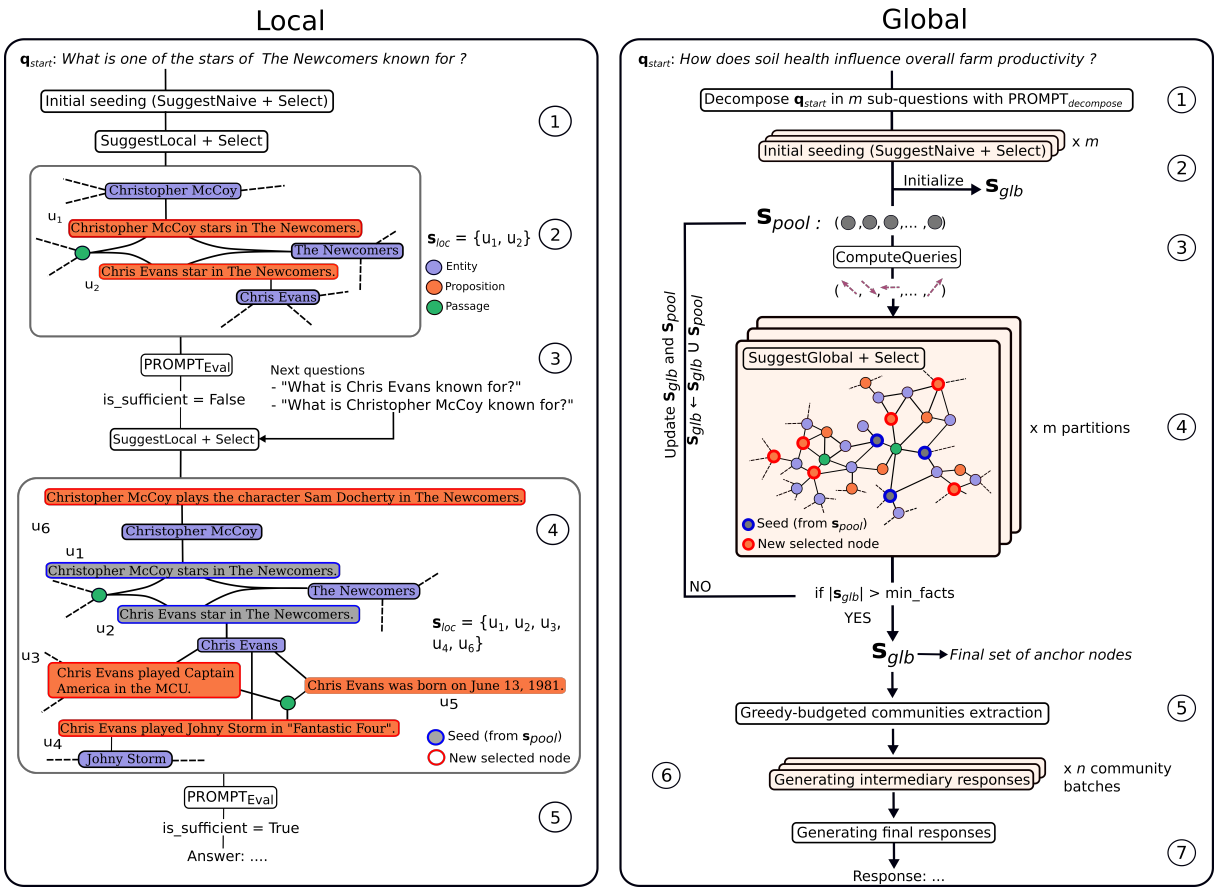

Figure 2: On the left, panel `Local` shows a step-by-step example for the `Local` mode. On the right panel, step-by-step description of the `Global` mode.

The suggestion $S_{\text{new}} = \text{SuggestLocal}_k(q, G, s_{\text{pool}})$ proceeds in three stages:

$$\begin{aligned} G^* &= \text{GExtract}(G, s_{\text{pool}}, l), \\ M &= \text{QueryAwareTransiton}(q, G^*, \lambda), \\ \pi &= \text{PPR}(M, s_{\text{pool}}), \\ S_{\text{new}} &= \text{top}_k(\pi) \end{aligned} \tag{8}$$

If the accumulated propositions remain insufficient to answer the query (determined by $\text{PROMPT}_{\text{Eval}}(q_{\text{start}}, s_{\text{loc}})$), additional targeted sub-questions are generated to guide the next iteration via $\text{PROMPT}_{\text{NextQ}}(q_{\text{start}}, s_{\text{loc}})$. In the running example (step 3), this yields two new queries, which in turn trigger a second Suggestion–Selection cycle. This cycle is seeded on s_{pool}, containing u_1 and u_2 (step 4). At this iteration, S_{new} contained the suggested propositions: u_3, u_4, u_5, and u_6 and the Select call pruned u_5. With s_{loc} now completed by u_3, u_4 and u_6, the query is answered in (step 5). Details and an illustrative example in Appendix B.2.

2.3.3 `Global`

While `Local` retrieval effectively handles fact-oriented or multi-hop reasoning tasks, abstract or conceptual queries *"How does soil health influence overall farm productivity?"* require a broader and more diverse exploration of the graph G. In such cases, identifying a missing fact or reasoning chain is insufficient, where a comprehensive answer spans multiple, complementary and non-local perspectives that need to be retrieved from G. Rather than using $\text{PROMPT}_{\text{NextQ}}$ to predict new directions/questions as in `Local`, `Global` refines queries after each Suggestion-Selection cycle. Gathered anchor propositions s_{glb}, are then used to identify communities in G. Communities emerge naturally from the graph's topology and can represent potential facets (i.e., individual aspects or perspectives) relevant to the query. Intermediate answers are generated from these communities, scored by relevance, and aggregated into the final response. A detailed diagram of the step-by-step process is presented in Figure 2-Global.

Steps 1-2: Seeding The parameter m controls the breadth of the exploration. We begin by decomposing the initial query q_{start} into m sub-queries using $\text{PROMPT}_{\text{decompose}}$. Each sub-query is sent to SuggestNaive and populates the pool of anchors s_{pool} after irrelevant propositions are pruned with Select. This corresponds to the *seeding* step and the first anchors added to s_{glb}.

Step 3: Compute Queries At each subsequent iteration, every proposition node in the current pool ($u_i \in s_{\text{pool}}$) becomes an independent exploration center, performing its own local walk through the graph. To guide these walks, we refine the query q_i for each proposition using relevance feedback (Rocchio Jr, 1971):

$$q_i = \alpha q_i^o + \beta q_i^+ - \gamma q_i^- . \tag{9}$$

Intuitively, q_i^o captures directions that previously led to u_i; q_i^+ encodes the proposition u_i as a relevance signal from Select, encouraging further exploration in this direction; and q_i^- discourages directions that were previously pruned. Therefore, query vectors q_i are refined by the LLM feedback provided by Select, guiding the exploration toward promising directions while avoiding previously pruned paths (further details in Appendix B.3).

Step 4: Iteratively explore and collect To prevent the search space from growing as more facts accumulate, s_{pool} is partitioned into m subsets. The SuggestGlobal strategy operates independently on each partition s_{part} with its associated queries $\mathbf{q}_{\text{part}}$. Each node u_i in s_{part} has its own query q_i in $\mathbf{q}_{\text{part}}$. Within a partition, a subgraph G^* is extracted around s_{part}, then, each proposition u_i acts as a singleton seed $\{u_i\}$ with its query q_i, performing an individual query-aware random walk. The resulting probability distributions are aggregated within the partition, and the top-k propositions are selected as new suggestions. In summary, $\text{SuggestGlobal}_k(\mathbf{q}_{\text{part}}, G, s_{\text{part}})$ is defined as:

$$\begin{aligned} G^* &= \text{GExtract}(G, s_{\text{part}}, l), \\ M_i &= \text{QueryAwareTransiton}(q_i, G^*, \lambda), \\ \pi_i &= \text{PPR}(M_i, \{u_i\}), \\ S_{\text{new}} &= \text{top}_k\big(\sum_i \pi_i\big) \end{aligned} \tag{10}$$

The process is iterated until `min_facts` are collected (or `max-iter` iterations are completed). See details in Appendix B.4.

Step 5: Identifying communities The final set of collected anchor propositions s_{glb} is used to extract associated communities from G. Communities are identified using a hierarchical Leiden algorithm (Traag et al., 2019). We then follow a greedy budgeted strategy: each community c is assigned a score $\frac{|nodes(c) \setminus S'|}{size(c)}$, representing how many yet-uncovered anchor propositions (S') it includes relative to its size. Given a budget limit of B total nodes, communities with the highest score are iteratively added to maximize coverage over anchor propositions. See details of the procedure in B.5.

Steps 6-7: Generating answers Each community contains a heterogeneous mix of nodes: propositions, entities, and passages. Entities highlight central topics, propositions ground the key facts, and passages provide broader context and connect propositions together. Community content is divided into chunks of pre-specified token size and used to generate intermediary answers. Intermediary answers are ranked and combined into the final prompt, inserting the most relevant information at the beginning and the end ("lost-in-the-middle" effect (Liu et al., 2024)), before generating the final answer.

3 Experimental Setup

We evaluate `Naive`, `Local`, and `Global` modes across complementary QA settings. Our evaluation spans: (i) **Simple QA**, testing the ability to retrieve isolated factual evidence; (ii) **Complex QA**, requiring the retrieval and composition of multiple evidence (eg., multi-hop queries); and (iii) **Abstract QA**, involving conceptual or multi-faceted queries that require broad, long-form synthesis beyond explicit facts.

3.1 Datasets

Simple QA Following prior work (Gutiérrez et al., 2024), we evaluated on a subset of 1,000 queries from PopQA (Mallen et al., 2023) and included GraphRAG-Benchmark (Xiang et al., 2025) Task 1 (*Fact Retrieval*), covering two distinct corpora: Medical, containing NCCN clinical guidelines, and Novel, a collection of pre-20th-century literary texts from Project Gutenberg.

Complex QA. We used the 1,000-query subsets of the multi-hop QA datasets HotPotQA (Yang et al., 2018) and MusiQue (Trivedi et al., 2022) from Gutiérrez et al. (2025). We also included

two GraphRAG-Benchmark tasks: *Complex Reasoning*, which requires chaining multiple evidence, and *Contextual Summarization*, which requires synthesis of fragmented information. For them, we follow the Answer Accuracy metric (Xiang et al., 2025).

Abstract QA. To evaluate abstract queries, we follow the LightRAG setup (Guo et al., 2025) and generate abstract questions on three corpora from the UltraDomain benchmark (college-level textbooks): Agriculture, Computer Science, and Legal (Qian et al., 2025). We compare responses on 4 dimensions with LLM-as-a-judge (Gu et al., 2025): Comprehensiveness, Diversity, Empowerment and finally Overall. See details and examples of queries in Appendix C.

3.2 Settings and Baselines

For Simple and Complex QA, we evaluated three structure-augmented RAG baselines: GraphRAG (Edge et al., 2025), LightRAG (Guo et al., 2025), and HippoRAG 2 (Gutiérrez et al., 2025). For both Simple and Complex QA, we used $k = 20$ propositions for `Naive` and `Local` modes, assessing the latter with `max-iter` $\in \{1, 3\}$. Hyperparameters analysis can be found in Appendix D. To isolate the benefits of proposition-level retrieval, we also include a vanilla passage-level RAG baseline that uses the same prompting configuration as the `Naive` mode.

For Abstract QA, we evaluate the `Global` mode with varying numbers of collected anchor propositions (200–1000). We compare two variants of query refinement: `Rocchio-style feedback` using $(\alpha{=}1,\ \beta{=}0.7,\ \gamma{=}0.15)$ (Rocchio Jr, 1971), incorporating both selected and pruned propositions; and `Simple-feedback`, where q_i ignores signals from Select $(\alpha{=}1,\ \beta{=}\gamma{=}0)$. For fairness, we evaluate against GraphRAG and LightRAG, as both explicitly support abstract-level QA with dedicated global/hybrid modes.

To ensure fair comparison, all baselines use the same embedding model (`bge-large-en-v1.5`) and the same open LLM `Gemma-3-27B` (Team et al., 2025) for indexing and inference using vLLM (Kwon et al., 2023) on one H100. For experiments on GraphRAG-Benchmark, we align with the associated protocol and used GPT-4o-mini (OpenAI et al., 2024) for both indexing and inference. For additional details, please see Appendix E.

4 Results

Table 1 reports QA performance for Simple and Complex QA tasks. On Simple QA, `Naive` mode and Vanilla-RAG on passages outperform graph-based approaches by a significant margin, particularly on PopQA. However, the structural advantages of graph-based methods become apparent in Complex QA settings, notably in multi-hop scenarios where ToPG-`Local` demonstrates superior performance. Interestingly, even when configured with `max-iter` $= 1$, ToPG-`Local` already exhibits significant improvements in multi-hop settings compared to its `Naive` mode. Increasing iterations to `max-iter` $= 3$ yields substantial gains in multi-hop tasks but offers only marginal improvements in Complex Reasoning and Contextual Summarization (Medical and Novel corpora). For summarization tasks, both GraphRAG (local) and HippoRAG 2 also achieve competitive performance.

Figure 3 illustrates the win rates of ToPG-`Global` against baselines on Abstract QA across four criteria (see an example in Appendix F). ToPG-`Global` significantly outperforms LightRAG across all configurations, reaching comparable performance with GraphRAG ($\approx 50\%$ win rate) on the Agriculture and CS datasets, though it underperforms on the Legal dataset. While GraphRAG consistently outperforms on the Comprehensiveness axis, ToPG achieves greater diversity and is perceived as more empowering in its answers. For all criteria except Comprehensiveness, increasing the number of collected facts shows a positive impact that plateaus around 600 propositions, beyond which performance stagnates or degrades. In contrast, feedback settings for query refinement show only a negligible impact on overall performance, providing a minor improvement only in Comprehensiveness.

Figure 4 compares the average token cost per abstract query, indicating that LightRAG has the lowest token cost for both input and output tokens. GraphRAG is identified with the highest token cost, particularly regarding input tokens. ToPG is cheaper than GraphRAG in completion tokens when configured with less than 600 collected anchors, but is more costly on the MusiQue dataset.

5 Discussion

Graph-based approaches demonstrate competitive performance, particularly in Complex QA (multi-

Method	MusiQue	HotPotQA	PopQA	MEDICAL†			NOVEL†		
				FR	CR	CS	FR	CR	CS
Vanilla-RAG	19.7 / 30.6	52.7 / 65.5	49.2 / 62.2	63.7	57.6	63.7	58.8	41.4	50.1
GraphRAG (local)	17.8 / 26.70	47.3 / 60.2	38.1 / 52.6	38.6	47.0	41.9	49.3	50.9	**64.4**
LightRAG (local)	16.7 / 25.62	48.0 / 59.9	39.7 / 53.4	62.6	63.3	61.3	58.6	49.1	48.9
HippoRAG 2	24.7 / 36.2	55.1 / 66.9	38.4 / 48.6	66.3	62.0	63.1	60.1	53.4	64.1
ToPG-Naive	19.5 / 30.3	49.2 / 61.0	**51.6 / 63.9**	**72.9**	68.5	67.7	67.3	**55.6**	63.7
ToPG-Local (1)	28 / 41.1	55.3 / 67.8	48.4 / 59.5	72.5	68.5	**68.8**	67.0	55.0	61.2
ToPG-Local (3)	**34.0 / 47.0**	**59.3 / 72.7**	48.9 / 60.2	72.6	**69.2**	68.3	**67.6**	53.9	61.0
Δ Local (3) - Naive	↑ 14.5 / 16.7	↑ 10.1 / 11.7	↓ 2.7 / 3.7	↓ 0.3	↑ 0.5	↑ 0.6	↑ 0.3	↓ 1.7	↓ 2.7

Table 1: Results on Simple and Complex QA tasks, highlighting the **best** and second-best results. Performance (Exact Match / F1) on Simple and Multi-Hop QA (Left: MusiQue, HotPotQA, PopQA) and GraphRAG-Benchmark tasks (Right: Fact Retrieval, Complex Reasoning, Contextual Summarization) measured using the Answer Accuracy metric. ToPG-Local (1 and 3) report results for (max-iter = 1 and 3) respectively. Δ Local (3) - Naive shows the difference between Local (max-iter = 3) and Naive.

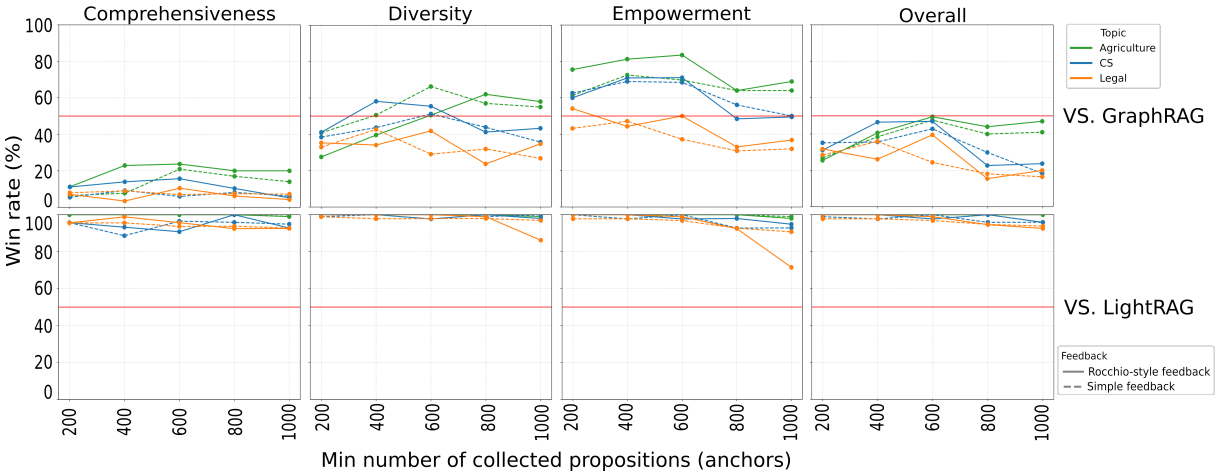

Figure 3: Win rates (%) of ToPG against GraphRAG and LightRAG across 3 corpora and 4 criteria, with increasing number of collected propositions (200-1000) and w/ or w/o Rocchio-style feedback.

hop), where the graph layer effectively connects disparate named entities central to the query. Similarly to (Han et al., 2025), we also note that this structural advantage, however, is often detrimental or minimal for standard factual QA, where proposition-level retrieval with ToPG-Naive achieves higher information density due to their self-contained and factoid content.

While baselines typically construct a standard KG with subject-predicate-object triples, their traversal often relies purely on topological heuristics (e.g., neighbours, random walks), thus neglecting the semantics encoded in the predicate. ToPG proposes a query and graph aware Suggestion mechanism to explicitly leverage the semantics of propositions, coupled with an LLM-driven Selection step that provides explicit feedback for the next iteration, but, entails the overall token cost. This Suggestion-Selection mechanism, even with only one iteration (max-iter = 1), significantly improves performance over ToPG-Naive and alternative baselines in multi-hop settings.

In abstract QA, both GraphRAG and ToPG-Global rely on iterative graph exploration and exploit the inner graph modularity to extract and generate intermediary answers from node communities. While this process significantly increases token costs, it significantly improves the depth (comprehensiveness, diversity, empowerment) of generated answers over simpler keyword expansion strategies (e.g., LightRAG). Moreover, ToPG is designed for easier scalability and updates as it avoids pre-computing community summaries and instead uses Suggestion-Selection cycles for community exploration. Our observations suggest that the utility of collecting additional anchors is saturated by the current LLM's reasoning capacity, implying that further benefits would only arise when using a

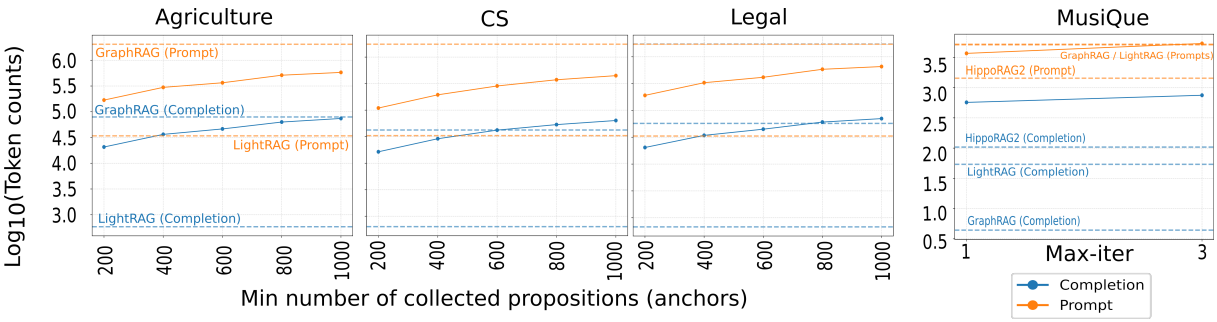

Figure 4: Log_{10}(Token-counts) between baselines evaluated on Agriculture, CS, Legal, MusiQue datasets.

stronger base model.

Overall, our results suggest that query-aware exploration over a graph of granular information units is the critical component, rather than the formal structure of the KG with strict predicates. Traversal of the proposed heterogeneous graph through an effective Suggestion-Selection mechanism shows robust performance and versatility across different QA tasks.

6 Related Work

Early strategies for complex question answering combine retrieval with reasoning via interleaving DPR with CoT or question decomposition techniques (Trivedi et al., 2023; Press et al., 2023; Patel et al., 2022; Shao et al., 2023), an approach likewise employed in the `Local` mode. Furthermore, propositions have been explored as an efficient granularity level, particularly for fact-oriented QA (Chen et al., 2024c) and claim or fact checking (Min et al., 2023; Kamoi et al., 2023). To address the need for global structural awareness, recent approaches construct KGs directly from the corpus (Zhang et al., 2025). These systems seed retrieval using DPR over entities or triples and then navigate the resulting graph using topological heuristics such as community detection (GraphRAG) (Edge et al., 2025), ego-network (LightRAG) (Guo et al., 2025), path search (PathRAG) (Chen et al., 2025), or Personalized PageRank (HippoRAG, HippoRAG 2) (Gutiérrez et al., 2024, 2025).

Combining propositions with graph structure, Wang and Han (2025) proposes to apply a similar approach to HippoRAG on a graph where nodes represent entities and passages, and edges link entities that co-occur within the same proposition. Luo et al. (2025) instead constructs a graph of propositions and perform neighborhood expansion after an initial seeding step. Unlike these approaches, ToPG leverages its Suggestion-Selection cycles and query-aware traversal to support three distinct modes tailored to different QA requirements: factoid, multi-hop, and abstract.

The Selection phase, which provides LLM-based feedback, also aligns with a broader line of work on LLM-guided KG exploration (Sun et al., 2024; Chen et al., 2024a; Ma et al., 2025). These approaches typically alternate phases of search and pruning over entities and relations in the KG. Finnaly, in contrast to GraphRAG or RAPTOR (Sarthi et al., 2024), which rely on pre-processed summaries for abstract QA (Xu et al., 2022; Papakostas and Papadopoulou, 2023), ToPG instead derives intermediary answers directly from the communities extracted around anchor nodes obtained through multiple Suggestion-Selection cycles.

7 Conclusion

ToPG reconciles fact-level granularity with graph connectivity through a heterogeneous graph composed of passages, propositions, and entities. The proposed graph navigation strategy based on iterative Suggestion-Selection cycles, while simple by design, proves highly versatile and adaptable to diverse QA requirements. The strategic modulation of the query and the collected evidence enables distinct operational modes: `Naive` (for factoid retrieval), `Local` (for complex, multi-hop reasoning), and `Global` (for abstract questions). Overall, our experiments demonstrate the efficacy of this framework and suggest that structure-augmented RAG architectures should prioritize query-aware graph traversal and factual granularity over the restrictive formal structure of traditional KGs.

8 Limitations

A primary limitation of our framework is the computational overhead in token cost, both during in-

dexing and inference. Similar to other structure-augmented methods, the process of extracting propositions and building the graph significantly increases indexing costs compared to standard RAG. During inference, token consumption is inflated by the LLM-driven Selection phase (in `Local` mode) and the generation of intermediate community answers (in `Global` mode). While these mechanisms are essential for answer depth, they make ToPG less suitable for cost-critical scenarios compared to lighter alternatives like LightRAG. Future work could mitigate this by replacing the LLM selector with a specialized, lightweight classifier or by fine-tuning prompts for token efficiency.

Second, performance is also bound by the quality of the underlying graph. Relying on embedding similarity for entity disambiguation can occasionally introduce noisy or misleading edges. Furthermore, while proposition extraction enhances information density, it may result in minor information loss compared to full paragraphs. Therefore, integrating external knowledge bases (e.g., Wikipedia or DBpedia) for more robust entity linking and maintaining hybrid access to original passages could be beneficial. However, we deliberately restricted our evaluation to the proposition level for this work.

Finally, while ToPG offers three distinct operational modes (`Naive`, `Local`, `Global`), the current framework lacks an automated routing mechanism. A learned classifier capable of dynamically selecting the optimal mode based on the query complexity would make the framework more end-to-end.

References

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. *Preprint*, arXiv:2502.14902.

Liyi Chen, Panrong Tong, Zhongming Jin, Ying Sun, Jieping Ye, and Hui Xiong. 2024a. Plan-on-graph: Self-correcting adaptive planning of large language model on knowledge graphs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Sihao Chen, Hongming Zhang, Tong Chen, Ben Zhou, Wenhao Yu, Dian Yu, Baolin Peng, Hongwei Wang, Dan Roth, and Dong Yu. 2024b. Sub-sentence encoder: Contrastive learning of propositional semantic representations. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1596–1609, Mexico City, Mexico. Association for Computational Linguistics.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024c. Dense X retrieval: What retrieval granularity should we use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15159–15177, Miami, Florida, USA. Association for Computational Linguistics.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2025. From local to global: A graph rag approach to query-focused summarization. *Preprint*, arXiv:2404.16130.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. A survey on llm-as-a-judge. *Preprint*, arXiv:2411.15594.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025. LightRAG: Simple and fast retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10746–10761, Suzhou, China. Association for Computational Linguistics.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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. *Preprint*, arXiv:2502.14802.

Haoyu Han, Li Ma, Harry Shomer, Yu Wang, Yongjia Lei, Kai Guo, Zhigang Hua, Bo Long, Hui Liu, Charu C. Aggarwal, and Jiliang Tang. 2025. Rag vs. graphrag: A systematic evaluation and key insights. *Preprint*, arXiv:2502.11371.

Taher H. Haveliwala. 2002. Topic-sensitive pagerank. In *Proceedings of the 11th International Conference on World Wide Web*, WWW '02, page 517–526, New York, NY, USA. Association for Computing Machinery.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2).

Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. WiCE: Real-world entailment

for claims in Wikipedia. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7561–7583, Singapore. Association for Computational Linguistics.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

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

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

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. NV-embed: Improved techniques for training LLMs as generalist embedding models. In *The Thirteenth International Conference on Learning Representations*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Haoran Luo, Haihong E, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin Kuang, Meina Song, Yifan Zhu, and Anh Tuan Luu. 2025. HypergraphRAG: Retrieval-augmented generation via hypergraph-structured knowledge representation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

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 *The Thirteenth International Conference on Learning Representations*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Costas Mavromatis and George Karypis. 2025. GNN-RAG: Graph neural retrieval for efficient large language model reasoning on knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16682–16699, Vienna, Austria. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Konstantinos Papakostas and Irene Papadopoulou. 2023. Model analysis & evaluation for ambiguous question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4570–4580, Toronto, Canada. Association for Computational Linguistics.

Pruthvi Patel, Swaroop Mishra, Mihir Parmar, and Chitta Baral. 2022. Is a question decomposition unit all we need? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4553–4569, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. 2025. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM Web Conference 2025 (TheWebConf 2025)*, Sydney, Australia. ACM.

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

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning.

2024. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274, Singapore. Association for Computational Linguistics.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel 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 Representation Learning*, volume 2024, pages 3868–3898.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

Jingjin Wang and Jiawei Han. 2025. PropRAG: Guiding retrieval with beam search over proposition paths. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6223–6238, Suzhou, China. Association for Computational Linguistics.

Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. 2025. When to use graphs in rag: A comprehensive analysis for graph retrieval-augmented generation. *Preprint*, arXiv:2506.05690.

Fangyuan Xu, Junyi Jessy Li, and Eunsol Choi. 2022. How do we answer complex questions: Discourse structure of long-form answers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3556–3572, Dublin, Ireland. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Junnan Dong, Hao Chen, Yi Chang, and Xiao Huang. 2025. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*.

A Knowledge Base Extraction

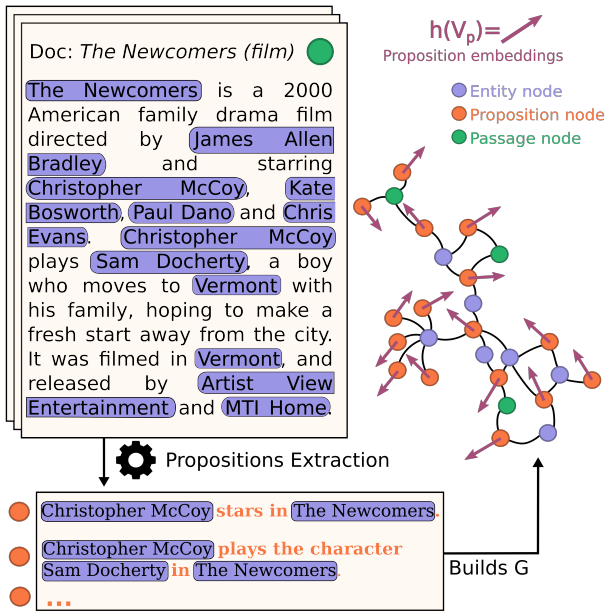

Figure 5: Knowledge base extraction process. Propositions and entities are extracted from input passages and populate the graph. Entity embeddings (used for synonym resolution) are omitted for clarity reasons.

An illustration of the knowledge base extraction is presented in Figure 5. The prompt strategy used for entities and propositions extraction is presented in Figure 6. For synonym reconciliation, given an encoder h, two entities e and e' are considered synonymous if $\text{cosine}\big(h(e), h(e')\big) \geq \theta$ (default 0.9).

We also report statistics for the knowledge base construction (indexing) stage of our approach. Table 2 summarizes the resulting graph sizes, including counts of passages, propositions, and entities, as well as the total number of edges. The overall indexing cost, using the MusiQue corpus as a reference, is also provided and compared against baseline systems in Table 3.

B Supplementary Methods

B.1 Query Aware Transition: an illustrative example

Figure 7 provides an illustrative and intuitive representation behind the construction of the query-aware transition matrix M. The panel $P1$ shows a hypothetical input subgraph G^*, with four annotated nodes (A, B, C and D) that serve as reference points for the next panels. $P2$ describes the proposition-projected graph associated with T_s, as a *propositions to propositions* graph. For instance, nodes around D are all connected to the same entity in $P1$, creating a clique in the resulting projected graph in $P2$. The width of the arrows is proportional to the transition probability between two nodes, according to their connectivity (through entities and passages) in the original graph. $P3$ describes the second component of M: T_n. In this graph, the attraction of a node relative to its neighbors, indicated by the width of the arrow, is proportional to its similarity to the query (default: cosine similarity). Nodes A, B, C and D become attractive as their embeddings are similar to the query compared to other nodes (e.g., in the neighborhood of D). In $P4$, we exemplify the results of running a PPR using A as the starting node and following the built M transition matrix, balancing the transitions between T_s and T_n. In this example, proposition nodes like D or B would be among the top-ranked nodes.

B.2 `Local` mode

Algorithm 1 describes the query process in `Local` mode. An illustrative example is also provided in Figure 8.

B.3 `Global` mode: Compute queries

The approach for query vector refinement is inspired by the Rocchio algorithm (Rocchio Jr, 1971), which applies relevance feedback to refine a query by weighting vectors of relevant and non-relevant documents.

We adapt this principle to compute a refined query vector q_i for each newly collected proposition node $u_i \in s_{\text{pool}}$. The refinement process combines: the directions that lead to u_i in the previous walks, the semantic representation of u_i itself, and, the directions that were pruned.

Let $\mathcal{C}_i$ denote the set of partitions where u_i was identified. For a given partition $c \in \mathcal{C}_i$, let $q_{k^*}^{(c)}$ be the query vector of the walker that most likely reached u_i, where $k^* = \arg\max_k \pi_k^{(c)}(u_i)$. Furthermore, let $\bar{S}_{\text{new}}^{(c)} = S_{\text{new}}^{(c)} \setminus s_{\text{new}}^{(c)}$ be the set of candidate nodes that were pruned by the Select procedure in that partition and $\bar{h}\big(\bar{S}_{\text{new}}^{(c)}\big)$ their averaged embedding.

The query vector for u_i is then given by:

$$
\begin{aligned}
q_i &= \alpha\, q_i^o + \beta q_i^+ - \gamma q_i^- \quad \text{where,} \\
q_i^o &= \frac{1}{|\mathcal{C}_i|} \sum_{c \in \mathcal{C}_i} q_{k^*}^{(c)}, \\
q_i^+ &= h(u_i), \\
q_i^- &= \frac{1}{|\mathcal{C}_i|} \sum_{c \in \mathcal{C}_i} \bar{h}\big(\bar{S}_{\text{new}}^{(c)}\big)
\end{aligned}
\tag{11}
$$

The coefficients $\alpha, \beta,$ and γ are positive weights that modulate the influence of the initial, positive, and negative feedback components, respectively.

B.4 `Global` mode: Collecting anchor nodes

Algorithm 2 describes the iterative exploration and collection process that builds the set of anchor propositions, before community extraction in `Global` mode.

B.5 `Global` mode: Greedy community selection with budget

A complete description of the greedy procedure is presented in Algorithm 3. Candidate communities $c \in C$ are pre-fitlered by size (number of nodes): $10 \leq |c| \leq 150$ and the budget B is fixed to 8000 in the experiments.

C Abstracts Questions: Protocol and Examples

We follow the procedure described by LightRAG authors[3]. To emulate a large variety of potential queries, the LLM is first instructed to generate 5 potential users with 5 related tasks for each, given a summary of the corpus. For each task, 5 questions

[3] https://github.com/HKUDS/LightRAG

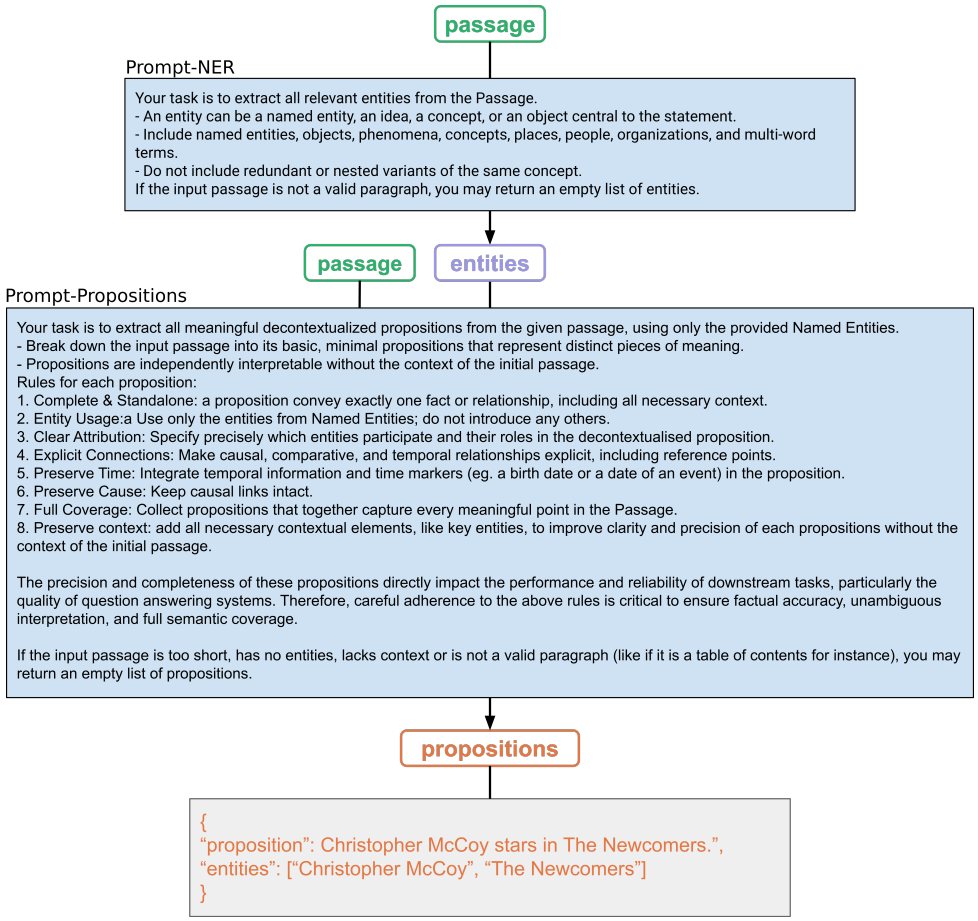

Figure 6: Prompts for Named Entity Recognition and Propositions Extraction. First, named entities are extracted from the passage (`Prompt-NER`). Then, using the previously extracted entities and the original passage, propositions are extracted with `Prompt-Propositions`. Propositions are returned with their associated entities.

	MusiQue	HotPotQA	PopQA	Agriculture	CS	Legal	GB-Medical	GB-Novel
# passages	11,704	9,959	9,101	9,055	7,337	16,169	883	2,400
# propositions	83,247	77,409	73023	9,2840	58,322	84,134	8,442	37,868
# entities	82,721	82,909	79,783	62,341	31,108	35,732	3,955	27,071
# edges	350,436	333,799	309,812	613,688	320,440	786,049	49,863	14,9731

Table 2: Number of nodes (passages, propositions and entities) and edges in the graph associated with each corpora used in our experiments.

	ToPG	LightRAG†	GraphRAG†	HippoRAG 2†
Prompts	70.5M	68.5M	115.5M	9.2M
Completion	11.9M	12.3M	36.1M	3.0M

Table 3: Token usage comparison (prompt and completion) at indexing time for the baselines on the MuSiQue corpus (11,656 passages for 1.3M tokens). †Values reported from Gutiérrez et al. (2025).

are generated that require a high-level understanding of the corpus. Below in Figure 9 is a subset of questions generated from the Agriculture corpus containing textbooks on beekeeping.

D Hyperparameters Evaluation

Figure 10 reports the performance of different combinations of PPR damping factors and λ values on the MusiQue dataset, using the `Local` mode with `max-iter=3`. Across both damping settings (0.5 and 0.85), we observe only minor variation in EM

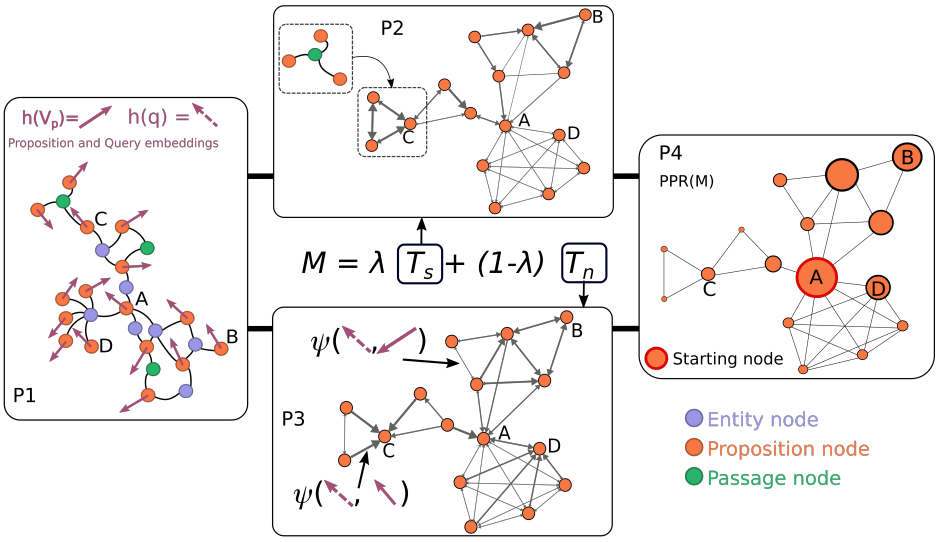

Figure 7: Illustrative example of the construction and realization of the Query Aware Transitions matrix M on an hypothetical subgraph G^*. $P1$ shows the subgraph G^* with 4 landmark nodes A, B, C and D. $P2$ and $P3$ describe the two components of M: T_s and T_n. The width of the arrows is proportional to the transition probability between nodes. $P4$ illustrates the ranking obtained from the stationary distribution π of probabilities with the PPR using M. The larger the node the greater the final probability and rank.

and F1, indicating that the restart probability has limited influence on retrieval quality in these settings.

In contrast, λ, which controls the relative contribution of semantic transitions (T_n) versus structural transitions (T_s), has a pronounced impact. Performance degrades as λ goes to 1 and the semantic component is canceled. This highlights the importance of T_n in suppressing semantically irrelevant paths when neighbors are unrelated to the query.

These observations motivated our choice of default hyperparameters: a damping factor of 0.85 and a balanced $\lambda = 0.5$.

E Dataset and Baseline Details

E.1 Dataset details

Simple and Complex QA datasets Subsets of MusiQue (`CC-By-4.0` License), HotPotQA (`CC-By-4.0` License) and PopQA (`MIT` License) have been extracted from the repository provided by Gutiérrez et al. (2025)[4]. Each corpus is composed of 1,000 questions that require retrieval over one or several passages originating from Wikipedia. Additional details can also be found in Table 2.

GraphRAG-Benchmark Corpora We evaluate on the two corpora of GraphRAG-Benchmark[5]. The Medical corpus (NCCN Guidelines) integrates data from the National Comprehensive Cancer Network (NCCN) clinical guidelines, covering diagnosis criteria, treatment protocols, and drug interactions. Additionally, the Novel corpus (Project Gutenberg) is a curated collection of pre-20th-century novels from the Project Gutenberg library. These texts exhibit complex narrative and temporal relationships. Finally, we use the same Answer Accuracy metric as used in the benchmark (see Xiang et al. (2025)), which combines semantic similarity with statement-level fact checking.

UltraDomain Corpora Abstract QA uses three specialized corpora from the UltraDomain corpus[6], with sizes specified in Table 4.

Abstract QA (LLM-as-a-Judge) Following Guo et al. (2025), abstract queries are evaluated using LLM-as-a-Judge across four criteria:

- **Comprehensiveness:** How much detail does the answer provide to cover all aspects and details of the question?
- **Diversity:** How varied and rich is the an-

[4] `https://huggingface.co/datasets/osunlp/HippoRAG_2`

[5] `MIT` License: `https://github.com/GraphRAG-Bench/GraphRAG-Benchmark`

[6] Apache 2.0 License: `https://huggingface.co/datasets/TommyChien/UltraDomain`

Algorithm 1 Local mode with Suggestion-Selection cycles

Require: Graph $G = (V, E)$, initial query q_{start}, parameter `max-iter`

Initialize $Q \leftarrow \{q_{\text{start}}\}$
$S_0 \leftarrow \text{SuggestNaive}(q_{\text{start}}, \emptyset)$ ▷ Initial seeding in the graph
$s_{\text{pool}} \leftarrow \text{Select}(q_{\text{start}}, S_0)$
$s_{\text{pool-new}} \leftarrow s_{\text{pool}}$
while `iteration` $<$ `max-iter` **do**
  for all $q \in Q$ **do** ▷ Gather propositions for all questions in $s_{\text{pool-new}}$
    $S_{\text{new}} \leftarrow \text{SuggestLocal}(q, s_{\text{pool}})$ ▷ Suggestions seeded on s_{pool} and biased toward q
    $s_{\text{new}} \leftarrow \text{Select}(q, S_{\text{new}})$
    $s_{\text{pool-new}} \leftarrow s_{\text{pool-new}} \cup s_{\text{new}}$
  end for
  $s_{\text{loc}} \leftarrow s_{\text{loc}} \cup s_{\text{pool-new}}$ ▷ Complete s_{loc}
  $s_{\text{pool}} \leftarrow s_{\text{pool-new}}$
  $s_{\text{pool-new}} \leftarrow \emptyset$
  if $\text{PROMPT}_{\text{Eval}}(q_{\text{start}}, s_{\text{loc}})$ returns an answer **then**
    return answer
  else
    $Q \leftarrow \text{PROMPT}_{\text{NextQ}}(q_{\text{start}}, s_{\text{loc}})$ ▷ Evaluate with s_{loc}
  end if
  `iteration` $++$
end while
return failure to determine answer

Corpus	Content Focus	Size (Tokens)
Agriculture	Beekeeping, agricultural policy, farmers, diseases and pests	1.9M
Computer Science (CS)	Machine learning, data processing	2.0M
Legal	Corporate finance, regulatory compliance, finance	4.7M

Table 4: Details and size metrics for the UltraDomain corpus used in Abstract QA evaluation.

swer in providing different perspectives and insights on the question?

- **Empowerment:** How well does the answer help the reader understand and make informed judgments about the topic?

- **Overall:** The final aggregate score combining the three criteria.

We used `Gemma-3-27B` (Team et al., 2025) as the LLM during the evaluation. `Gemma-3-27B` is licensed under the *Gemma Terms of Use*[7]. Details on the prompts can be found on the GitHub repository[8].

[7] https://ai.google.dev/gemma/terms

[8] The link will be made publicly available following the anonymity period.

E.2 Baseline details

The configurations for all baseline models (HippoRAG 2, LightRAG, GraphRAG, and ToPG) are detailed in Table 5. On the granularity level, HippoRAG 2, LightRAG, GraphRAG operate with passage-level context. LightRAG and GraphRAG additionally augment context with auxiliary KG elements (entities/relations).

For a fair comparison on the Abstract QA task, a domain-specific set of topic-related entities was defined during the indexing stage. These entities, used by GraphRAG and LightRAG, are grouped by domain:

- **Agriculture:** organization, geo, event, agriculture, economic, environment.

- **Computer Science (CS):** organization, technology, software, metric, mathematics, hardware, computer_science, networking.

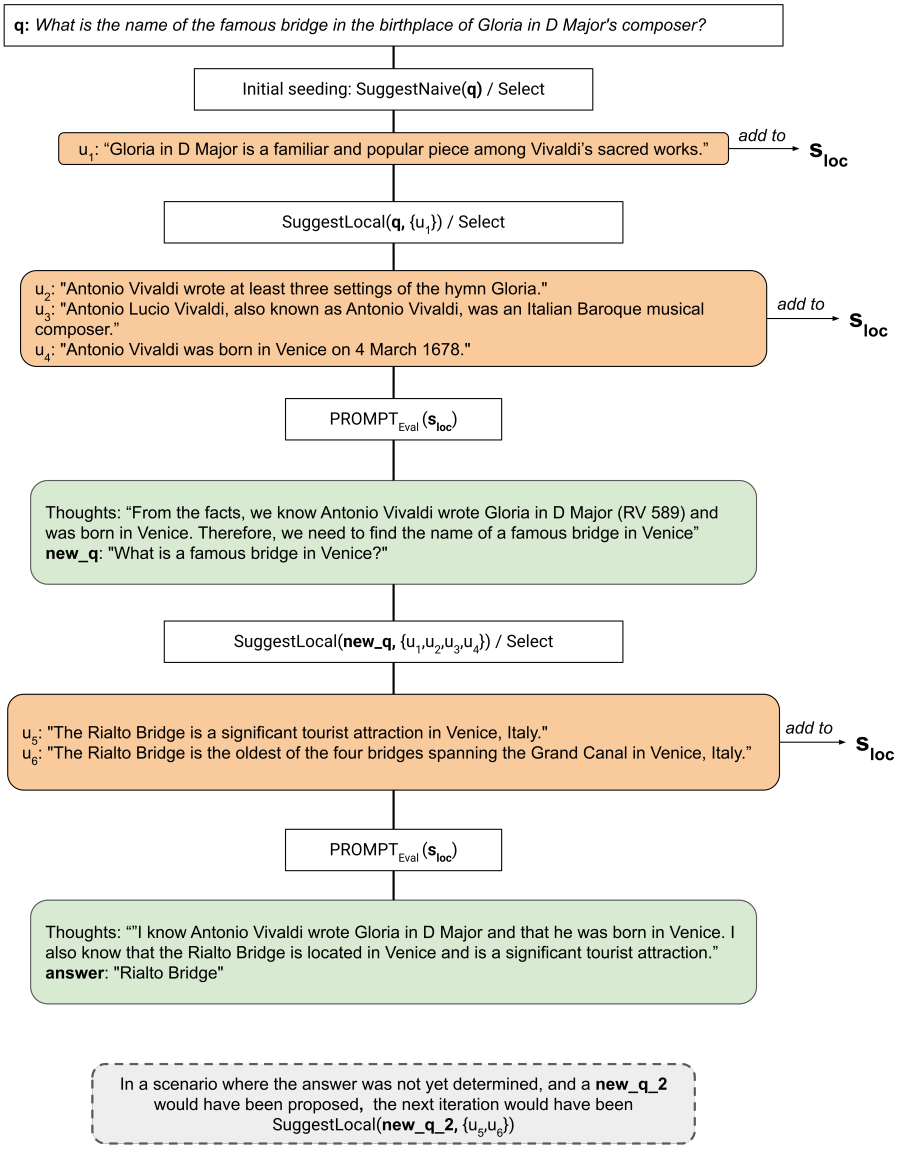

Figure 8: Illustrative example of the `Local` mode.

```
### User 1: Hobbyist Beekeeper
- Task 1: Understanding Beekeeping Basics
  - Question 1: What are the key steps a beginner should consider before starting beekeeping?
  - Question 2: Why is the commitment level crucial for successful beekeeping?
  - Question 3: What common mistakes do new beekeepers make that can lead to failure?
  - Question 4: How does beekeeping activity vary by region and season?
  - Question 5: How important is networking with other beekeepers and organizations?
- Task 2: Managing Beehive Health
  - Question 1: What impact do environmental factors have on hive health?
  - Question 2: How can a beekeeper effectively control mite populations?
...
### User 2: Small-Scale Market Farmer
- Task 1: Planning Farm Production
  - Question 1: What are the major factors to consider when planning crop production?
....
```

Figure 9: Example of abstract questions generated for the Agriculture Corpora

- **Legal:** organization, geo, legal, regulation, financial, asset, risk, law, financial_instrument.

We empirically found that for the Simple/Complex QA tasks, both LightRAG and GraphRAG performed optimally using their local

Algorithm 2 Anchors selection via Iterative Suggestion-Selection (`Global` mode)

Require: Graph G, query q_{start}, breadth m, `max-iter`, minimum facts `min_facts`

$Q \leftarrow \text{PROMPT}_{\text{decompose}}(q_{\text{start}}, m)$ ▷ Decompose initial query into m sub-queries
$s_{\text{glb}} \leftarrow \emptyset$
$s_{\text{pool}} \leftarrow \emptyset$
`iteration` $\leftarrow 0$
for each $q \in Q$ **do** ▷ Initialize s_{pool} with the m questions
    $S_0 \leftarrow \text{SuggestNaive}(q, \emptyset)$
    $s_{\text{pool}} \leftarrow s_{\text{pool}} \cup \text{Select}(q, S_0)$
end for
$s_{\text{glb}} \leftarrow s_{\text{pool}}$
while $|s_{\text{glb}}| <$ `min_facts` **and** `iteration` $<$ `max-iter` **do**
    $s_{\text{pool-new}} \leftarrow \emptyset$
    $\mathbf{q}_{\text{pool}} \leftarrow \text{ComputeQueries}(s_{\text{pool}})$ ▷ Compute queries for the new selected nodes
    for each partition s_{part} in $\text{Partition}(s_{\text{pool}}, m)$ **do** ▷ Each partition is explored independently
        $S_{\text{new}} \leftarrow \text{Suggest}_{\text{Global}}(\mathbf{q}_{\text{part}}, G, s_{\text{part}})$
        $s_{\text{new}} \leftarrow \text{Select}(q_{\text{start}}, S_{\text{new}})$
        $s_{\text{pool-new}} \leftarrow s_{\text{pool-new}} \cup s_{\text{new}}$
    end for
    $s_{\text{glb}} \leftarrow s_{\text{glb}} \cup s_{\text{pool-new}}$ ▷ Complete the global s_{glb} and prepare the next seeds (s_{pool})
    $s_{\text{pool}} \leftarrow s_{\text{pool-new}}$
    `iteration` $++$
end while
return s_{glb} ▷ Final pool of collected propositions (anchors)

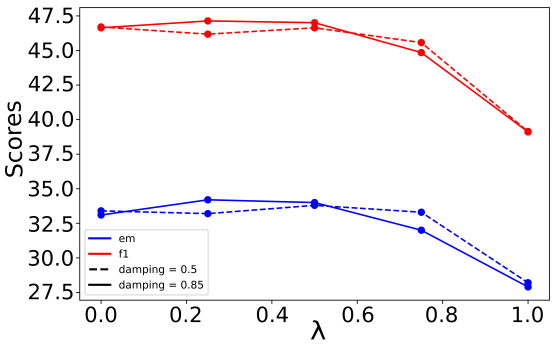

Figure 10: Impact of the damping factor and λ parameters on model performance measured by F1-score and Exact Match (EM) on MusiQue.

search mode with a smaller `top_k` $= 5$ compared to standard default settings (60 for LightRAG and 10 for GraphRAG). We hypothesized that the resulting large context ($\geq$ 8k tokens) is detrimental for accurate factual QA with the used LLM.

To establish a strong baseline for comparison against our proposed strategy, the global mode of GraphRAG was configured with `community_level` $= 2$. While this choice significantly increased the granularity of community search, it also comes at a substantial computational cost (consuming on average 79k and 2.1M tokens for Completion and Prompts, per query.

F Example Abstract QA and evaluation

An example of evaluation with LLM-as-a-judge on the Agriculture corpora considering the 4 criteria (Comprehensiveness, Diversity, Empowerment and Overall) is provided in Table 6.

Model	Task	Parameter	Value
HippoRAG 2	**Simple/Complex QA**	top_k	5
GraphRAG	**Simple/Complex QA**	mode	local
		max_context_tokens	8000
		text_unit_prop	0.5
		community_prop	0.25
		top_k_mapped_entities	5
		top_k_relationships	5
	Abstract QA	mode	local
		community_level	2
		use_community_summary	True
		min_community_rank	0
		max_tokens	12000
LightRAG	**Simple/Complex QA**	mode	local
		top_k	5
		chunk_top_k	5
		max_total_tokens	6000
	Abstract QA	mode	hybrid
		TOP_K	40
		CHUNK_TOP_K	10
		MAX_TOTAL_TOKENS	32000
ToPG	**Simple/Complex QA**	λ	0.5
		damping d	0.85
		cosine_threshold θ	0.4
		subgraph $G*$ max size l	500
		temperature τ	0.1
		top_k	20
	Abstract QA	top_k	20
		breadth m	10
		node budget B	8000
		min_community_size	10
		max_community_size	150
		max_tokens_report	8000
		passage_token_limit	500
		max_tokens_community_chunks	8000

Table 5: Parameter settings for the RAG baselines (HippoRAG 2, LightRAG, GraphRAG, and ToPG), across the evaluation tasks (Simple QA, Complex QA, and Abstract QA). Additional configuration parameters defined for ToPG include: `max_tokens_report` (the maximum token limit available for the final response context), `passage_token_limit` sets the limit for tokens used from community passages, within the `max_tokens_community_chunks` available for intermediary answers. For a comprehensive description of other baseline-specific parameters, please refer to their respective documentation.

Algorithm 3 Greedy Budgeted Communities Extraction

Require: S: anchor nodes, B: budget limit, $\mathcal{C}$: candidate communities

$b \leftarrow 0$, $S' \leftarrow \emptyset$, $C' \leftarrow \emptyset$ ▷ budget used, nodes covered, communities selected

while $S' \neq S$ and $b < B$ **do**

  $c^* \leftarrow \arg\max_{c \in \mathcal{C}} \frac{|\text{nodes}(c) \setminus S'|}{\text{size}(c)}$ ▷ best coverage / size ratio

  $C' \leftarrow C' \cup \{c^*\}$ ▷ Update candidates, nodes covered and budget

  $S' \leftarrow S' \cup \text{nodes}(c^*)$

  $b \leftarrow b + \text{size}(c^*)$

end while

return C'

Query	How does soil health influence overall farm productivity?
GraphRAG	**The Central Role of Soil Health in Farm Productivity** Soil health is fundamentally ... **Key Components of Healthy Soil** Several key components contribute to overall soil health. **Organic matter** is paramount, improving soil structure.... **Agricultural Practices & Soil Health** Numerous agricultural practices directly impact soil health. **No-till farming** significantly improves soil health ... **The Impact of Soil Health on Yield & Sustainability** Maintaining soil health is crucial for maximizing agricultural output... **Soil Health & Economic Viability** Soil health is intrinsically linked to agricultural economics... In conclusion, soil health is not merely a component ...
ToPG	**Soil Health and Farm Productivity: A Comprehensive Report** Improving soil health is increasingly recognized as central ... **The Central Role of Organic Matter** Organic matter is paramount to soil health and fertility... **Practices for Enhancing Soil Health** Several key practices contribute to improved soil health... **Soil Biology and Nutrient Availability** Healthy soil directly impacts nutrient availability for plants... **Water Management and Soil Structure** Soil health significantly impacts water retention and availability... **Long-Term Sustainability and Resilience** Improving soil health is not a complex process and can lead to cost-effective ways to feed the world and restore land... In conclusion, prioritizing soil health is essential for maximizing farm productivity, ensuring long-term sustainability, and building resilience.
LLM Decision: Comprehensiveness	**Winner: Answer 2** **Explanation:** Both answers cover the core components of soil health and its impact on farm productivity. [...] Answer 2 also acknowledges challenges and considerations, like the adjustment period for transitioning to new practices, which adds to its completeness.
LLM Decision: Diversity	**Winner: Answer 2** **Explanation:** While both answers touch on various aspects of soil health, Answer 2 demonstrates greater diversity in its perspectives. [...] Answer 2 also acknowledges the potential downsides of conventional practices, offering a more balanced view.
LLM Decision: Empowerment	**Winner: Answer 2** **Explanation:** Answer 2 is more empowering because it provides concrete examples and quantifiable data that allow the reader to understand the *magnitude* of the benefits associated with improved soil health. [...] It states that soil health is important, but doesn't provide the same level of evidence to support that claim in a way that empowers the reader to act.
LLM Decision: Overall Winner	**Winner: Answer 2** **Explanation:** Answer 2 is the overall winner because it excels in all three criteria. [...] While Answer 1 provides a solid overview of the topic, Answer 2 goes further in explaining the *why* and *how* of soil health, making it a more valuable and insightful resource for anyone seeking to understand this critical aspect of agricultural productivity.

Table 6: Example of an evaluation of Abstract QA using LLM-as-a-judge, comparsing answers from GraphRAG and ToPG. The model used is `Gemma-3-27B`.