Connection-Aware Graph Pruning in RAG for Multi-Hop Question Answering

Jiaxin Wu[1], Miao Fan[1,*], Wenjie Zhou[2], Chen Ma[3], Xiangzeng Liu[4], Haoyi Xiong[5]

[1]Beijing Institute of Graphic Communication, [2]Renmin University of China,
[3]City University of Hong Kong, [4]Xidian University, [5]Independent Scholar

***Abstract*—Retrieval-augmented generation (RAG) for multi-hop question answering is often compromised by the low signal-to-noise ratio in retrieved contexts, necessitating effective pruning. However, standard relevance-based pruning strategies frequently discard intermediate evidence that lacks high query similarity, inadvertently severing the reasoning chain and leading to incorrect answers. To address this limitation, we introduce a connection-aware graph-pruning framework for post-retrieval processing in RAG that explicitly preserves the evidence-chain structure under a fixed token budget. We model the retrieved context by constructing a sentence-level evidence graph over the top-N retrieved documents, in which nodes represent candidate sentences and edges encode plausible multi-hop transitions. We then formulate pruning as budgeted subgraph selection, preserving coherent evidence pathways that link question-anchored nodes to potential answer candidates. Specifically, our method uses shortest-path augmentation to recover non-obvious intermediate evidence required to repair fragmented reasoning routes, followed by connectivity-preserving pruning that eliminates redundancy to maximize information density. Experiments on HotpotQA and FanOutQA demonstrate that, under identical token budgets, our framework consistently surpasses relevance-based baselines in answer accuracy and supporting-fact retention, while exhibiting superior robustness against retrieval noise.**

***Index Terms*—multi-hop question answering, retrieval-augmented generation, graph pruning, subgraph selection**

I. Introduction

Retrieval-augmented generation (RAG) has emerged as a standard paradigm for grounding large language models (LLMs) in knowledge-intensive tasks by conditioning generation on external evidence [9]–[11]. While RAG pipelines often succeed on single-hop questions that can be answered with highly relevant passages, multi-hop question answering (QA) requires synthesizing evidence distributed across multiple documents. A fundamental challenge here is identifying intermediate evidence, which consists of facts that connect entities or relations across hops but often lack significant lexical overlap with the original query.

Preserving the integrity of these evidence chains is a prerequisite for reliable complex reasoning. However, the volume of retrieved text often exceeds the LLM's input token budget. As a result, context pruning becomes unavoidable. Long-context windows alone are insufficient, as LLMs often underuse buried evidence and may degrade with longer inputs [12], [13]. Therefore, RAG systems routinely prune retrieved candidates before prompting. Yet, standard relevance-based pruning, which selects sentences based on local similarity or ranking scores, is particularly brittle for multi-hop QA. Intermediate evidence often appears irrelevant until earlier hops are resolved, so removing it yields a context that is superficially on-topic but lacks a connected reasoning path from the question to the answer. Prior work mitigates context limitations through improved retrieval, ranking, or compression [3]–[6], [14]. While graph-based retrieval has been explored to capture relational dependencies [7], [15], most existing pruning strategies still optimize for individual relevance rather than structural connectivity. Consequently, they risk discarding low-scoring but topologically critical bridge nodes, severing the reasoning chains essential for deduction.

To address this problem, we propose connection-aware graph pruning, which introduces an explicit structural signal into post-retrieval processing. We organize retrieved evidence at sentence granularity into a graph whose edges approximate plausible multi-hop transitions. We then select a budgeted

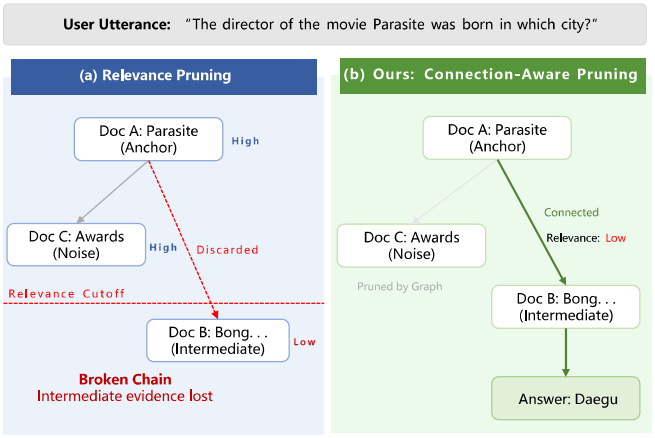

Fig. 1. Regarding the user's utterance, the reasoning chain connects the film Parasite to its director, Bong Joon-ho, and then to Daegu, his birthplace. While relevance-based methods (a) often fail to retrieve intermediate evidence (Doc B) that acts as a low-scoring though essential bridge. In contrast, our approach (b) models the context as a graph. It selects a connected subgraph, preserving the structural integrity of the reasoning path.

*Correspondence: fanmiao@bigc.edu.cn

subset that prioritizes maintaining reachable pathways from question-anchored nodes to answer-oriented candidates. As illustrated in Figure 1, our method restores missing bridge sentences and prunes redundant content while preserving reachability under the token budget. As the procedure only requires standard retrieval scores and token costs, it can be applied as a plug-and-play module without retraining retrievers or readers.

Our contributions are:

- We study post-retrieval pruning for multi-hop RAG from a connectivity perspective, and show that relevance-only pruning misaligns with multi-step reasoning.
- We implement this idea using a practical two-stage procedure: bridge recovery via shortest-path augmentation , and greedy pruning that preserves connectivity under a fixed token budget.
- We empirically validate the proposed pruning strategy on HotpotQA and FanOutQA, showing improved answer accuracy and evidence retention under fixed token budgets.

II. Related Work

Multi-hop question answering (QA) benchmarks assess whether models can reconstruct full reasoning chains spanning multiple documents, where multi-hop refers to the need for multiple inferential steps. HotpotQA provides sentence-level supporting facts. Subsequent datasets, such as 2WikiMultiHopQA and MuSiQue, enforce multi-step reasoning [8], [16]. FanOutQA further stresses long-context, multi-branch dependencies. These settings highlight a key practical issue in RAG. Errors often arise not only from document retrieval but also from discarding intermediate evidence while selecting relevant context for models. For retrieval, BM25 and DPR are strong baselines [17], [18], while multi-step retrieval and graph traversal methods aim to uncover weakly matched intermediate evidence [3]. Recent graph-enhanced agentic retrieval further strengthens multi-hop discovery under weak lexical overlap [23]. As retrieved text frequently exceeds input budgets, RAG systems rely on reranking, pruning, and compression [9], [11]. Approaches include rank-generation coupling [4], [19], learned compression [5], [6], and explicit context refinement via compression or sentence reconstruction [21], [22]. Graph-based organization has also been explored for multi-hop RAG. This includes graph construction [15], path-oriented prompting [7], and textual-subgraph retrieval or reorganization methods [24], [25]. Despite these advances, most refinements remain relevance-based. This is brittle in multi-hop QA because intermediate evidence can be low-scoring until earlier hops are resolved. Taken together, this motivates our connection-aware pruning: we explicitly preserve evidence-chain connectivity under a fixed token budget.

III. Problem Formulation

Local pruning based solely on relevance often fails in multi-hop settings by discarding intermediate evidence with weak query similarity but vital structural roles. This breaks reasoning chains. Although recent work improves retrieval or compression, they generally lack the means to enforce evidence connectivity explicitly.

A. Task Definition

We focus on the context pruning stage in a Retrieval-Augmented Generation (RAG) pipeline. Given a multi-hop question q and a set of retrieved candidate units $C = \{v_1, \ldots, v_M\}$, each unit v_i has an associated token cost $c(v_i)$ and a local relevance score $s(v_i)$. In this work, candidate units are instantiated at the sentence level by segmenting the top-N retrieved documents into sentences. In practical settings, the total length $\sum_i c(v_i)$ typically exceeds the LLM's input budget B. Our goal is to select a subset $S \subseteq C$ such that $\sum_{v \in S} c(v) \leq B$, while maximizing the likelihood that S preserves the complete reasoning chain required to answer q.

B. Evidence Graph Construction

To capture the inter-dependencies between candidates, we model the retrieved context $\mathcal{C}$ as a directed Evidence Graph $G = (V, E)$. The node set V corresponds to the candidate units in $\mathcal{C}$. The edge set E represents potential reasoning steps, constructed based on semantic or lexical relations (e.g., entity co-reference, hyperlinks, or shared rare tokens). Unlike standard selection methods that treat V as independent items, the graph structure implies that the utility of a node v depends not only on its own score $s(v)$ but also on its function in linking other nodes.

C. Node Roles in the Evidence Graph

We define specific semantic roles for nodes within the graph:

- **Source Terminals (T_q):** Nodes that contain entities or key phrases explicitly mentioned in the question q.
- **Target Terminals (T_a):** Nodes with high local relevance $s(v)$ that act as potential answer candidates.
- **intermediate evidence:** Intermediate nodes $v \notin T_q \cup T_a$ that act as connectors. While they may have low $s(v)$, they are necessary to form a path $u \rightsquigarrow v \rightsquigarrow t$ from a source $u \in T_q$ to a target $t \in T_a$

D. Connectivity-Aware Objective

We define the Chain Coverage $\Phi(S)$ of a selected subset S. $\Phi(S)$ is the fraction of target terminals that remain accessible from source terminals in the induced subgraph $G[S]$:

$$\Phi(S) = \frac{1}{|T_a|} \sum_{t \in T_a} \mathbb{I}\left(\exists u \in T_q \text{ s.t. } u \rightsquigarrow_{G[S]} t\right). \tag{1}$$

The pruning problem is formally cast as a *budgeted connectivity-preserving maximization*:

$$\max_{S \subseteq V} \quad \mathcal{J}(S) = (1-\lambda) \sum_{v \in S} s(v) + \lambda \cdot \Phi(S), \tag{2}$$

$$\text{s.t.} \quad \sum_{v \in S} c(v) \leq B. \tag{3}$$

Here, λ balances local relevance and structural integrity. This formulation explicitly penalizes removing bridge nodes: even

if removing a bridge saves budget, doing so decreases $\Phi(S)$ by breaking the path to a target, thereby lowering the total objective $\mathcal{J}(S)$. We use Equation 2 as a motivating objective rather than claiming an exact solution to the underlying combinatorial problem.

IV. Proposed Solution

To approximately solve the budgeted connectivity-preserving maximization in Equation 2 and 3. We propose connection-aware graph pruning, a post-retrieval refinement module that reduces prompt length while explicitly preserving multi-step evidence connectivity. The module is model-agnostic: it only requires retrieved candidates with local relevance scores and their token costs. It can be composed with iterative retrieval [3] and rank-aware RAG pipelines [4]. It's an explicit objective to preserve the connected reasoning routes needed for multi-hop deduction.

A. Evidence Graph Construction

Sentence units and node attributes. In Step (a) of Figure 2, given the top-N retrieved documents, we split them into sentence-level units $\mathcal{C} = \{v_1, \ldots, v_M\}$ and treat each unit as a node $v \in V$. Each node is associated with a token cost $c(v)$ and a local relevance score $s(v) \in \mathbb{R}_{\geq 0}$ (e.g., from a dense retriever or reranker). Optionally, we discard evident noise using a lightweight pre-filter (e.g., when $s(v)$ is very low) [5]. This yields a directed evidence graph $G = (V, E)$.

Edge construction and edge weights. Edges represent plausible multi-hop transitions between evidence units. For an ordered pair (u, v), we add an edge $u \to v$ if at least one of the following holds: (i) u and v are close within the same document (local discourse continuity), (ii) they share salient entities (entity-based transitions), or (iii) they are linked by explicit hyperlinks when available [20]. We assign each edge a transition strength

$$w(u, v) = \alpha\, w_{\text{doc}}(u, v) + \beta\, w_{\text{ent}}(u, v) + \eta\, w_{\text{link}}(u, v), \quad (4)$$

where w_{doc} can be a decaying function of sentence distance, w_{ent} measures entity overlap (e.g., TF-IDF-weighted Jaccard over NER spans), and $w_{\text{link}} \in \{0, 1\}$ indicates hyperlink relations. For shortest-path augmentation, we convert transition strength to a nonnegative edge cost:

$$\ell(u, v) = \frac{1}{w(u, v) + \epsilon}, \quad (5)$$

with $\epsilon > 0$.

Terminals. We define two sets of terminals to instantiate chain connectivity. The source terminals T_q are nodes that contain the entities or key phrases mentioned in the question q. In practice, target terminals T_a are instantiated as the top-K nodes by local relevance $s(v)$, or by response likelihood when such scores are available. Thus, the multi-hop pruning problem is operationalized as preserving coherent routes from T_q to this top-K target set in G.

B. Connection-Aware Graph Pruning

We employ a two-stage heuristic aligned with Equation 2. First, we repair fragmented evidence routes by explicitly recovering missing intermediate evidence. Second, we prune the repaired subgraph to satisfy the token budget while preserving connectivity. The key distinction is that pruning is a selective removal process constrained by connectivity, rather than generic content compression.

Stage 1: Intermediate evidence recovery by multi-source shortest-path augmentation. We initialize the selected set

$$S \leftarrow T_q \cup T_a. \quad (6)$$

For each target $t \in T_a$, we compute the minimum-cost distance from any source using multi-source Dijkstra:

$$\text{dist}(t) = \min_{u \in T_q} \text{dist}(u \rightsquigarrow t; G), \quad (7)$$

where path cost is the sum of edge costs ℓ. If t is reachable, we recover its shortest path P_t and augment:

$$S \leftarrow S \cup \bigcup_{t \in T_a} P_t. \quad (8)$$

The newly added nodes on P_t constitute intermediate evidence: they may have modest $s(v)$ but are structurally necessary to connect question-anchored sources to answer-oriented targets. To control expansion, we optionally impose a hop limit $|P_t| \leq L$ and restrict augmentation to the top-K' targets by $s(t)$.

Stage 2: Budgeted connectivity-preserving pruning If the augmented selection violates the token budget, that is $\sum_{v \in S} c(v) > B$, we iteratively prune nodes until feasibility. We reuse the objective from Equation 2:

$$\mathcal{J}(S) = (1 - \lambda) \sum_{v \in S} s(v) + \lambda \Phi(S), \quad (9)$$

and define the marginal loss incurred by removing node v:

$$\Delta\mathcal{J}(v; S) = \mathcal{J}(S) - \mathcal{J}(S \setminus \{v\}). \quad (10)$$

We prune the node with the smallest loss per token cost:

$$v^* = \arg\min_{v \in \mathcal{R}(S)} \frac{\Delta\mathcal{J}(v; S)}{c(v)}, \qquad S \leftarrow S \setminus \{v^*\}, \quad (11)$$

where $\mathcal{R}(S)$ denotes the set of removable nodes. To explicitly protect evidence connectivity, we constrain removability by preventing large drops in chain coverage:

$$\mathcal{R}(S) = \{v \in S \setminus T_q : \Phi(S \setminus \{v\}) \geq \Phi(S) - \tau\}, \quad (12)$$

with tolerance τ (default 0). Operationally, $\Phi(\cdot)$ is recomputed by reachability tests from T_q to T_a in the induced subgraph $G[S]$. This pruning criterion removes redundant or peripheral nodes while preserving intermediate evidence that is essential for keeping targets reachable. We note that connectivity is only a structural proxy for reasoning quality and does not guarantee the semantic correctness of a retained path. If the evidence graph contains spurious edges, the method may preserve a noisy bridge simply because it is necessary for reachability.

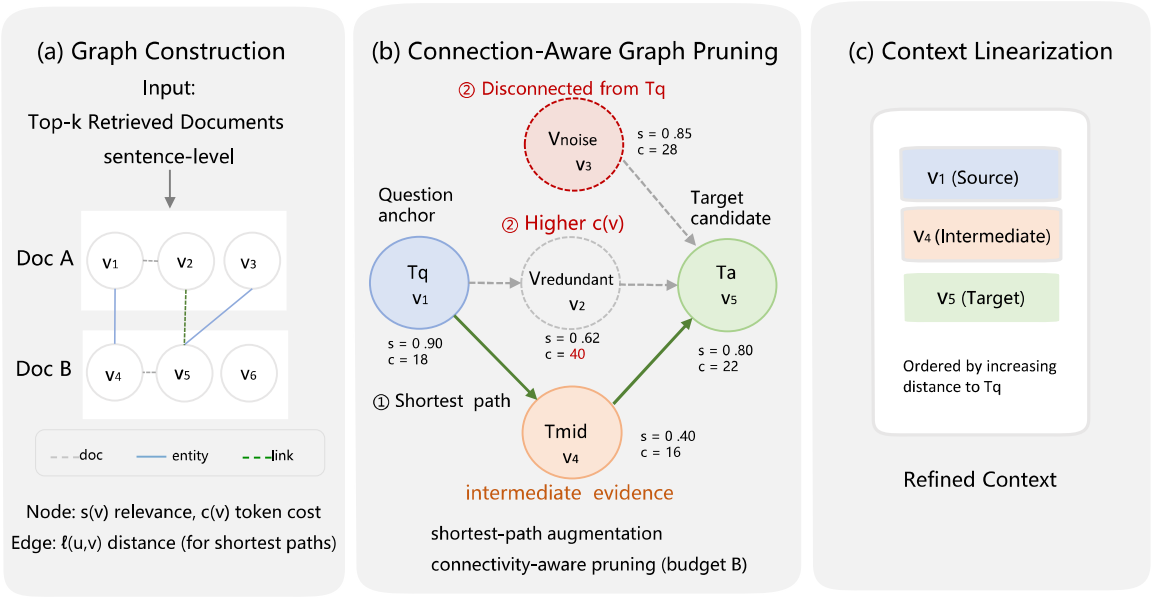

Fig. 2. Overview of connection-aware graph pruning: (a) Retrieved passages are split into sentence units and organized as an evidence graph with intra-document, entity, and hyperlink edges. (b) We recover necessary intermediate evidence via shortest-path augmentation from question anchors T_q to target candidates T_a, then prune redundant or disconnected nodes under a token budget $\sum c(v) \leq B$ while preserving reachability. (c) The retained subgraph is linearized into an ordered prompt to form the refined context.

C. Context Linearization for Prompting

The final selected nodes S induce a connected or near-connected evidence subgraph. To generate a coherent prompt, we linearize the selected evidence by prioritizing recovered reasoning routes. We rank nodes by their shortest-path distance from the sources in $G[S]$:

$$\pi(v) = \min_{u \in T_q} \mathrm{dist}(u \rightsquigarrow v; G[S]), \tag{13}$$

and output nodes in nondecreasing $\pi(v)$ order, breaking ties by document order to preserve local discourse coherence. This yields a structured input that reflects multi-step dependencies, aligning with supporting-fact evaluations in multi-hop benchmarks such as HotpotQA.

V. Experiments

We assess post-retrieval pruning for multi-hop QA along three complementary dimensions: (i) Answer correctness, measuring whether the reader produces the correct final answer; (ii) Evidence fidelity, measuring whether gold supporting facts are retained after pruning; and (iii) Chain integrity, measuring whether the retained evidence preserves multi-hop reachability between question-anchored cues and answer-oriented candidates.

A. Evaluation Setup

a) Benchmarks: We evaluate on HotpotQA (development set), which provides sentence-level supporting-fact annotations for evidence-level assessment [1]. We also use FanOutQA (development set), which targets fan-out style multi-document reasoning with long-context dependencies [2]. We follow the official evaluation protocols and report results on the development splits.

b) RAG pipeline and candidate pool: All methods share the same RAG pipeline to isolate the effect of pruning. We use a DPR-style dense retriever to retrieve the top-N documents per question and a FiD-style multi-document reader for answer generation. Each retrieved document is segmented into sentence units. All pruning methods operate on the same sentence-level candidate set V. Unless noted otherwise, N=30, yielding approximately $|V| \approx$ 300–800 sentences per query.

c) Token budget and packing: We enforce an input budget B measured by the reader tokenizer, including the question and all retained evidence. We set B=2048 for HotpotQA and B=3072 for FanOutQA to reflect the longer average evidence requirements in FanOutQA. For ranking-based baselines, candidates are appended in descending priority until the next unit would exceed B.

d) Baselines: We compare against representative relevance-based and learned refinement methods. Top-K (Sentence) selects sentence units by relevance $s(v)$ under budget. Top-K (Document) ranks documents by retrieval score and appends their sentences in original order until the budget is exhausted. RankRAG [4] couples ranking with generation-oriented training; PROVENCE [5] performs learned sentence pruning via sequence labeling; and CompAct [6] refines retrieved context under a fixed budget. For methods that are not directly compatible with our sentence-level post-retrieval pruning setting, we adapt their core scoring or pruning strategy within our unified pipeline. Differences mainly reflect pruning quality rather than retrieval quality. All

TABLE I

MAIN RESULTS UNDER IDENTICAL TOKEN BUDGETS. HIGHER IS BETTER FOR ALL METRICS. CONNECTION-AWARE PRUNING IMPROVES EVIDENCE RETENTION (SUPPF1) AND CONNECTIVITY (EPC), YIELDING HIGHER ANSWER ACCURACY.

	HotpotQA, B=2048				FanOutQA, B=3072		
Method	**EM**	**F1**	**SuppF1**	**EPC**	**EM**	**F1**	**EPC**
Top-K (Sentence)	60.8	72.4	60.9	0.52	24.6	44.8	0.41
Top-K (Document)	59.7	71.6	58.3	0.49	23.9	44.1	0.39
RankRAG [4]	61.5	73.0	62.0	0.55	25.4	45.6	0.44
PROVENCE [5]	61.2	73.3	64.1	0.58	25.1	46.2	0.47
CompAct [6]	61.0	72.9	63.4	0.57	25.0	45.9	0.46
Ours	**63.1**	**75.0**	**68.2**	**0.68**	**27.3**	**48.4**	**0.60**

baselines use the same retriever, reader, candidate pool, and budget definition.

e) Metrics: We report EM and F1 for answer correctness. For HotpotQA, we additionally report Supporting Fact F1 (SuppF1), which measures whether gold supporting sentences remain in the retained context after pruning, rather than whether the system predicts supporting facts. For both datasets, we report Evidence Path Coverage (EPC), the fraction of target terminals $t \in T_a$ reachable from any source terminal $u \in T_q$ in the induced subgraph $G[S]$.

f) Method configuration: Evidence graphs include intra-document proximity edges, entity-overlap edges, and hyperlink edges when available. We set $|T_a|{=}K{=}10$ and derive T_q by matching question entities/keywords. Unless otherwise specified, we set $\lambda{=}0.5$ in Equation 2 and use a hard connectivity constraint $\tau{=}0$ in Equation 12. These settings are fixed on the development set and used consistently across all reported runs. We report the average over three random seeds.

B. Main Results

Table I summarizes the main comparison under identical budgets. The key observation is that relevance-centric selection improves local informativeness but frequently breaks multi-hop chains, leading to lower EPC and reduced supporting-fact retention. In contrast, our method explicitly recovers and preserves intermediate evidence required for connectivity, resulting in consistent gains in EPC and SuppF1. Importantly, these structural improvements translate into higher downstream answer accuracy (F1/EM), demonstrating that preserving evidence connectivity is not merely diagnostic and is empirically associated with better multi-hop QA performance in our evaluated settings.

C. Evidence Composition Analysis

We analyze how different methods allocate a fixed token budget on HotpotQA. Each retained sentence is classified as either on-path evidence, lying on a source-target reasoning path in $G[S]$, or disconnected evidence, unreachable from any question anchor. Table II shows that other methods retain a substantial amount of disconnected evidence, whereas our method allocates a significantly larger fraction of the budget to on-path evidence, indicating more effective preservation of multi-hop reasoning chains.

TABLE II

EVIDENCE COMPOSITION UNDER A FIXED BUDGET ON HOTPOTQA, SHOWING THE PROPORTION OF ON-PATH VERSUS DISCONNECTED EVIDENCE.

Method	**On-path (%)**	**Disconnected (%)**
Top-K (Sentence)	42.1	57.9
CompAct [6]	48.6	51.4
Ours	**68.3**	**31.7**

D. Robustness to Retrieval Noise

To evaluate robustness under imperfect retrieval, we inject irrelevant documents into the candidate pool. Specifically, for each query, we add a noise set of size $r \cdot N$ drawn uniformly from the corpus, where $r \in \{0, 0.25, 0.5, 0.75, 1.0\}$. Figure 3 shows that relevance-based strategies degrade rapidly as noise increases, because distractors can receive high local relevance and crowd out necessary intermediate evidence. By contrast, disconnected noise nodes contribute little to EPC and are preferentially removed by connectivity-preserving pruning.

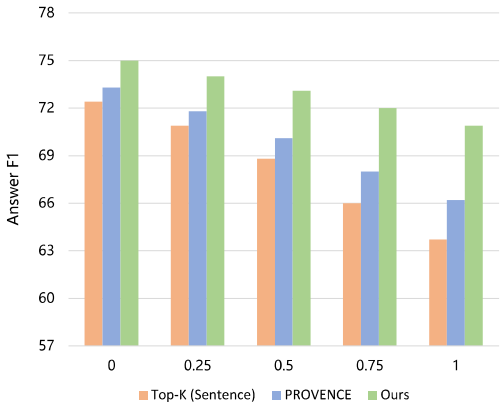

Fig. 3. Robustness to retrieval noise on HotpotQA. Connection-aware pruning degrades more slowly as distractors increase, consistent with EPC-based selection.

E. Ablation Study

We conduct a focused ablation study to isolate the contributions of the three core design choices: (i) shortest-path

augmentation for recovering missing intermediate evidence, (ii) connectivity-aware objective that explicitly optimizes chain integrity, and (iii) graph connectivity signals that instantiate multi-hop transitions. All ablations are evaluated on HotpotQA under the same budget B=2048. To elaborate on the impact of each ablation, Table III shows that removing shortest-path augmentation produces the most significant drop in EPC, indicating that explicit route repair is necessary to reconnect sources and targets. Furthermore, setting λ=0 collapses the method into cost-aware relevance pruning, substantially reducing SuppF1 and EPC and, correspondingly, degrading answer F1.

TABLE III
ABLATION STUDY ON HOTPOTQA, B=2048, ANALYZING THE CONTRIBUTION OF EACH COMPONENT IN THE PROPOSED METHOD.

Configuration	F1	SuppF1	EPC
Full method	**75.0**	**68.2**	**0.68**
w/o shortest-path augmentation	73.2	63.1	0.56
w/o connectivity term (λ=0)	72.8	61.7	0.52
w/o entity-overlap edges	73.5	64.0	0.59

VI. CONCLUSION

In this paper, we investigate context pruning for multi-hop question answering under a fixed token budget. We propose connection-aware graph pruning, which builds an evidence graph over retrieved candidates. Our experiments on multi-hop benchmarks show that it preserves intermediate evidence better than relevance-based baselines, improving answer accuracy and robustness to retrieval noise. These results show the value of modeling evidence connectivity for context pruning in complex reasoning tasks. However, our method has several limitations. First, its performance depends on the quality of the target set T_α. If answer-supporting evidence is not covered by the top-K targets, subsequent bridge recovery and pruning may be less effective. Second, graph construction and pruning hyperparameters, such as edge-weight design, λ, and the pre-filter rule. These can affect both connectivity estimation and robustness. We leave a systematic sensitivity analysis of these factors for future work.

ACKNOWLEDGMENTS

This work was supported by the Fundamental Research Funds for the Municipal Universities of Beijing (No. 04190125001/015).

REFERENCES

[1] Z. Yang *et al.*, "HotpotQA: A dataset for diverse, explainable multi-hop question answering," in *Proc. 2018 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 2369–2380.

[2] A. Zhu, A. Hwang, L. Dugan, and C. Callison-Burch, "FanOutQA: A multi-hop, multi-document question answering benchmark for large language models,"in *Proc. 62nd Annu. Meeting Assoc. Comput. Linguistics (ACL) (Vol. 2: Short Papers)*, Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 18–37.

[3] H. Trivedi *et al.*, "Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics (ACL) (Vol. 1: Long Papers)*, Toronto, Canada: Association for Computational Linguistics, 2023, pp. 10014–10037.

[4] Y. Yu *et al.*, "RankRAG: Unifying context ranking with retrieval-augmented generation in LLMs," in *Advances in Neural Information Processing Systems*, vol. 37, 2024.

[5] N. Chirkova, T. Formal, V. Nikoulina, and S. Clinchant, "Provence: Efficient and robust context pruning for retrieval-augmented generation," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2025.

[6] C. Yoon, T. Lee, H. Hwang, M. Jeong, and J. Kang, "CompAct: Compressing retrieved documents actively for question answering," in *Proc. 2024 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Miami, FL, USA: Association for Computational Linguistics, 2024, pp. 21424–21439, doi: 10.18653/v1/2024.emnlp-main.1194.

[7] B. Chen *et al.*, "PathRAG: Pruning graph-based retrieval augmented generation with relational paths," *arXiv*:2502.14902, 2025.

[8] X. Ho, A.-K. D. Nguyen, S. Sugawara, and A. Aizawa, "Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps," in *Proc. 28th Int. Conf. Comput. Linguistics (COLING)*, Barcelona, Spain (Online): International Committee on Computational Linguistics, 2020, pp. 6609–6625.

[9] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[10] P. Jiang, S. Ouyang, Y. Jiao, M. Zhong, R. Tian, and J. Han, "Retrieval and structuring augmented generation with large language models," in *Proc. 31st ACM SIGKDD Conf. Knowl. Discovery Data Mining (KDD)*, 2025, pp. 6032–6042.

[11] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in *Proc. Conf. Eur. Chapter Assoc. Comput. Linguistics (EACL)*, 2021.

[12] N. F. Liu *et al.*, "Lost in the middle: How language models use long contexts," *Trans. Assoc. Comput. Linguistics (TACL)*, 2024.

[13] Y. Bai *et al.*, "LongBench: A bilingual, multitask benchmark for long context understanding," in *Proc. Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2024.

[14] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to retrieve, generate, and critique through self-reflection," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2024.

[15] J. Larson and S. Truitt, "GraphRAG: Unlocking LLM discovery on narrative private data," *Microsoft Research Blog*, 2024.

[16] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "MuSiQue: Multihop questions via single-hop question composition," *Trans. Assoc. Comput. Linguistics*, vol. 10, pp. 539–554, 2022.

[17] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[18] V. Karpukhin *et al.*, "Dense passage retrieval for open-domain question answering," in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, 2020, pp. 6769–6781.

[19] W. Yang *et al.*, "End-to-end open-domain question answering with BERTserini," in *Proc. 2019 Conf. North American Chapter Assoc. Comput. Linguistics: Human Language Technologies (NAACL-HLT) (Demonstrations)*, Minneapolis, MN, USA: Association for Computational Linguistics, 2019, pp. 72–77.

[20] A. Asai, K. Hashimoto, H. Hajishirzi, R. Socher, and C. Xiong, "Learning to retrieve reasoning paths over Wikipedia graph for question answering," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2020.

[21] F. Xu *et al.*, "RECOMP: Improving Retrieval-Augmented LMs with Compression and Selective Augmentation," *arXiv preprint arXiv:2310.04408*, 2023.

[22] T. Hwang *et al.*, "DSLR: Document Refinement with Sentence-Level Re-ranking and Reconstruction to Enhance Retrieval-Augmented Generation," in *Proc. 3rd Workshop on Knowledge Augmented Methods for NLP*, 2024.

[23] Z. Shen *et al.*, "GeAR: Graph-enhanced Agent for Retrieval-augmented Generation," in *Findings of the Assoc. Comput. Linguistics: ACL*, 2025.

[24] Y. Hu *et al.*, "GRAG: Graph Retrieval-Augmented Generation," in *Findings of the Assoc. Comput. Linguistics: NAACL*, 2025.

[25] S. Kim *et al.*, "ReGraphRAG: Reorganizing Fragmented Knowledge Graphs for Multi-Perspective Retrieval-Augmented Generation," in *Findings of the Assoc. Comput. Linguistics: EMNLP*, 2025.