# RESCOPERAG: EVIDENCE-GUIDED RETRIEVAL SCOPE ADAPTATION FOR MULTI-STEP REASONING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Retrieval-augmented generation (RAG) relies on effective retrieval to identify relevant evidence from large and complex corpora. Existing approaches improve evidence acquisition through diverse retrieval strategies, but typically fix the retrieval scope from the initial query. This limitation is particularly pronounced in multi-step reasoning, where intermediate evidence can reveal new targets and evolve the information need beyond the initial scope. We propose ReScopeRAG, an iterative graph retrieval framework for multi-step reasoning. It dynamically adapts the retrieval scope as accumulated evidence reveals new targets. The framework first models dependencies among sub-queries to organize the reasoning process. A Master–Sub-Agent architecture then coordinates evidence acquisition and scope adaptation, enabling the system to progressively shift the retrieval scope as new targets emerge. Experiments on multiple general multi-hop question answering datasets show that ReScopeRAG achieves superior performance compared with conventional retrieval, graph-based, and agentic RAG baselines.

## 1 INTRODUCTION

Retrieval-augmented generation (RAG) has become a central paradigm for grounding large language models (LLMs) in external knowledge (Lewis et al., 2020). The core challenge of retrieval is how to find truly relevant information from large and complex corpora. Existing RAG methods have approached this challenge from several directions. Traditional retrieval methods identify query-relevant passages using sparse, dense, or hybrid representations (Robertson & Walker, 1994; Karpukhin et al., 2020; Izacard et al., 2021; Chen et al., 2024). Structured retrieval methods organize corpus information through entities, relations, or hierarchies, enabling retrieval to exploit the structural relationships among distributed evidence. (Sarthi et al., 2024; Edge et al., 2024; Guo et al., 2025; Gutiérrez et al., 2025; Chen et al., 2026). More recent approaches further make retrieval iterative or agent-controlled, employing query decomposition or reformulation, retrieval triggering, and agent planning to acquire evidence progressively (Trivedi et al., 2023; Jiang et al., 2023; Asai et al., 2024; Nguyen et al., 2025; Maragheh et al., 2025).

Despite these advances, an important aspect of retrieval remains underexplored: how should the retrieval scope be determined? The retrieval scope determines which evidence can be accessed and thus sets an upper bound on retrieval and generation quality. An overly broad scope introduces irrelevant noise, while an overly narrow scope compromises the completeness of the retrieved evidence. Effective retrieval therefore requires balancing evidence coverage with search focus. As illustrated in Figure 1 a) and b), existing methods typically determine a static retrieval scope directly from the initial query (Karpukhin et al., 2020; Izacard et al., 2021; Sarthi et al., 2024; Edge et al., 2024; Guo et al., 2025; Maragheh et al., 2025). Subsequent retrieval may modify the query, search direction, or retrieval strategy, but usually continues to operate within the scope defined at initialization, whether it is restricted to a specific community, neighborhood, or path, or spans the entire corpus.

This limitation is particularly pronounced in multi-hop reasoning. Multi-hop queries often involve information needs that are only partially observable at the beginning. During reasoning, intermediate evidence may reveal a previously implicit target entity, disambiguate a relation, or introduce a new sub-goal. As the information need evolves, evidence that becomes relevant after one reasoning step may fall outside the retrieval scope selected at initialization. Thus, an effective retrieval process



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

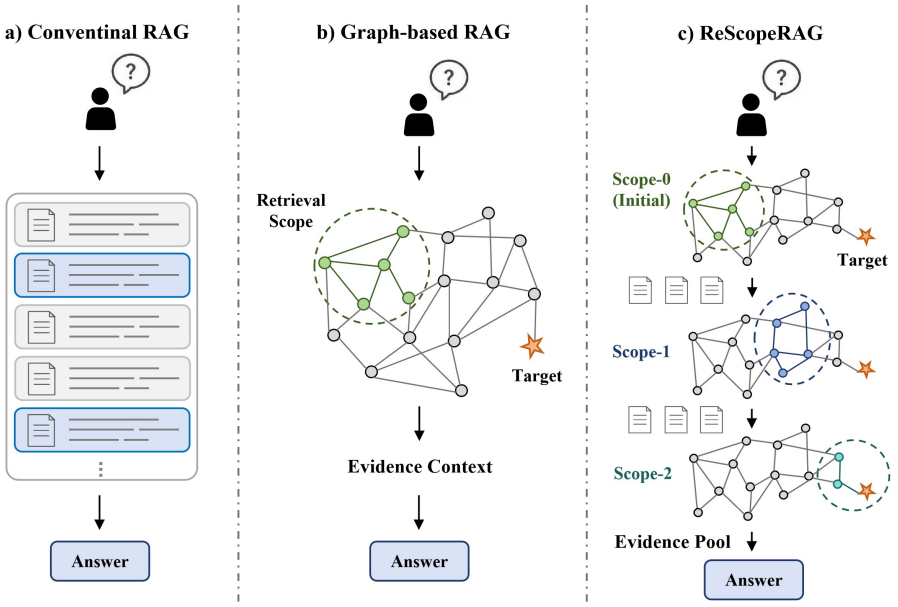


Figure 1: Comparison of retrieval paradigms. Existing methods typically determine a static retrieval scope from the initial query. ReScopeRAG treats the retrieval scope as an evolving state and adjusts it across reasoning steps as evidence reveals new targets and information needs.

requires an explicit mechanism to adapt the retrieval scope as new targets and information needs emerge.

A graph-based representation provides a natural basis for dynamically defining the retrieval scope. By connecting distributed information through semantic and relational dependencies, a graph organizes related evidence into coherent knowledge regions rather than isolated text units. These regions provide structured units for defining and adjusting the retrieval scope.

So, we propose ReScopeRAG, an evidence-driven graph retrieval framework that dynamically adjusts the retrieval scope for multi-step reasoning. Figure 1 c) illustrates the key distinction between ReScopeRAG and existing retrieval paradigms. Existing graph-based retrieval approaches typically select evidence within an initially defined search area. ReScopeRAG instead allows retrieval to move across knowledge regions when evidence indicates a shift in the information need. This transforms graph-based retrieval from one-pass selection to iterative scope adaptation, enabling the graph structure to serve as a dynamic search space.

ReScopeRAG first decomposes the input question into sub-query plans to make evolving retrieval targets explicit. Their dependencies are modeled as a directed acyclic structure, allowing prerequisite evidence to guide downstream search. Retrieval then proceeds iteratively through a Master–Sub-Agent architecture. The Master maintains the global retrieval state, while multiple Sub-Agents conduct localized retrieval within their current regions. A shared Evidence Pool facilitates information exchange between local retrieval and global reasoning. Together, these components form an evidence-driven retrieval process that progressively adjusts the retrieval scope as the information need evolves.

Our contributions are summarized as follows:

- We propose ReScopeRAG, a evidence-guided iterative graph retrieval framework that formulates multi-step retrieval as dynamic retrieval-scope adjustment over knowledge regions. This enables graph-based retrieval to adapt its scope as reasoning evolves.
- We introduce dependency-aware query decomposition and a Master–Sub-Agent retrieval architecture, which together align retrieval with the information flow of multi-step reasoning. This explicitly represents evolving information needs to inform retrieval execution.

- We evaluate ReScopeRAG against conventional retrieval, graph-based RAG, and agentic RAG baselines. Experimental results show that ReScopeRAG achieves superior performance across multi-step reasoning tasks, demonstrating the effectiveness of evidence-driven retrieval-scope adaptation.

The code and data are available at `https://anonymous.4open.science/r/ReScopeRAG-code`.

## 2 Related Work

### 2.1 Knowledge Organization for Retrieval-Augmented Generation

Existing RAG studies organize retrievable knowledge in different forms, ranging from flat text-based representations to structured and graph-based representations.

Flat retrieval treats the corpus as independent text units and relies on lexical or dense semantic matching (Robertson & Walker, 1994; Karpukhin et al., 2020; Guu et al., 2020), with later approaches improving semantic matching and retrieval robustness through late interaction (Khattab & Zaharia, 2020), contrastive learning (Izacard et al., 2021), large-scale text encoders (Ni et al., 2022), and hybrid retrieval representations (Chen et al., 2024).

Structured approaches organize passages into richer retrieval structures, such as hierarchical trees (Sarthi et al., 2024) or complementary semantic and relational indices (Zhang et al., 2025).

Graph-based methods explicitly introduce graph structures into knowledge organization. GraphRAG constructs entity-centric graphs and community summaries for global and local question answering (Edge et al., 2024), while LightRAG employs dual-level retrieval over low- and high-level relational knowledge (Guo et al., 2025). Inspired by human long-term memory, HippoRAG leverages knowledge graphs and Personalized PageRank (PPR) for retrieval, with HippoRAG2 improving passage integration (Gutiérrez et al., 2024; 2025). PathRAG retrieves key relational paths to organize evidence for generation (Chen et al., 2026). Other studies incorporate knowledge graphs, subgraphs, or graph neural representations into RAG pipelines (Zhu et al., 2025; He et al., 2024).

Our work follows the line of graph-based knowledge organization, with a focus on dynamically exploiting knowledge regions in graph structures for multi-hop question answering.

### 2.2 Adaptive Retrieval for Multi-Step Reasoning

Complex questions often require retrieval to proceed across multiple reasoning steps. One direction of work decomposes complex queries into simpler sub-questions. Decomposed Prompting solves complex tasks through modular sub-task prompts (Khot et al., 2022). Least-to-Most Prompting decomposes a difficult problem into simpler subproblems and solves them sequentially (Zhou et al., 2022).

Another direction of work makes retrieval iterative or adaptive. IRCoT interleaves chain-of-thought reasoning with retrieval (Trivedi et al., 2023), while Iter-RetGen alternates retrieval and generation to refine the evidence context across iterations (Shao et al., 2023). Self-RAG trains models to decide when to retrieve and how to use retrieved evidence (Asai et al., 2024). Adaptive-RAG selects retrieval and reasoning strategies according to query complexity (Jeong et al., 2024).

Recent studies also introduce agentic retrieval mechanisms. GraphReader explores graph-structured text representations with an agent (Li et al., 2024). MAIN-RAG and MA-RAG use multiple agents to decompose, retrieve, and integrate information for complex questions (Chang et al., 2025; Nguyen et al., 2025). A-RAG incorporates agent planning or tool use into the retrieval-augmented generation process (Maragheh et al., 2025).

These lines of work suggest that adaptive retrieval for multi-step reasoning provides an opportunity to further exploit the structural benefits of graph-based knowledge organization. ReScopeRAG explores this direction through evidence-driven adjustment of retrieval scopes across knowledge regions.

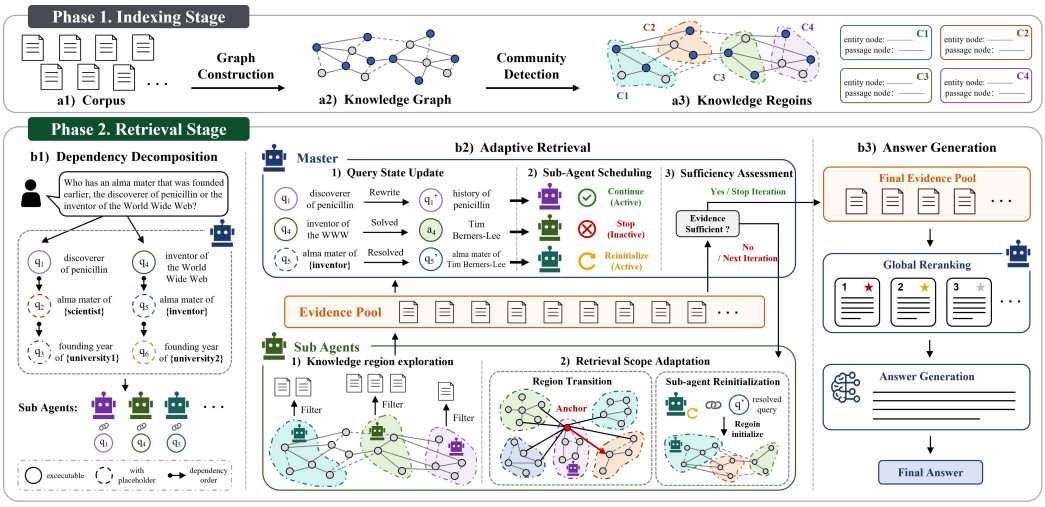


Figure 2: Overall framework of ReScopeRAG. The framework consists of two stages: the Indexing Stage constructs knowledge regions from the corpus, and the Retrieval Stage sequentially performs Dependency Decomposition, Adaptive Retrieval, and Answer Generation.

# 3 METHOD

## 3.1 OVERALL FRAMEWORK

The key challenge addressed by ReScopeRAG is how to dynamically adjust the retrieval scope as multi-step reasoning reveals new retrieval targets. To this end, we propose a retrieval framework built upon an indexing stage. The indexing stage constructs structured knowledge regions from the corpus that serve as a foundation, while our proposed retrieval stage iteratively explores and adjusts retrieval scopes. An overview of the proposed framework is illustrated in Figure 2.

### 3.1.1 INDEXING STAGE

Given a raw corpus, we first construct a knowledge graph following the graph construction pipeline of HippoRAG2 (Gutiérrez et al., 2025). The resulting graph is then partitioned into semantically coherent knowledge communities using the Leiden community detection algorithm (Traag et al., 2019). These communities serve as structured knowledge regions for the adaptive retrieval process.

### 3.1.2 RETRIEVAL STAGE

The adaptive retrieval stage adopts a Master–Sub-Agent architecture, consisting of a Master, multiple Sub-Agents, and an Evidence Pool. The Master maintains the global retrieval state, while Sub-Agents perform localized retrieval within different retrieval scopes. The Evidence Pool supports communication between local retrieval and global reasoning.

Our framework operates as an evidence-driven iterative retrieval loop. At each iteration, the newly collected evidence updates the global retrieval state, which in turn guides the subsequent selection and exploration of retrieval scopes. The process terminates when sufficient evidence is collected or a preset iteration limit is reached.

## 3.2 DEPENDENCY DECOMPOSITION

Complex queries often involve interlinked information, where resolving later parts depends on intermediate results from earlier ones. To go beyond single-step retrieval, we propose a dependency-aware approach. It decomposes a query into a structured execution graph, with explicit dependencies that enforce progressive reasoning. This ensures that downstream retrieval is grounded in validated intermediate knowledge, rather than unguided guesses.

#### 3.2.1 QUERY FORMULATION

Given an input query $Q$, the Master decomposes it into a set of sub query plans $\mathcal{P} = \{p_i\}_{i=0}^{M-1}$. Each plan is represented by four fixed fields,

$$p_i = \{\mathrm{id}_i, \mathrm{text}_i, \mathrm{depends_on}_i, \mathrm{placeholder}_i\},$$

where $\mathrm{id}_i$ is a unique integer identifier starting from 0, $M$ is the max number of sub query that is allowed to decompose, and $\mathrm{text}_i$ is an natural-language question targeting a single factual attribute or relation. The field $\mathrm{depends_on}_i$ is either empty for an independent plan or contains the identifier of at most one prerequisite plan, allowing independent branches to be executed in parallel while dependent plans form directed reasoning chains. The $\mathrm{placeholder}_i$ field is null for independent plans; otherwise, it specifies the placeholder slot in $\mathrm{text}_i$ and a semantic hint describing the entity to be resolved. Together, these plans form a directed acyclic dependency structure that determines the information flow among retrieval tasks.

During retrieval, evidence acquired from a prerequisite plan is leveraged to resolve the placeholder within its dependent plan. The instantiated sub-query then serves as an updated query for subsequent retrieval, progressively refining the retrieval target along the dependency chain.

#### 3.2.2 SUB-AGENT INITIALIZATION

After query formulation, the Master initializes one Sub-Agent for each sub-query. Each Sub-Agent maintains an independent retrieval state at iteration $t$,

$$S_i^{(t)} = \left(g_i^{(t)}, c_i^{(t)}, \mathcal{V}_i^{(t)}, \mathcal{A}_i^{(t)}, h_i^{(t)}, \alpha_i^{(t)}\right),$$

where $g_i^{(t)}$ denotes the current query gap, $c_i^{(t)}$ is the current knowledge region, $\mathcal{V}_i^{(t)}$ records the visited regions, $\mathcal{A}_i^{(t)}$ contains the navigation anchors, $h_i^{(t)}$ denotes the number of retrieval steps, and $\alpha_i^{(t)}$ indicates whether the agent remains active.

At initialization, the query gap is set to the assigned sub-query, and all Sub-Agents are activated, i.e., $g_i^{(0)} = \mathrm{text}_i$, $\alpha_i^{(0)} = 1$. To determine the initial retrieval scope, the Master extracts a representative entity from the sub-query and matches it to the nearest graph entity in the embedding space. The matched entity is used as the initial navigation anchor, and its associated knowledge region is assigned as $c_i^{(0)}$.

### 3.3 ADAPTIVE RETRIEVAL

As complex reasoning unfolds, newly acquired evidence reveals previously implicit target entities and information needs, causing the relevant retrieval scope to evolve across iterations. We therefore formulate online retrieval as an evidence-driven iterative process consisting of three successive steps: Knowledge Region Exploration, Master Reasoning and Scheduling, and Retrieval Scope Adaptation.

#### 3.3.1 KNOWLEDGE REGION EXPLORATION

At iteration $t$, each active Sub-Agent $(SA_i)$ performs localized retrieval within the knowledge region $c_i^{(t)}$ specified by its current retrieval scope. Guided by the current query gap $g_i^{(t)}$, Personalized PageRank (PPR) retrieves the top-ranked candidate passages:

$$C_i^{(t)} = \mathrm{TopK}\Big(\mathrm{PPR}\Big(G[c_i^{(t)}], g_i^{(t)}\Big)\Big),$$

where $G[c_i^{(t)}]$ denotes the subgraph associated with the current knowledge region.

The Sub-Agent $(SA_i)$ filters the candidate passages to obtain the relevant evidence set,

$$E_i^{(t)} = \mathrm{Filter}\Big(C_i^{(t)}, g_i^{(t)}\Big),$$

which is then added to the Evidence Pool. In addition, the top-ranked graph entities are preserved as navigation anchors for subsequent retrieval-scope adjustment.

#### 3.3.2 REASONING AND SCHEDULING

After all active Sub-Agents complete exploration at iteration $t$, the Master analyzes the accumulated evidence $E^{(t)}$ in the Evidence Pool and updates each query state according to three cases: (i) a sub-query supported by the current evidence is marked as solved; (ii) an unsolved sub-query without a placeholder is rewritten to target the remaining information need; and (iii) a resolvable placeholder is replaced with the identified entity, yielding an updated query gap $g_i^{(t+1)}$.

Based on the updated query state, the Master assigns each Sub-Agent a scheduling decision,

$$d_i^{(t)} \in \{\text{continue}, \text{stop}, \text{reinitialize}\}.$$

A solved sub-query leads to deactivation, $\alpha_i^{(t+1)} = 0$, while a rewritten query remains active for continued exploration. When placeholder resolution substantially changes the retrieval target, the corresponding Sub-Agent is marked for reinitialization.

Also, the Master assesses whether the accumulated evidence is sufficient to answer the original query. If sufficient, the retrieval process terminates and proceeds to answer generation; otherwise, the updated query states and scheduling decisions are used for retrieval-scope adaptation.

#### 3.3.3 RETRIEVAL SCOPE ADAPTATION

When retrieval continues, each active Sub-Agent adjusts its retrieval scope according to its scheduling decision. Region transition serves as the primary mechanism for retrieval-scope adjustment. The retained navigation anchors are used to identify candidate knowledge regions connected to the current region. The Sub-Agent ranks these regions according to the semantic relevance between the updated query gap $g_i^{(t+1)}$ and the most relevant graph nodes within each region. It then probes the candidates sequentially using the exploration procedure described in Section 3.3.1. Probing terminates upon obtaining sufficient local evidence or exhausting the candidate budget, which bounds the number of regions explored within each transition. The selected region becomes the current knowledge region $c_i^{(t+1)}$, after which the visited regions and navigation anchors are updated accordingly.

When placeholder resolution substantially changes the retrieval target, the Sub-Agent reinitializes its retrieval scope following Section 3.2.2, resetting its retrieval state before resuming from the new scope.

### 3.4 ANSWER GENERATION

After retrieval terminates, the Master ranks the accumulated evidence according to its relevance to the original query. The top-$k$ passages are selected to form the final evidence context, based on which the Master produces the final answer.

## 4 EXPERIMENTS

### 4.1 EXPERIMENTAL SETUP

#### 4.1.1 DATASETS

We evaluate ReScopeRAG on three widely used multi-hop question answering datasets: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022). For each dataset, we randomly sample 1,000 questions for evaluation. Following HippoRAG2 (Gutiérrez et al., 2025), the retrieval corpus contains the gold supporting passages and distractor passages associated with the sampled questions.

#### 4.1.2 BASELINES

We compare ReScopeRAG with three groups of baselines: conventional retrieval, graph-based RAG, and agentic RAG.

Table 1: Question answering performance (in %) on three multi-hop QA datasets. Best results are shown in bold and second-best results are underlined.

| Method | 2Wiki | | MuSiQue | | HotpotQA | | Average | |
|---|---|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| ***Conventional Retrieval*** | | | | | | | | |
| BM25 | 41.30 | 42.78 | 17.70 | 22.56 | 57.10 | 64.20 | 38.70 | 43.18 |
| bge-m3 | 56.00 | 57.52 | 35.00 | 42.96 | 69.40 | 77.68 | 53.47 | 59.39 |
| contriever | 46.60 | 48.07 | 26.60 | 33.86 | 62.50 | 70.75 | 45.23 | 50.89 |
| contriever (msmarco) | 55.30 | 57.19 | 37.20 | 45.20 | 64.60 | 72.58 | 52.37 | 58.32 |
| GTR (T5-base) | 53.20 | 54.97 | 30.50 | 37.97 | 62.00 | 69.25 | 48.57 | 54.06 |
| ***Graph-based RAG*** | | | | | | | | |
| GraphRAG | 56.40 | 63.21 | 27.50 | 33.97 | 64.90 | 74.07 | 49.60 | 57.08 |
| LightRAG (local) | 22.90 | 27.58 | 6.90 | 9.02 | 37.40 | 43.48 | 22.40 | 26.69 |
| LightRAG (mix) | 39.10 | 41.61 | 23.40 | 30.24 | 49.90 | 56.70 | 37.47 | 42.85 |
| HippoRAG2 | 57.70 | 59.39 | 35.70 | 42.82 | **73.10** | 79.80 | 55.50 | 60.67 |
| ArchRAG | 46.10 | 53.09 | 19.00 | 30.98 | 50.60 | 60.04 | 38.57 | 48.04 |
| ***Agentic RAG*** | | | | | | | | |
| MA-RAG | **70.80** | **79.35** | 38.20 | 46.11 | 63.20 | 71.66 | 57.40 | 65.71 |
| RAGentA | 60.30 | 64.11 | 39.30 | 47.78 | 71.60 | 79.94 | 57.07 | 63.94 |
| **ReScopeRAG (Ours)** | 63.00 | 65.97 | **44.60** | **52.55** | 72.70 | **80.32** | **60.10** | **66.28** |

Table 2: Retrieval performance (in %) on three multi-hop QA datasets. We report Recall@2 and Recall@5 at the passage level. Best results are shown in bold and second-best results are underlined.

| Method | 2Wiki | | MuSiQue | | HotpotQA | | Average | |
|---|---|---|---|---|---|---|---|---|
| | R@2 | R@5 | R@2 | R@5 | R@2 | R@5 | R@2 | R@5 |
| BM25 | 43.10 | 52.32 | 25.89 | 32.64 | 52.00 | 67.00 | 40.33 | 50.65 |
| bge-m3 | 66.83 | 71.13 | 45.27 | 59.51 | 72.35 | 84.35 | 61.48 | 71.66 |
| contriever | 51.18 | 62.25 | 34.43 | 47.00 | 56.85 | 71.95 | 47.49 | 60.40 |
| contriever (msmarco) | 63.02 | 69.57 | 44.82 | 58.31 | 63.45 | 78.10 | 57.10 | 68.66 |
| GTR (T5-base) | 61.68 | 68.17 | 41.05 | 53.42 | 59.40 | 72.15 | 54.04 | 64.58 |
| HippoRAG2 | 66.53 | 72.95 | 44.44 | 57.84 | 77.30 | **88.60** | 62.76 | 73.13 |
| **ReScopeRAG (Ours)** | **73.02** | **75.65** | **57.68** | **66.67** | **78.15** | 80.90 | **69.62** | **74.41** |

**Conventional Retrieval.** We include BM25 (Robertson & Walker, 1994), BGE-M3 (Chen et al., 2024), Contriever (Izacard et al., 2021) and its MS MARCO fine-tuned variant, and GTR (T5-base) (Ni et al., 2022) as conventional retrieval-based baselines. These methods cover both lexical and dense retrieval paradigms.

**Graph-based RAG.** We compare against GraphRAG (Edge et al., 2024), LightRAG (Guo et al., 2025), HippoRAG2 (Gutiérrez et al., 2025), and ArchRAG (Wang et al., 2026). These methods leverage structured knowledge representations for multi-hop evidence retrieval. For LightRAG, we evaluate both its local and mix retrieval modes.

**Agentic RAG.** We also compare with training-free agentic RAG methods, including MA-RAG (Nguyen et al., 2025) and RAGentA (Besrour et al., 2025), which employ agentic or multi-agent retrieval strategies for iterative evidence acquisition and multi-step question answering.

Table 3: Ablation and retrieval-scope analysis of ReScopeRAG. We report Recall@5 and F1 on the three benchmark datasets.

| Method | 2Wiki | | MuSiQue | | HotpotQA | | Average | |
|---|---|---|---|---|---|---|---|---|
| | R@5 | F1 | R@5 | F1 | R@5 | F1 | R@5 | F1 |
| w/o Dependency Decomposition | 66.35 | 59.42 | 58.63 | 46.77 | 69.95 | 73.80 | 64.98 | 59.99 |
| w/o Scope Adaptation | 17.00 | 24.30 | 20.50 | 19.87 | 25.45 | 40.09 | 20.98 | 28.09 |
| Global Scope | 69.83 | 57.69 | 49.53 | 40.47 | 78.10 | 77.50 | 65.82 | 58.55 |
| **ReScopeRAG (full)** | **75.65** | **65.97** | **66.67** | **52.55** | **80.90** | **80.32** | **74.41** | **66.28** |

#### 4.1.3 Evaluation Metrics

For retrieval performance, we report passage-level Recall@2 and Recall@5. Recall@N is reported only for methods with ranked passage-level outputs.

For QA performance, we report Exact Match (EM) and token-based F1 scores. Since our evaluated methods generate free-form responses, we require each method to additionally output a short answer to ensure a fair comparison.

#### 4.1.4 Implementation Details

All experiments use Qwen3.6-35B-A3B as the backbone LLM and bge-large-en as the embedding model. For baselines, we primarily follow their original implementations and configurations.

For ReScopeRAG, we set the maximum numbers of decomposed sub-queries and online retrieval iterations to 5 and 6, respectively. Each active Sub-Agent contributes at most 3 evidence passages per iteration, and the final answer is generated from top-5 evidence passages. Other implementation details are provided in Appendix B.

### 4.2 Main Results

Table 1 and Table 2 report the QA and retrieval performance of ReScopeRAG compared with conventional retrieval, graph-based, and agentic RAG methods.

For retrieval performance, ReScopeRAG achieves near-best results across all three datasets, obtaining the highest scores on most retrieval metrics. It obtains an average Recall@2 of 69.62%, improving over the strongest graph-based baseline HippoRAG2 by 6.86 points. The improvement is particularly pronounced on MuSiQue, where ReScopeRAG improves Recall@5 for 7.16 points, demonstrating its effectiveness in acquiring evidence for complex multi-step reasoning. These results indicate that dynamically reconfiguring retrieval scopes enables more effective exploration of relevant knowledge regions.

For QA performance, ReScopeRAG achieves competitive results across all datasets. It obtains the best EM and F1 scores on MuSiQue among the evaluated methods, reaching 44.60% EM and 52.55% F1, and achieves comparable performance with agentic RAG approaches on HotpotQA. Although some agentic methods obtain higher average QA scores, they primarily optimize answer-level reasoning through iterative generation, whereas ReScopeRAG focuses on improving the underlying evidence acquisition process. The strong retrieval performance suggests that ReScopeRAG provides a more reliable foundation for multi-hop reasoning.

Overall, compared with graph-based RAG approaches, ReScopeRAG demonstrates the effectiveness of dynamically adapting retrieval scopes over fixed graph-based retrieval strategies. Compared with agentic RAG methods, ReScopeRAG achieves competitive QA performance while maintaining a training-free framework, highlighting its ability to improve multi-hop evidence acquisition without additional model optimization.

Table 4: Statistics of dependency-aware query decomposition across the evaluated datasets.

| Metric | 2Wiki | MuSiQue | HotpotQA |
|---|---|---|---|
| Average Sub-queries | 2.335 | 2.427 | 1.973 |
| Multi-sub-query Ratio (%) | 99.40 | 99.00 | 83.30 |
| Average Placeholders | 0.838 | 1.234 | 0.597 |
| Placeholder Query Ratio (%) | 67.00 | 93.50 | 51.70 |
| Placeholder Sub-query Ratio (%) | 35.89 | 50.84 | 30.26 |
| Average Dependency Depth | 0.669 | 1.264 | 0.636 |
| Placeholder Resolution Rate (%) | 98.33 | 74.47 | 82.08 |

# 5 Discussions

## 5.1 Ablation Study

Table 3 reports the ablation results of ReScopeRAG. Removing either dependency-aware decomposition, which decomposes the original query into interdependent sub-queries, or retrieval-scope adaptation, leads to substantial performance degradation across all three datasets. In particular, removing retrieval-scope adaptation causes the largest drop in both Recall@5 and F1, confirming that dynamically adapting the retrieval scope is the key factor for effective multi-hop evidence acquisition. Removing dependency-aware decomposition also consistently reduces performance, demonstrating the importance of explicitly modeling inter-query dependencies for progressively refining downstream retrieval targets.

Beyond the ablation results, we further compare ReScopeRAG with a global-scope variant, in which each Sub-Agent performs retrieval over the entire knowledge graph at every iteration while keeping the other components unchanged. Global Scope falls behind ReScopeRAG by 8.59 and 7.73 points in average Recall@5 and F1, respectively. The gap is particularly pronounced on MuSiQue, where ReScopeRAG improves Recall@5 and F1 by 17.14 and 12.08 percentage points, respectively. These results suggest that simply expanding the retrieval scope is insufficient. In contrast, ReScopeRAG dynamically adapts the retrieval scope according to the evolving evidence, achieving a better balance between evidence coverage and retrieval focus.

## 5.2 Analysis by Decomposition Complexity

We also analyze the decomposition structures generated by Master. As shown in Table 4, questions are generally decomposed into multiple sub-queries. Placeholder usage is also substantial across all datasets, accompanied by non-trivial dependency depths, indicating that retrieval targets often depend on evidence obtained from earlier sub-queries. These patterns are especially pronounced on MuSiQue, which exhibits the highest placeholder prevalence and dependency depth. We further examine the resolution of these placeholders during online retrieval. The generally high resolution rates indicate that intermediate evidence can successfully instantiate most underspecified retrieval targets, while the lower resolution on MuSiQue reflects the greater difficulty introduced by its denser placeholders and deeper dependencies. Overall, the result shows that dependency-aware information flow is not a marginal case. It provides the basis for retrieval-scope adaptation.

# 6 Conclusion

In this paper, we introduced ReScopeRAG, an iterative graph retrieval framework that dynamically adapts retrieval scopes based on evidence acquired during multi-step reasoning. By combining dependency-aware sub-query planning with Master–Sub-Agent retrieval, ReScopeRAG allows accumulated evidence to guide dynamic scope adjustment across graph-induced knowledge regions. This transforms graph-based RAG from retrieving within an initially fixed scope to adapting retrieval space as reasoning progresses. Experiments on general multi-hop question answering datasets show that ReScopeRAG achieves superior or competitive performance against multiple RAG baselines. The results suggest that explicitly modeling retrieval-scope evolution is a promising direction for improving evidence acquisition in multi-hop question answering.

### AI USE STATEMENT

In this work, we used generative AI tools for language polishing, including grammar correction and rephrasing of the manuscript. We have not used generative AI tools for experimental design, data collection, analysis, or result interpretation. We have reviewed all AI-assisted work. All AI-assisted text was carefully reviewed and revised by the authors to ensure it accurately reflects our intended meaning and original research contributions. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

This work does not involve human subjects, personally identifiable information, or sensitive data. We use publicly available datasets for evaluation.

### REPRODUCIBILITY STATEMENT

The details of our method are provided in Section 3, and the datasets, baselines, metrics, and experimental settings are described in Section 4.1. Additional experimental results are provided in Appendix A, and additional implementation details are provided in Appendix B. The code and data are available at `https://anonymous.4open.science/r/ReScopeRAG-code`.

## REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pp. 9112–9141, 2024.

Ines Besrour, Jingbo He, Tobias Schreieder, and Michael Färber. Ragenta: Multi-agent retrieval-augmented generation for attributed question answering. *arXiv preprint arXiv:2506.16988*, 2025.

Chia-Yuan Chang, Zhimeng Jiang, Vineeth Rakesh, Menghai Pan, Chin-Chia Michael Yeh, Guanchu Wang, Mingzhi Hu, Zhichao Xu, Yan Zheng, Mahashweta Das, et al. Main-rag: Multi-agent filtering retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2607–2622, 2025.

Boyu Chen, Zirui Guo, Zidan Yang, Yuluo Chen, Junze Chen, Zhenghao Liu, Chuan Shi, and Cheng Yang. Pathrag: Pruning graph-based retrieval augmented generation with relational paths. In *Proceedings of the AAAI conference on artificial intelligence*, volume 40, pp. 30183–30191, 2026.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tian Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. In *EMNLP (Findings)*, pp. 10746–10761, 2025.

Bernal J Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569, 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*, 2025.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907, 2024.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)*, pp. 7036–7050, 2024.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7969–7992, 2023.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 39–48, 2020.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*, 2022.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Shilong Li, Yancheng He, Hangyu Guo, Xingyuan Bu, Ge Bai, Jie Liu, Jiaheng Liu, Xingwei Qu, Yangguang Li, Wanli Ouyang, et al. Graphreader: Building graph-based agent to enhance long-context abilities of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 12758–12786, 2024.

Reza Yousefi Maragheh, Pratheek Vadla, Priyank Gupta, Kai Zhao, Aysenur Inan, Kehui Yao, Jianpeng Xu, Praveen Kanumala, Jason Cho, and Sushant Kumar. Arag: Agentic retrieval augmented generation for personalized recommendation. *arXiv preprint arXiv:2506.21931*, 2025.

Thang Nguyen, Peter Chin, and Yu-Wing Tai. Ma-rag: Multi-agent retrieval-augmented generation via collaborative chain-of-thought reasoning. *arXiv preprint arXiv:2505.20096*, 2025.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, et al. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9844–9855, 2022.

Stephen E Robertson and Steve Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*, pp. 232–241. Springer, 1994.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, volume 2024, pp. 32628–32649, 2024.

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

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019.

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 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)*, pp. 10014–10037, 2023.

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Nan Zhang, Prafulla Kumar Choubey, Alexander Fabbri, Gabriel Bernadett-Shapiro, Rui Zhang, Prasenjit Mitra, Caiming Xiong, and Chien-Sheng Wu. Sirerag: Indexing similar and related information for multihop reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 100413–100432, 2025.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. Knowledge graph-guided retrieval augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8912–8924, 2025.

# A Additional Experimental Results

## A.1 Case Study

Figure 3 presents a case from 2WikiMultiHopQA. It illustrates why retrieval-scope adaptation is necessary for multi-hop reasoning over graph-induced knowledge regions. In this case, the initial scopes assigned to the four Sub-Agents contain no supporting evidence, so a static-scope variant would have no direct access to the required gold passages. ReScopeRAG first move the retrieval scope from the initial region to clusters containing information of two films, where the directors of Dus and The Negotiator are identified as Anubhav Sinha and F. Gary Gray. This step shows that region transition is useful for acquiring intermediate evidence before the full reasoning target is known. More importantly, the retrieved film evidence is not only used as answer context; it changes the retrieval target of the dependent sub-queries by resolving their placeholders. The Master therefore stops the solved SA1 and SA3, and continues only SA2 and SA4, avoiding unnecessary exploration after the prerequisite evidence has served its role.

The later rounds clarify two roles of Sub-Agent reinitialization. For the SA2 branch, reinitialization resets the local retrieval state, allowing the Sub-Agent to re-search C109 after Anubhav Sinha is identified, avoiding being blocked by the previous visit record. For the SA4 branch, reinitialization relocates the Sub-Agent from the earlier film-centered scope to a region closer to F. Gary Gray's birth-year evidence, reducing unnecessary exploration before reaching the biography. Once the accumulated evidence is sufficient in Round 4, the Master terminates retrieval before reaching the maximum iteration budget of six rounds. The case provides an intuitive trace of how dependency-aware information flow, region transition, reinitialization, and Master scheduling interact during retrieval-scope adaptation.

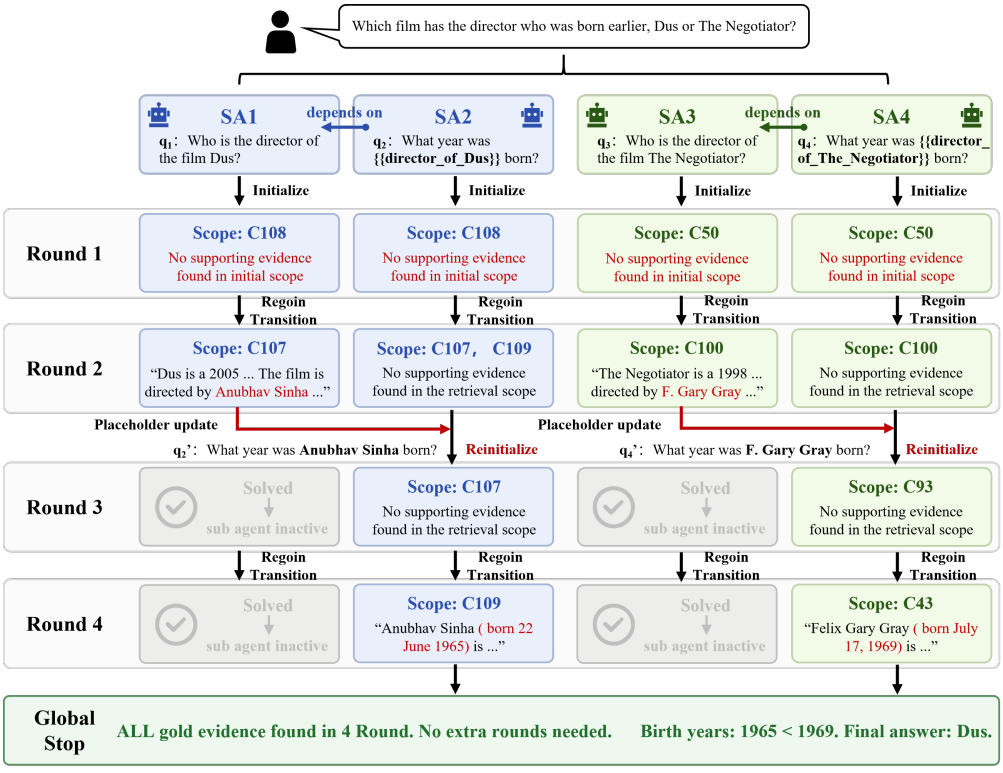


Figure 3: Case study of ReScopeRAG.

## A.2 Effect of Iterative Retrieval Budget

We investigate the effect of the maximum retrieval iterations by varying the iteration budget from 1 to 6, as shown in Figure 4.

Across all datasets, increasing the retrieval budget consistently improves both retrieval and answer quality, although the gains gradually diminish after several rounds. Additional iterations substantially improve Recall@n, indicating that later rounds successfully discover supporting evidence beyond the initial retrieval scope. These retrieval gains are consistently reflected in EM and F1, showing that better evidence acquisition leads to better answer generation.

The improvements become marginal after four iterations, suggesting that most useful evidence has already been collected. This trend aligns with ReScopeRAG's design, where early iterations resolve placeholders and relocate the retrieval scope, and later ones refine evidence within it. Moreover, ReScopeRAG terminates much earlier in practice, requiring only 3.09, 4.21, and 4.69 iterations on average for HotpotQA, 2WikiMultiHopQA, and MuSiQue, respectively. This indicates that the Master can effectively stop retrieval once sufficient evidence has been accumulated, avoiding unnecessary retrieval cost.

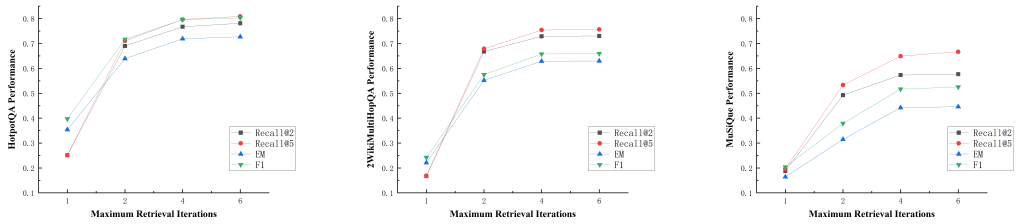


Figure 4: Effect of iterative retrieval budget on retrieval and question answering performance.

# B Implementation Details

## B.1 ReScopeRAG Configuration

We use the following configurations for ReScopeRAG. The maximum number of online retrieval iterations limits the number of evidence-driven retrieval rounds for each question, while the maximum number of decomposed sub-queries bounds the number of reasoning targets produced by the Master. For single-query questions that the Master identifies as unnecessary to decompose, we initialize the Sub-Agent with the 5 highest-ranked candidate clusters to provide multiple starting retrieval scopes. At each iteration, each active Sub-Agent can contribute at most 3 passages to the Evidence Pool, and only top-5 passages are retained as the final evidence for answer generation. During retrieval-scope navigation, the candidate region probing budget is set to 5, limiting the maximum number of candidate regions examined when determining the next retrieval scope. The PPR damping parameter is set to 0.5.

Table 5: Implementation configurations of ReScopeRAG.

| **Configuration** | **Value** |
|---|---|
| Maximum online retrieval iterations | 6 |
| Maximum decomposed sub-queries | 5 |
| Initial clusters for single-query questions | 5 |
| Maximum passages per Sub-Agent | 3 |
| Final evidence passages | 5 |
| Candidate region probing budget | 5 |
| PPR damping parameter | 0.5 |

## B.2 Baseline Configuration

For all baselines, we follow their official implementations and configurations with minimal modifications. To ensure a fair comparison, we use the same LLM and embedding model across methods

and modify prompts for methods that do not originally produce short answers. The following describes the additional method-specific settings applied for fair comparison.

For conventional retrieval baselines, we set the retrieval top-$k$ to 5 for answer generation.

For graph-based methods, HippoRAG2 and GraphRAG are run with their default settings. For LightRAG, we use the top-5 retrieved chunks for answer generation, as its chunk granularity is approximately aligned with that of the original passage across all three datasets. ArchRAG originally generates multiple short answers, so we compute EM and F1 for each answer and take the highest score for each question.

For agent-based baselines, we set the number of evidence passages used at each step to 5 for MARAG to match our experimental setting. RAGentA is run with its official configuration.

### B.3 Prompt Templates

We provide the main prompt templates used by the Master and Sub-Agents during the retrieval stage.

#### B.3.1 Dependency-aware Query Decomposition

The Master first decomposes the original question into sub-queries while explicitly modeling their dependency relations and unresolved placeholders.

```
System:

You decompose a user question into atomic sub-questions for knowledge-graph retrieval.

HARD RULES:
1. Each sub-question asks for ONE concrete fact (one entity attribute or one relation fact
     ).
2. At most ONE {{placeholder}} per sub-question. Direct hops: no {{}} and depends_on=[].
3. Chain-only depends_on: at most ONE predecessor id per sub-question. Trust this graph
     exactly.
4. Independent parallel hops (e.g. compare A vs B): both direct, depends_on=[].
5. Do NOT keep comparative questions; split into separate factual lookups with entity
     names.
6. Order sub-questions logically; do not duplicate.

Output STRICT JSON only:
{"sub_queries":[{"id":int,"text":string,"depends_on":[int,...],
"placeholder":null|{"slot":string,"hint":string}}, ...]}

List at most {cap} items.

Examples:

Q: "What year was A's father born?"
{"sub_queries":[
  {"id":0,"text":"Information about A","depends_on":[],"placeholder":null},
  {"id":1,"text":"What year was {{father_of_A}} born?","depends_on":[0],
   "placeholder":{"slot":"father_of_A","hint":"A's father's full name"}}
]}

Q: "Who was born earlier, A or B?"
{"sub_queries":[
  {"id":0,"text":"Birth year of A","depends_on":[],"placeholder":null},
  {"id":1,"text":"Birth year of B","depends_on":[],"placeholder":null}
]}

Q: "Who was born earlier, A's father or B's director?"
{"sub_queries":[
  {"id":0,"text":"Information about A","depends_on":[],"placeholder":null},
  {"id":1,"text":"What year was {{father_of_A}} born?","depends_on":[0],
   "placeholder":{"slot":"father_of_A","hint":"A's father's full name"}},
  {"id":2,"text":"Information about B","depends_on":[],"placeholder":null},
  {"id":3,"text":"What year was {{director_of_B}} born?","depends_on":[2],
   "placeholder":{"slot":"director_of_B","hint":"B's director's full name"}}
]}

User:

Question: {question}
```

### B.3.2 QUERY-GAP-GUIDED CLUSTER NAVIGATION

At each retrieval round, the Sub-Agent ranks candidate clusters according to the current query gap and query-aligned evidence.

```
System:

You are an inter-cluster navigation decision-maker.
Given query_gap and query-aligned snippets (top_entities / top_passages),
rank clusters by likelihood of resolving the gap.

Output strict JSON only:
{"ranked_cluster_ids": [<int>, ...],
 "agent_confidence": <float 0..1>}.

Every id must be from the candidate list; most likely first; no duplicates.

User:

Query gap:
{query_gap}

Candidate clusters:
{candidate_clusters}
```

### B.3.3 EVIDENCE ANALYSIS AND QUERY-GAP UPDATING

After each retrieval round, the Master analyzes the accumulated evidence, determines whether the original question can be answered, updates the query gap of each Sub-Agent, and resolves placeholders when sufficient evidence becomes available.

```
System:

You are a master coordinator for multi-agent, multi-hop retrieval.

1. DECISION (is_sufficient):
Using ONLY the evidence pool, decide if the ORIGINAL question can be fully answered.
Set is_sufficient=true ONLY when EVERY part of the original question is explicitly
     supported by evidence  not inferred, not partial, not one-sided.

- Comparison / multi-entity questions (e.g. "same country?", "who was born later?",
  "which film has the director born first?"): you MUST have the required fact for EACH
  entity or EACH side before is_sufficient=true.
- If ANY agent's sub_query is still unanswered from the pool, is_sufficient=false.
- Mentioning an entity name or related topic is NOT enough; the specific asked attribute
  (country, birth year, director name, lifespan, etc.) must appear in evidence.
- When in doubt, is_sufficient=false.

2. PER-AGENT BRIDGE:
For EACH agent, output query_gap for the NEXT retrieval round  tailored to THAT
agent's sub_query only. Use bridge entities from the evidence pool.

3. PLACEHOLDER SUBSTITUTION (agents with non-null placeholder):
- Resolve the slot ONLY using evidence found by the predecessor agent in depends_on
  (chain rule: sq1 uses sq0's findings; sq3 uses sq2's findings).
- Set placeholder_substituted=true ONLY on the round when {{slot}} is FIRST replaced
  with a UNIQUELY IDENTIFIABLE entity (full person name, full work title, etc.).
- If the slot was already replaced in a prior round, set placeholder_substituted=false
  even when you rephrase query_gap (e.g. "Birth year of X" -> "When was X born?").
- Do NOT set true for vague relations only ("father", "the company", "director").
- When true, query_gap MUST be the FULL searchable question AFTER substitution.
- When not resolved, placeholder_substituted=false; query_gap may keep {{slot}}.
- Direct agents (placeholder=null): placeholder_substituted=false,
  substituted_entity="".

Return STRICT JSON only (no markdown, no thinking, no nested JSON inside strings):
{"is_sufficient": bool,
 "confidence": number,
 "reason": string,
 "agent_gaps": [
   {"agent_id": string,
    "query_gap": string,
```

```
    "reason": string,
    "placeholder_substituted": bool,
    "substituted_entity": string}, ...
 ]}.

agent_gaps must include every agent_id from the input list exactly once.

Keep EVERY reason field to ONE short sentence.
Do NOT put chain-of-thought, self-correction, or draft JSON inside any reason.

Examples (agent_gaps entries only):

Resolved (placeholder_substituted=true):
sub_query: "What year was {{father_of_A}} born?",
depends_on:[0],
evidence: "A's father is John Smith"
->
query_gap:"What year was John Smith born?",
placeholder_substituted:true,
substituted_entity:"John Smith"

Partial (placeholder_substituted=false):
evidence: "A has a businessman father"
->
query_gap:"What year was {{father_of_A}} born?",
placeholder_substituted:false,
substituted_entity:""

Direct agent:
sub_query:"Information about A"
->
query_gap:"Information about A",
placeholder_substituted:false,
substituted_entity:""

User:

Original question:
{original_question}

Evidence pool:
{evidence_pool}

Active sub-agents:
{sub_agents}
```

### B.3.4 SUB-AGENT SCHEDULING

The Master determines whether each Sub-Agent should continue retrieval based on the current evidence and retrieval state.

```
System:

You are the master orchestrator for multi-agent retrieval.

Output strict JSON only with shape:
{"agents":[{"agent_id":string,"active":bool,"reason":string},...]}.

Decide for EACH sub-agent whether they should run another retrieval round
(active=true) or stop (active=false).

CRITICAL  do NOT stop an agent while its sub_query is still unanswered:

- Set active=false ONLY when ALL of the following hold for that agent:
  (a) its sub_query is already fully answered by the current evidence pool, OR
  (b) this round's report shows an unrecoverable error, OR
  (c) another agent has strictly superseded the same need and the pool already contains
      everything this agent's sub_query requires.

- Set active=true when assessment.is_sufficient=false AND this agent's sub_query or
  current_query_gap still seeks missing facts needed for the original question.

- Set active=true when this agent's query_gap in assessment.agent_gaps still targets
  unresolved information (including {{slot}} not yet filled).
```

```
- Do NOT set active=false just because this round added few/no new evidence, or because
  the agent visited many clusters  more navigation rounds may still be needed.

- Do NOT set active=false for one side of a comparison/multi-entity question while the
  other side is still being retrieved; each agent owns its own sub_query until satisfied.

- If assessment.is_sufficient=true, all agents should be stopped (handled upstream);
  otherwise default to keeping agents active unless you are certain that agent's
  sub_query is complete in the evidence pool.

- When uncertain whether an agent's sub_query is satisfied, prefer active=true.

User:

Current retrieval state:
{retrieval_state}
```