

TSA-RAG: Structure-Preserving Retrieval-Augmented Generation via Tiered Document Graph

Kaiyuan Zhang¹ Chong Feng^{1,2,*} Ge Shi¹ Lingnan Xu¹ Bo Wang¹

¹School of Computer Science and Technology, Beijing Institute of Technology

²Key Laboratory of Intelligent Information Processing and Content Security,
Ministry of Industry and Information Technology, China

{zky, fengchong, gshi, bwang}@bit.edu.cn

nunnunko777@gmail.com

Abstract

Retrieval-Augmented Generation (RAG) effectively mitigates hallucinations and knowledge obsolescence in Large Language Models (LLMs). However, prevailing RAG systems typically retrieve evidence from discrete chunks or through decoupled structures, making it difficult to jointly preserve cross-document navigability and within-document coherence on complex tasks. To address this challenge, we propose **TSA-RAG** (Tiered-Structure-Aware RAG), a unified framework that recasts retrieval as structure-preserving navigation over a Tiered Document Graph (TDG). The TDG offers a joint retrieval space that explicitly couples relational topology with hierarchical skeletons, enabling seamless traversal across granularities. Building on this representation, TSA-RAG navigates evidence through a progressive *Expand–Route–Retrieve* paradigm, which expands query-focused subgraphs under path-consistency constraints, routes the query to structural entry points within relevant documents, and retrieves subtree-structured evidence preserving discourse continuity. Extensive experiments on seven QA datasets demonstrate that TSA-RAG consistently outperforms strong baselines while yielding more compact retrieval contexts. Quantitative analyses further confirm the scalability and robustness of our framework. Code is available at <https://github.com/ZKYQUQ/TSA-RAG>.

1 Introduction

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Guu et al., 2020; Borgeaud et al., 2022) has proven effective in mitigating hallucinations and knowledge obsolescence (Ji et al., 2023; Ram et al., 2023) in Large Language Models (LLMs) (Vaswani et al., 2017; Brown et al., 2020; Touvron et al., 2023; Reid et al., 2024). Despite

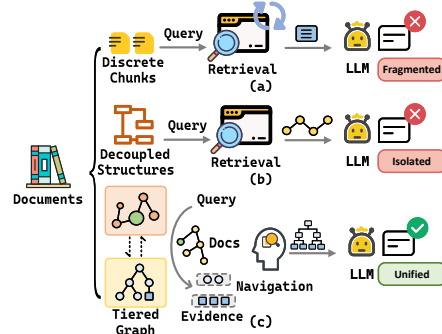


Figure 1: Comparison of RAG paradigms. Prevailing methods focus on either (a) discrete chunk retrieval, which yields fragmented evidence, or (b) decoupled structure construction with isolated retrieval dimensions. In contrast, (c) TSA-RAG introduces a unified tiered document graph for structure-preserving navigation.

rapid advancements, the performance of RAG on knowledge-intensive and multi-hop tasks remains bottlenecked by fundamental retrieval issues: *what* evidence is retrieved, and in *what form*.

Prevalent RAG systems (Jiang et al., 2023; Asai et al., 2024) predominantly operate on independent text chunks (Chen et al., 2024; Duarte et al., 2024; Yu et al., 2024). However, treating evidence as a bag of flat spans leads to retrieval noise and fragments the inherent logic of the corpus, ignoring how information is actually organized. Ideally, an effective evidence set should satisfy three desiderata: (i) *Navigability* across documents, enabling the traversal of corpus-wide relational links for multi-hop reasoning; (ii) *Coherence* within discourse, preserving local structural context rather than fragmented snippets; (iii) *Compactness* of retrieved context, avoiding redundant noise that impairs the generation phase.

We formulate the task of meeting these criteria as structure-preserving navigation, which recasts retrieval as progressive traversal from global relational topology to precise local evidence. However, existing approaches suffer from **structural decou-**

*Corresponding author.

pling, optimizing only a subset of these desiderata in isolation. Hierarchy-oriented methods (Sarthi et al., 2024; Jin et al., 2025; Xu et al., 2025) exploit within-document outlines to retrieve coherent nodes, yet remain confined to isolated contexts, failing to bridge cross-document reasoning paths. Conversely, graph-oriented methods (Edge et al., 2024; Fang et al., 2025; Huang et al., 2025; Chen et al., 2026) traverse entity-level connections for multi-hop retrieval but sacrifice within-document continuity, aggregating fragmented facts without preserving discourse structure. Consequently, prior systems force a trade-off between global navigability and local coherence. As summarized in Figure 1, they either retrieve discrete chunks (Figure 1a) or construct decoupled structure for retrieval (Figure 1b). Our objective, as depicted in Figure 1c, is to unify both dimensions into a single retrieval space that supports structure-preserving navigation.

To address this gap, we propose **TSA-RAG** (Tiered-Structure-Aware RAG), a unified framework that recasts retrieval as structure-preserving navigation over a **Tiered Document Graph (TDG)**. The TDG offers a joint retrieval space that explicitly couples relational topology with hierarchical skeletons, allowing seamless transition from corpus-level traversal to evidence-level localization. TSA-RAG then orchestrates navigation through a synergistic *Expand–Route–Retrieve* paradigm. (1) The *Expand* stage performs path-consistent graph expansion, constructing a query-focused subgraph from seed documents by leveraging path-consistency signals to prioritize semantically cohesive multi-hop chains. (2) The *Route* stage conducts coarse-grained document routing, which identifies promising documents together with structural entry points utilizing lightweight document skeletons, deciding where to look before scanning detailed content. (3) The *Retrieve* stage executes fine-grained knowledge retrieval, extracting pertinent evidence while preserving enclosing discourse structure for continuity, producing a compact and coherent context for generation.

Our contributions are as follows:

- We introduce the Tiered Document Graph (TDG), which resolves structural decoupling by unifying cross-document navigability and within-document coherence into a single retrieval space.
- We propose the *Expand–Route–Retrieve* navigation paradigm, which progressively tra-

verses the TDG from relational topology to precise evidence.

- Experiments show that TSA-RAG outperforms strong baselines, improving F1 by +3.2 on knowledge-inductive QA and up to +5.0 on multi-hop QA over structure-based methods, producing more compact retrieval contexts.

2 Related Work

2.1 Naive Retrieval-Augmented Generation

Naive RAG methods (Lewis et al., 2020; Guu et al., 2020; Izacard and Grave, 2021; Ram et al., 2023) segment external corpora into fixed-length chunks and leverage dense retrievers to retrieve relevant passages (Guu et al., 2020; Borgeaud et al., 2022; Izacard et al., 2023; Shi et al., 2024). While effective at mitigating hallucinations in knowledge-intensive tasks, these methods operate on isolated text fragments, limiting the acquisition of coherent and comprehensive knowledge (Zhuang et al., 2024; Cheng et al., 2025; Wu et al., 2025).

2.2 Iterative and Adaptive RAG

To address the limitations of fixed single-pass retrieval, iterative methods (Kim et al., 2023; Wang et al., 2025; Li et al., 2025; Agrawal et al., 2025) progressively decompose complex queries into sub-questions for multi-round evidence accumulation. For instance, CoRAG (Wang et al., 2025) trains models for iterative query decomposition and explores alternative reasoning paths via diverse decoding strategies. Adaptive methods (Jiang et al., 2023; Asai et al., 2024; Jeong et al., 2024; Guan et al., 2025) further invoke retrieval on demand, balancing external and parametric knowledge while mitigating noise. SELF-RAG (Asai et al., 2024) integrates retrieval with generation through a self-reflection mechanism that executes on-demand queries and evaluates generated content. However, these approaches focus solely on when and what to retrieve, leaving the retrieval space flat and neglecting both relational topology between documents and hierarchical skeletons within them.

2.3 Structure-Based RAG

Recent research has shifted towards explicitly modeling corpus structures to enhance retrieval. Hierarchy-oriented methods (Sarthi et al., 2024; Jin et al., 2025; Xu et al., 2025) exploit within-document structures to capture holistic semantics.

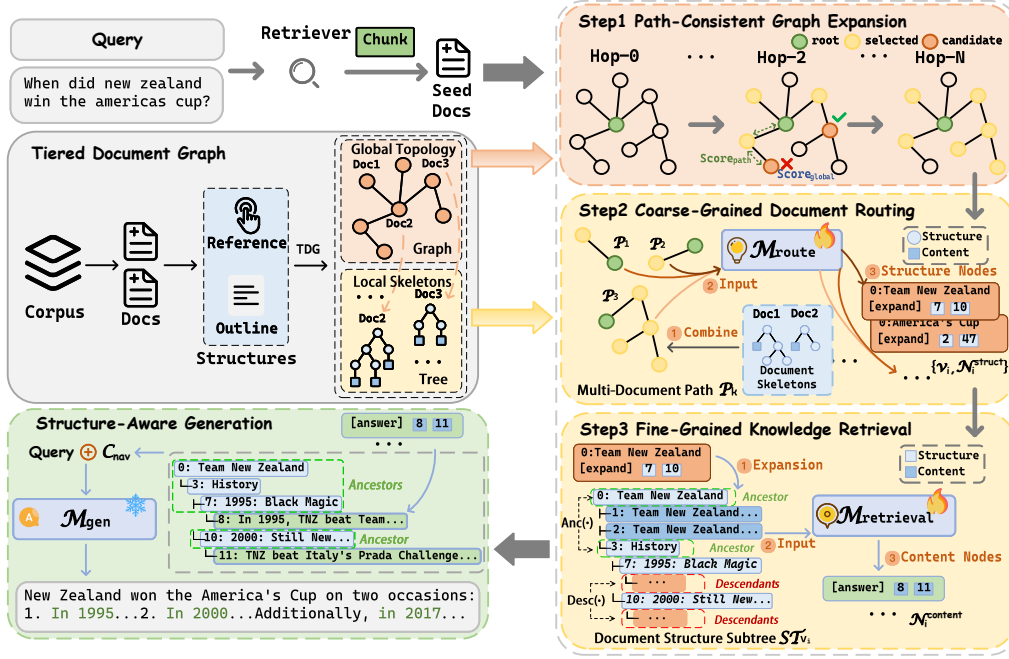


Figure 2: Overview of TSA-RAG. The Tiered Document Graph (TDG) couples a global document-link graph with per-document hierarchical trees (middle-left). During inference, TSA-RAG navigates evidence via the *Expand-Route-Retrieve* paradigm (right): *Expand* builds query-focused subgraphs from seed documents; *Route* identifies relevant documents and structural entry points via lightweight skeletons; *Retrieve* extracts subtree-structured evidence to form the compact context C_{nav} for generation (bottom-left).

LongRefiner (Jin et al., 2025) transforms documents into structured trees for node-level scoring, and RDR² (Xu et al., 2025) introduces a routing mechanism for dynamic hierarchical passage traversal. While effective at preserving local coherence, they remain confined to isolated documents, overlooking cross-document navigability. Graph-oriented methods (Edge et al., 2024; Fang et al., 2025) construct knowledge graphs to capture entity-level connections, supporting multi-hop retrieval. GraphRAG (Edge et al., 2024) partitions the graph into hierarchical communities and aggregates summaries for holistic generation. Recent works further explore hierarchical index organization (Huang et al., 2025) and relational path pruning (Chen et al., 2026). However, by operating over flat entity-level graphs, these methods ignore the hierarchical discourse structure within documents, producing decontextualized evidence that hinders coherent reasoning. Existing paradigms thus face a trade-off between global navigability and local coherence. TSA-RAG resolves this by establishing a unified retrieval space that couples cross-document topology with within-document skeletons, enabling structure-preserving navigation toward compact, subtree-structured evidence.

3 Methodology

Conventional RAG systems retrieve disjoint text chunks $\mathcal{C}$ from a corpus $\mathcal{D}$, ignoring the logic of the corpus. As shown in Figure 2, we address this by introducing the Tiered Document Graph (TDG), which offers a unified retrieval space that underlies our framework (Section 3.1). Building on TDG, we present the *Expand-Route-Retrieve* workflow (Section 3.2) comprising three stages: Path-Consistent Graph Expansion, Coarse-Grained Document Routing, and Fine-Grained Knowledge Retrieval. Finally, we describe Structure-Aware Generation and the training strategy (Section 3.3).

3.1 Tiered Document Graph

We recast retrieval as structure-preserving navigation over a Tiered Document Graph $\mathcal{H} = (\mathcal{G}, \{\mathcal{T}_v\}_{v \in \mathcal{V}})$, which unifies the retrieval space by coupling global relational topology with local hierarchical skeletons. Rather than independent dimensions, the two layers provide complementary structural signals for navigation, where $\mathcal{G}$ supports cross-document traversal, while each $\mathcal{T}_v$ supports within-document evidence localization.

Global Relational Topology. Documents naturally exhibit explicit interconnections, forming

meaningful dependencies and reasoning paths. We model this as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}_{\text{rel}})$, where each node $v_i \in \mathcal{V}$ represents a document $d_i \in \mathcal{D}$. The edge set $\mathcal{E}_{\text{rel}}$ encapsulates the relational connectivity, where edge (v_i, v_j) indicates a semantic link from document d_i to d_j . For each document d_i , we identify hyperlinks and citations pointing to other documents and instantiate them as directed edges. We further establish inverse edges where semantic relevance holds, yielding a densely connected network that supports multi-hop reasoning beyond single-document boundaries.

Local Hierarchical Skeletons. Documents contain inherent discourse structure, characterized by logical connectives that define relationships between fine-grained passages. For each document node $v_i \in \mathcal{V}$, we construct a hierarchical tree $\mathcal{T}_{v_i} = (\mathcal{N}_i, \mathcal{E}_{\text{hier}})$, where $\mathcal{N}_i$ represents a set of heterogeneous nodes and $\mathcal{E}_{\text{hier}}$ defines subordination and coordination relationships. The nodes are classified into two distinct types: (i) *Structure Nodes* (e.g., section headings) that serve as high-level thematic outlines; and (ii) *Content Nodes* (e.g., paragraphs) that encapsulate textual details. To construct $\mathcal{T}_{v_i}$, we leverage explicit outlines, encoded by formatting markers, to parse each document into a tree rooted at the title, branching into section headers as intermediate structure nodes, and terminating with paragraphs as leaf content nodes.

3.2 Structure-Preserving Navigation

Building on $\mathcal{H}$, we present a three-stage *Expand–Route–Retrieve* workflow that progressively narrows the retrieval scope through corpus-level graph traversal, document-level structural routing, and passage-level evidence extraction.

Path-Consistent Graph Expansion. Given seed documents $\mathcal{V}_{\text{seed}} \subset \mathcal{V}$ retrieved by a dense retriever, we expand query-focused subgraphs of closely related documents by traversing $\mathcal{G}$. The process is governed by a hybrid scoring mechanism that balances query alignment with path integrity.

Let $p = (v_0, v_1, \dots, v_t)$ denote an expansion path, where $v_0 \in \mathcal{V}_{\text{seed}}$ is the root node and each consecutive edge $(v_{k-1}, v_k) \in \mathcal{E}_{\text{rel}}$. The relevance of a candidate node v_t is evaluated based on two complementary dimensions.

To ensure the expansion remains focused on the intent of query q , we compute semantic similarity

between v_t and q as *global relevance score*:

$$s_{\text{global}}(v_t, q) = \text{sim}(\mathbf{e}_{v_t}, \mathbf{e}_q) \quad (1)$$

where $\mathbf{e}_{v_t}$ and $\mathbf{e}_q$ denote dense embeddings, and $\text{sim}(\cdot, \cdot)$ is instantiated as cosine similarity. We further incorporate a local perspective to capture cross-document semantic coherence and fine-grained dependencies. For an expansion path p , we compute the semantic similarity between each consecutive pair of neighboring nodes and define *path consistency* score as the cumulative product:

$$s_{\text{path}}(p) = \prod_{k=1}^t \text{sim}(\mathbf{e}_{v_{k-1}}, \mathbf{e}_{v_k}) \quad (2)$$

The *hybrid expansion score* integrates both dimensions for a candidate node v_t :

$$s_{\text{exp}}(v_t | p, q) = \alpha \cdot s_{\text{global}}(v_t, q) + (1 - \alpha) \cdot s_{\text{path}}(p) \quad (3)$$

where $\alpha \in [0, 1]$ controls the trade-off between global query alignment and local path coherence.

Starting from a seed node $v_{\text{root}} \in \mathcal{V}_{\text{seed}}$, we perform beam-style expansion by scoring neighbors of the currently retained nodes with s_{exp} and keeping the top- K candidates for the next hop, yielding an expanded subgraph $\mathcal{S}_{\text{sub}}$ that aggregates all retained nodes across hops. The multiplicative formulation of s_{path} accumulates pairwise coherence along the expansion path, thereby penalizing paths with weak successive semantic alignment and mitigating semantic drift. The complete expansion procedure is detailed in Appendix C (Algorithm 1).

Coarse-Grained Document Routing. To unfold the graph topology, we decompose $\mathcal{S}_{\text{sub}}$ into distinct multi-document paths, where each first-hop neighbor of v_{root} serves as a pivot, and a path $\mathcal{P}_k$ is formed by the seed, the pivot, and all nodes reachable from the pivot within $\mathcal{S}_{\text{sub}}$. To abstract path semantics while retaining structural awareness, we construct a *Document Skeleton* $\hat{d}_v$ for each node v , comprising its title, introduction, and structure nodes from $\mathcal{T}_v$, and represent the k -th path as:

$$\mathcal{P}_k = [\hat{d}_{v_{\text{root}}}, \hat{d}_{v_1^{(k)}}, \dots, \hat{d}_{v_m}] \quad (4)$$

where $v_1^{(k)}$ is the k -th neighbor of v_{root} , and $v_2, \dots, v_m$ are subsequent nodes expanded from $v_1^{(k)}$.

Drawing on the reading strategy of skimming structural cues before diving into full content, we

employ a specialized router model $\mathcal{M}_{\text{route}}$ to process these structured paths. Given each structured path $\mathcal{P}_k$ as input, $\mathcal{M}_{\text{route}}$ jointly identifies relevant documents and their internal structure nodes under which fine-grained evidence is likely encapsulated:

$$\{(v_i, \mathcal{N}_i^{\text{struct}})\} = \mathcal{M}_{\text{route}}(q, \mathcal{P}_k) \quad (5)$$

where v_i is a selected document and $\mathcal{N}_i^{\text{struct}} \subseteq \mathcal{N}_i$ is the set of structure nodes serving as entry points for subsequent fine-grained retrieval.

Fine-Grained Knowledge Retrieval. For each selected document v_i , we locate $\mathcal{N}_i^{\text{struct}}$ within the hierarchical tree $\mathcal{T}_{v_i}$ and apply a dual-direction expansion strategy. (i) *Descendant Expansion* incorporates all descendant nodes of $\mathcal{N}_i^{\text{struct}}$ to capture complete details, including nested subsections and leaf paragraphs. (ii) *Ancestor Preservation* retains the path from root to $\mathcal{N}_i^{\text{struct}}$ for logical coherence and contextual scope. We further incorporate nodes within the introduction section to ensure comprehensive semantic coverage.

This operation yields a pruned subtree $\mathcal{ST}_{v_i}$:

$$\mathcal{ST}_{v_i} = \mathcal{N}_i^{\text{struct}} \cup \text{Desc}(\mathcal{N}_i^{\text{struct}}) \cup \text{Anc}(\mathcal{N}_i^{\text{struct}}) \cup \mathcal{N}_i^{\text{intro}}, \quad (6)$$

where $\text{Desc}(\cdot)$ and $\text{Anc}(\cdot)$ return all descendant and ancestor nodes of a given set, respectively.

Mirroring the human behavior of intensive reading, a retrieval model $\mathcal{M}_{\text{retrieval}}$ analyzes the hierarchical context within $\mathcal{ST}_{v_i}$ to identify specific content nodes that contain pertinent evidence:

$$\mathcal{N}_i^{\text{content}} = \mathcal{M}_{\text{retrieval}}(q, \mathcal{ST}_{v_i}) \quad (7)$$

where $\mathcal{N}_i^{\text{content}}$ represents the set of extracted content nodes, constituting fine-grained evidence for grounding generation.

3.3 Generation and Training

Structure-Aware Generation. To preserve structural integrity, we aggregate all retrieved content nodes $\mathcal{N}_i^{\text{content}}$ together with their ancestor nodes to form a compact and coherent navigation context:

$$\mathcal{C}_{\text{nav}} = \bigcup_i (\mathcal{N}_i^{\text{content}} \cup \text{Anc}(\mathcal{N}_i^{\text{content}})) \quad (8)$$

The context is then linearized and fed into the generator $\mathcal{M}_{\text{gen}}$ to produce the final response:

$$y = \mathcal{M}_{\text{gen}}(q \oplus \mathcal{C}_{\text{nav}}) \quad (9)$$

where $\oplus$ denotes the concatenation operation.

Training Strategy. We construct training data sequentially following the inference pipeline. A powerful proprietary model serves as the teacher and is prompted with chain-of-thought (CoT) instructions to annotate structural labels. It first identifies relevant documents and structure nodes from multi-document paths to yield $\mathcal{D}_{\text{routing}}$, then extracts content nodes from subtrees to yield $\mathcal{D}_{\text{retrieval}}$.

$\mathcal{M}_{\text{route}}$ and $\mathcal{M}_{\text{retrieval}}$ are trained independently by maximizing the likelihood of the target labels:

$$\max_{\theta} \mathbb{E}_{(q, \mathcal{I}, y) \sim \mathcal{D}_{\tau}} \log p_{\theta}(y \mid q, \mathcal{I}) \quad (10)$$

where $\tau \in \{\text{routing}, \text{retrieval}\}$ denotes the task type, $\mathcal{I}$ represents the task-specific structured input ($\mathcal{P}_k$ or $\mathcal{ST}_{v_i}$), and y is the target label.

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate on seven QA datasets across three types. (1) Knowledge-Inductive QA: Following the widely adopted ALCE benchmark (Gao et al., 2023), we evaluate on ASQA (Stelmakh et al., 2022) for ambiguous factoid questions requiring multi-faceted answers, QAMPARI (Amouyal et al., 2023) for list-based entity questions spanning multiple passages, and ELI5 (Fan et al., 2019) for open-ended explanatory questions from Reddit. (2) Multi-Hop QA: HotpotQA (Yang et al., 2018) and 2WikiMultiHopQA (Ho et al., 2020), requiring multi-step cross-document reasoning. (3) Single-Hop QA: TriviaQA (Joshi et al., 2017) and PopQA (Mallen et al., 2023) for factual retrieval. Dataset details are provided in Appendix A.1.

Following prior work (Gao et al., 2023), we utilize the Wikipedia 2018 dump (Karpukhin et al., 2020) as the retrieval corpus for these datasets.

Evaluation Metrics. For correctness metrics, we use EM and D-F1¹ for ASQA, F1-5, REC-5 and PRE for QAMPARI, and CLAIM score² for ELI5. For short-form tasks, we report F1 score. We additionally report MAUVE for long-form tasks to evaluate fluency. Full metric definitions are provided in Appendix A.2.

Baselines. We compare TSA-RAG against four categories of methods. (1) *No-retrieval baselines* evaluate intrinsic LLM knowledge without retrieved context, covering open-source models

¹D-F1 (Disambig-F1) measures character-level F1 over disambiguated questions using a RoBERTa-based model.

²CLAIM score decomposes outputs into sub-claims and verifies each via NLI model.

Table 1: Results on knowledge-inductive QA datasets. [†] denotes proprietary models. Bold indicates the best non-proprietary result. **Gray bold** indicates the best proprietary result when it exceeds all non-proprietary models. * denotes results reported in RDR².

Methods	ASQA			QAMPARI			ELI5	
	EM	D-F1	MAUVE	F1-5	REC-5	PRE	CLAIM	MAUVE
<i>No Retrieval</i>								
ChatGPT [†]	34.1	27.4	18.2	17.7	18.5	20.8	23.6	14.4
GPT-4o [†]	41.4	33.9	23.3	26.0	28.7	30.5	26.5	20.2
DeepSeek-V3 [†]	43.0	33.1	21.9	23.4	28.7	23.3	26.3	15.6
Llama-2-13b	24.7	19.3	13.4	14.9	16.4	14.3	14.7	21.9
<i>Naive RAG</i>								
ChatGPT [†]	42.8	34.4	57.0	22.2	22.1	27.9	22.3	12.2
GPT-4o [†]	47.0	36.5	39.9	23.7	23.2	31.0	24.5	17.9
DeepSeek-V3 [†]	48.8	37.4	36.7	23.2	24.6	27.3	26.6	14.9
Llama-2-13b	36.5	26.9	31.4	21.0	22.0	21.6	14.9	20.8
<i>Adaptive & Advanced RAG</i>								
SELF-RAG*	37.5	27.5	77.7	6.5	9.0	6.4	11.8	42.8
OPEN-RAG*	39.9	24.0	17.2	2.5	3.9	2.3	11.9	19.1
FRONT*	41.5	—	<u>76.1</u>	—	11.9	22.6	9.3	<u>34.4</u>
<i>Structure-Based RAG</i>								
LongRefiner	33.5	26.2	60.8	13.9	14.2	15.9	14.5	29.0
RDR ^{2*}	<u>41.7</u>	<u>31.6</u>	61.2	<u>23.2</u>	<u>24.3</u>	<u>25.0</u>	<u>15.9</u>	22.8
TSA-RAG	43.4	32.5	62.7	26.4	29.3	27.3	16.8	23.8

Table 2: F1 results on four short-form QA datasets.

Methods	TQA	PopQA	HotpotQA	2Wiki
<i>No Retrieval</i>				
Llama-3.1-8B	63.7	28.4	28.1	31.7
<i>Retrieval-Based</i>				
Llama-3.1-8B	70.5	47.4	41.7	32.3
LongRefiner	<u>73.0</u>	45.9	<u>45.8</u>	35.0
RDR ²	72.5	<u>48.8</u>	44.8	<u>36.7</u>
TSA-RAG	73.8	50.7	49.8	37.2

(Llama-2-13b, Llama-3.1-8B) and proprietary models (ChatGPT (Ouyang et al., 2022), GPT-4o (OpenAI, 2024), DeepSeek-V3 (DeepSeek-AI, 2024)). (2) *Naive RAG* follows the standard retrieve-then-generate pipeline, where the top- k retrieved chunks are directly fed to the generator. (3) *Adaptive and Advanced RAG* methods include SELF-RAG (Asai et al., 2024), which introduces self-reflective tokens to guide retrieval; OPEN-RAG (Islam et al., 2024), which learns to retrieve from open-domain knowledge; and FRONT (Huang et al., 2024), which grounds generation with attribution. (4) *Structure-Based RAG* methods leverage document structure for retrieval, including LongRefiner (Jin et al., 2025), which scores and selects relevant nodes

from document trees, and RDR² (Xu et al., 2025), which performs iterative routing over hierarchical structures. Detailed baseline descriptions are provided in Appendix A.3.

4.2 Implementation Details

We sample 2,200 instances from the training split of each of ASQA, QAMPARI, and ELI5 to construct training data. DeepSeek-V3 serves as the teacher model and is prompted with Chain-of-Thought (CoT) (Wei et al., 2022) instructions to generate structural labels, yielding 32,096 routing samples and 36,042 retrieval samples. Both $\mathcal{M}_{\text{route}}$ and $\mathcal{M}_{\text{retrieval}}$ are initialized from Llama-3.1-8B-Instruct (Dubey et al., 2024) and fine-tuned via LoRA ($r = 8$, $\alpha = 16$) (Hu et al., 2022) for 5 epochs with a learning rate of $1e-5$. During inference, we utilize the Contriever-MS MARCO (Izacard et al., 2022) model to retrieve top-5 initial seed documents. The graph expansion parameters are set to $\alpha = 0.6$, beam width $K = 5$, and max hops $D_{\text{max}} = 5$. We use Llama-2-13b-chat (Touvron et al., 2023) as the reader for knowledge-inductive tasks, following the setup of SELF-RAG for fair comparison. For short-form tasks, we use Llama-3.1-8B-Instruct as the reader, following the setup of LongRefiner and RDR².

Table 3: Ablation studies of components and structures.

Methods	EM (HIT)	D-F1
TSA-RAG (Full)	43.4 (17.3)	32.5
<i>Component Ablation</i>		
w/o Knowledge Retrieval	37.8 (13.8)	29.2
w/o Graph Expansion	40.9 (15.1)	30.6
w/o Document Routing	41.3 (15.4)	32.1
<i>Structure Ablation</i>		
w/o Global Topology	40.9 (15.1)	30.6
w/o Local Skeletons	42.1 (16.2)	32.3
w/o Topology & Skeletons	40.8 (15.5)	31.9

4.3 Main Results

As shown in Tables 1 and 2, TSA-RAG consistently outperforms strong baselines across different tasks.

Substantial improvement over naive baselines. Compared to no-retrieval baselines, TSA-RAG achieves notable improvements over the backbone Llama-2-13b, with relative gains in correctness of 75.7%, 77.2%, and 14.3% on three knowledge-inductive datasets, respectively. It also surpasses a range of proprietary models on ASQA and QAMPARI. Under the naive RAG setting, proprietary models (e.g., GPT-4o, DeepSeek-V3) show performance degradations on QAMPARI due to noisy retrieval from complex questions, while TSA-RAG outperforms them, demonstrating strong robustness in filtering irrelevant information.

Advantages of structure-aware retrieval. Among advanced RAG methods, TSA-RAG achieves superior performance on knowledge-inductive tasks, where structure-based methods show clear advantages over unstructured ones on QAMPARI and ELI5. Within structure-based methods, TSA-RAG consistently outperforms RDR² by leveraging relational topology for cross-document reasoning beyond isolated document trees. The modest CLAIM gains on ELI5 are consistent across Wikipedia-based methods, reflecting the domain gap between static encyclopedic evidence and open-ended Reddit questions.

Multi-hop gains from global topology. Table 2 shows that TSA-RAG achieves the best F1 on four short-form datasets. Compared with retrieval-augmented Llama-3.1-8B, TSA-RAG improves F1 by 4.7% and 7.0% on two single-hop datasets, and by 19.4% and 15.2% on two multi-hop datasets. The substantial gains in multi-hop settings highlight the effectiveness of the graph expansion algorithm in capturing multi-step reasoning paths that standard retrieval fails to uncover.

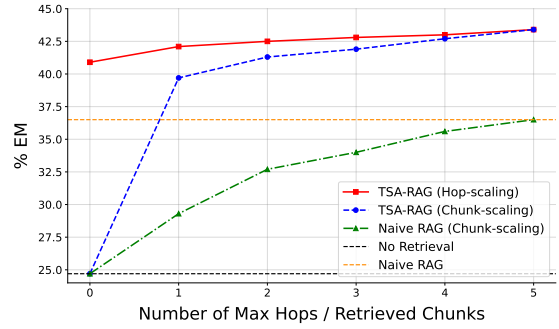


Figure 3: Scaling of max hops and retrieved chunks.

4.4 Ablation Study

Table 3 reports the ablation results on ASQA.

Workflow Component ablation. We first analyze the necessity of each stage in our framework. (1) **w/o Knowledge Retrieval** directly uses the subtrees constructed from routing results as context for generation, without fine-grained node selection. This causes the largest performance drop, with EM falling to 37.8, underscoring the retriever’s essential role in filtering structural noise and extracting precise evidence. (2) **w/o Graph Expansion** bypasses the expansion phase and constructs paths using only the initially retrieved seed documents. The notable performance decline confirms that relying solely on dense retrieval misses critical global relational dependencies, which our expansion algorithm effectively captures. (3) **w/o Document Routing** skips the routing stage and feeds full document trees directly into the retriever. The moderate performance drop demonstrates that the retriever remains robust when handling unpruned trees, while the router still contributes by improving efficiency and precision through pruning irrelevant branches.

Structural Representation ablation. We further investigate the structures within TDG. (1) **w/o Global Topology** removes links (equivalent to w/o Graph Expansion) and isolates documents, preventing cross-document navigability and lowering EM to 40.9. (2) **w/o Local Skeletons** flattens the document tree by removing hierarchical headings, reducing it to a two-level representation with document title as root and all paragraphs as direct children, which eliminates structural cues for routing and retrieval phases. This reduces EM to 42.1, suggesting that explicit hierarchy aids in locating fine-grained evidence within long contexts. (3) **w/o Topology & Skeletons** strips both structures. TSA-RAG still substantially outperforms naive RAG, verifying the robustness of our learned router and retriever be-

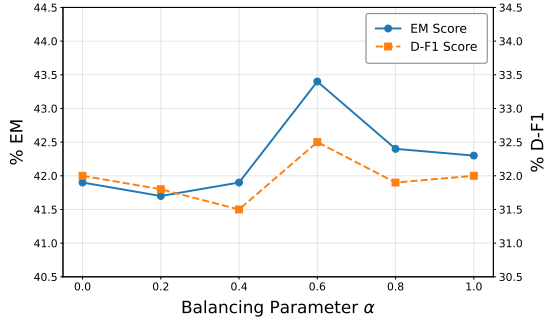


Figure 4: Impact of balancing parameter α .

yond structure-rich scenarios. The remaining gap to the full model further demonstrates the value of structural signals in providing discriminative cues.

Additional ablations separately assess the effects of structural representation and intermediate-module supervision, and evaluate structures induced from generic semantic cues when native hyperlinks and headings are unavailable (Appendices B.6 and B.8).

4.5 Quantitative Analysis

Test-Time Scaling. We conduct analyses on the ASQA dataset by varying the number of initial retrieved chunks and max hops parameter $D_{\max}$ in the graph expansion algorithm. **(i) Retrieved Chunks Scaling.** Increasing retrieved chunks yields diminishing returns for both TSA-RAG and naive RAG, with the largest gain occurring when retrieving the top-1 chunk. Notably, TSA-RAG at this setting outperforms naive RAG with top-5 chunks, demonstrating its effectiveness. **(ii) Max Hops Scaling.** We vary the value of $D_{\max}$ from 0 to 5, where 0 hop corresponds to ablating the graph expansion stage. TSA-RAG consistently outperforms naive RAG across all settings and improves steadily, validating the benefit of multi-hop exploration. These results highlight the flexibility of our framework, enabling inference-time trade-offs between computational cost and performance.

Parameter Sensitivity. We verify the impact of the balancing parameter α on the ASQA dataset. As shown in Figure 4, the performance exhibits a convex trend, peaking around $\alpha = 0.6$. Extremes lead to degradation: $\alpha = 0$ risks semantic drift via unconstrained expansion, while $\alpha = 1$ isolates documents and ignores topological cues. We set $\alpha = 0.6$ for all main experiments.

Robustness. As shown in Figure 5, we evaluate TSA-RAG’s robustness with different retrievers

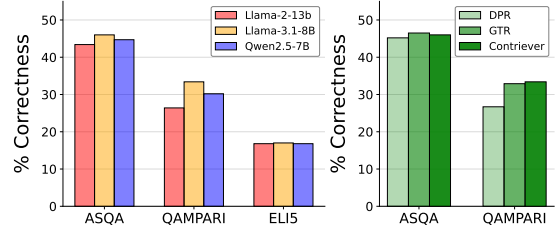


Figure 5: Robustness analysis of readers and retrievers.

Table 4: Efficiency analysis on ASQA.

Method	EM	Intermediate (In/Out)	Reader (In/Out)
Naive RAG	36.5	— / —	0.753 / 0.081
RDR ²	41.7	2.045 / 0.037	0.739 / 0.093
TSA-RAG	43.4	2.245 / 0.043	0.611 / 0.090

and readers. TSA-RAG consistently outperforms naive RAG across LLaMA and Qwen families, demonstrating strong adaptability to varying model architectures and scales. With three distinct retrievers and fixed Llama-2-13b, performance remains stable, confirming robustness to retriever choice.

Efficiency. We evaluate the efficiency of TSA-RAG by analyzing average token consumption (k) in a parallel inference setting. Starting from the top-5 initial seed documents, our framework expands to an average of 15 multi-document paths and 8 pruned document subtrees per query. As shown in Table 4, TSA-RAG incurs intermediate costs from structure-preserving navigation, processing 1.523k/0.033k tokens for routing and 0.722k/0.010k for retrieval, resulting in a marginally higher aggregated overhead compared to RDR² (2.045k/0.037k). Crucially, this process yields a more compact context for the reader, surpassing both RDR² and naive RAG, which demonstrates that leveraging global topology facilitates more focused within-document selection.

5 Conclusion

This work presents TSA-RAG, a unified framework that addresses the structural decoupling inherent in existing RAG systems by recasting retrieval as structure-preserving navigation over a Tiered Document Graph (TDG). By coupling relational topology with hierarchical skeletons, TDG enables the progressive *Expand–Route–Retrieve* paradigm to jointly optimize cross-document navigability and within-document coherence. Experiments on seven QA datasets demonstrate consistent improvements over strong baselines while maintaining com-

pact reader contexts, and quantitative analyses confirm the scalability and robustness of our framework. Beyond the specific framework, this work highlights that modeling document structures is a principled and effective approach to strengthening evidence acquisition in RAG systems for complex, knowledge-intensive tasks. For future work, we aim to extend TSA-RAG to structure-free corpora and explore domain adaptation strategies for broader applicability.

Limitations

TSA-RAG has two primary limitations.

Structural signal availability. TSA-RAG leverages explicit cross-document references (e.g., hyperlinks or citations) and formatting markers (e.g., section headings) when available. While our ablation (w/o Topology & Skeletons in Table 3) and the Generic-TDG experiment in Appendix B.8 show that the framework degrades gracefully and still outperforms naive RAG when native structures are unavailable, structural signals remain important for fully exploiting TDG’s navigation capabilities. Future work could explore inducing global topology via named-entity overlap or semantic similarity edges, and extracting local skeletons via learned outline prediction or LLM-assisted document parsing.

Domain generalization. The routing and retrieval models are trained on Wikipedia-based QA data, which may limit cross-domain generalization to specialized corpora such as biomedical literature or legal documents. Investigating few-shot adaptation and retrieval-augmented data synthesis to transfer the trained models to new domains remains an important direction for future work.

Ethical Considerations

The datasets and corpora used in our experiments are open-access resources, and no private or sensitive information is involved. The models and APIs adopted in this work are obtained from publicly accessible platforms and used in accordance with their respective terms and policies. Our study focuses on improving retrieval and reasoning over public knowledge corpora, and we do not foresee direct ethical concerns arising from this research.

References

Rishabh Agrawal, Murtaza Asrani, Hadi Youssef, and Apurva Narayan. 2025. Scmrage: Self-corrective mul-

ti-hop retrieval augmented generation system for llm agents. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS ’25, pages 50–58, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Samuel Amouyal, Tomer Wolfson, Ohad Rubin, Ori Yoran, Jonathan Herzig, and Jonathan Berant. 2023. [QAMPARI: A benchmark for open-domain questions with many answers](#). In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 97–110, Singapore. Association for Computational Linguistics.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. [Self-rag: Learning to retrieve, generate, and critique through self-reflection](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, and 9 others. 2022. [Improving language models by retrieving from trillions of tokens](#). In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, Proceedings of Machine Learning Research, pages 2206–2240. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Boyu Chen, Zirui Guo, Zidan Yang, Yuluo Chen, Junze Chen, Zhenghao Liu, Chuan Shi, and Cheng Yang. 2026. [Pathrag: Pruning graph-based retrieval augmented generation with relational paths](#). In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 30183–30191. AAAI Press.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024. [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.

- Rong Cheng, Jinyi Liu, Yan Zheng, Fei Ni, Jiazhen Du, Hangyu Mao, Fuzheng Zhang, Bo Wang, and Jianye Hao. 2025. [DualRAG: A dual-process approach to integrate reasoning and retrieval for multi-hop question answering](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31877–31899, Vienna, Austria. Association for Computational Linguistics.
- DeepSeek-AI. 2024. [Deepseek-v3 technical report](#). *CoRR*, abs/2412.19437.
- André V. Duarte, João DS Marques, Miguel Graça, Miguel Freire, Lei Li, and Arlindo L. Oliveira. 2024. [LumberChunker: Long-form narrative document segmentation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6473–6486, Miami, Florida, USA. Association for Computational Linguistics.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. [From local to global: A graph RAG approach to query-focused summarization](#). *CoRR*, abs/2404.16130.
- Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. [ELI5: long form question answering](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3558–3567. Association for Computational Linguistics.
- Jinyuan Fang, Zaiqiao Meng, and Craig MacDonald. 2025. [Kirag: Knowledge-driven iterative retriever for enhancing retrieval-augmented generation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, pages 18969–18985. Association for Computational Linguistics.
- Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. [Enabling large language models to generate text with citations](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6465–6488. Association for Computational Linguistics.
- Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. 2025. [Deeprag: Thinking to retrieval step by step for large language models](#). *CoRR*, abs/2502.01142.
- Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. [Hipporag: Neurobiologically inspired long-term memory for large language models](#). In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. [Retrieval augmented language model pre-training](#). In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. [Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps](#). In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [Lora: Low-rank adaptation of large language models](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Haoyu Huang, Yongfeng Huang, Junjie Yang, Zhenyu Pan, Yongqiang Chen, Kaili Ma, Hongzhi Chen, and James Cheng. 2025. [Retrieval-augmented generation with hierarchical knowledge](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 6044–6060. Association for Computational Linguistics.
- Lei Huang, Xiaocheng Feng, Weitao Ma, Yuxuan Gu, Weihong Zhong, Xiachong Feng, Weijiang Yu, Weihua Peng, Duyu Tang, Dandan Tu, and Bing Qin. 2024. [Learning fine-grained grounded citations for attributed large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14095–14113, Bangkok, Thailand. Association for Computational Linguistics.
- Shayekh Bin Islam, Md. Asib Rahman, K. S. M. Tozammel Hossain, Enamul Hoque, Shafiq Joty, and Md. Rizwan Parvez. 2024. [Open-rag: Enhanced retrieval augmented reasoning with open-source large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, Findings of ACL, pages 14231–14244. Association for Computational Linguistics.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. [Unsupervised dense information retrieval with contrastive learning](#). *Trans. Mach. Learn. Res.*, 2022.

- Gautier Izacard and Edouard Grave. 2021. [Leveraging passage retrieval with generative models for open domain question answering](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.
- Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24(1).
- Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. [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)*, pages 7036–7050, Mexico City, Mexico. Association for Computational Linguistics.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. [Survey of hallucination in natural language generation](#). *ACM Comput. Surv.*, 55(12).
- Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. [Active retrieval augmented generation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.
- Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yongkang Wu, Zhonghua Li, Ye Qi, and Zhicheng Dou. 2025. [Hierarchical document refinement for long-context retrieval-augmented generation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3502–3520, Vienna, Austria. Association for Computational Linguistics.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. [TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- 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.
- Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joon-suk Park, and Jaewoo Kang. 2023. [Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 996–1009, Singapore. Association for Computational Linguistics.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS ’20*, Red Hook, NY, USA. Curran Associates Inc.
- Rui Li, Quanyu Dai, Zeyu Zhang, Xu Chen, Zhenhua Dong, and Ji-Rong Wen. 2025. [Knowtrace: Bootstrapping iterative retrieval-augmented generation with structured knowledge tracing](#). In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2, KDD ’25*, pages 1470—1480, New York, NY, USA. Association for Computing Machinery.
- Alex Mullen, 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.
- OpenAI. 2024. [Gpt-4o system card](#). *CoRR*, abs/2410.21276.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’22*, Red Hook, NY, USA. Curran Associates Inc.
- Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. [In-context retrieval-augmented language models](#). *Transactions of the Association for Computational Linguistics*, 11:1316–1331.
- Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew M. Dai, Katie Millican, Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, and 34 others. 2024. [Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context](#). *CoRR*, abs/2403.05530.

- 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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. [REPLUG: Retrieval-augmented black-box language models](#). 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 8371–8384, Mexico City, Mexico. Association for Computational Linguistics.
- Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. [ASQA: Factoid questions meet long-form answers](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *CoRR*, abs/2307.09288.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Liang Wang, Haonan Chen, Nan Yang, Xiaolong Huang, Zhicheng Dou, and Furu Wei. 2025. [Chain-of-retrieval augmented generation](#). In *Advances in Neural Information Processing Systems*, volume 38, pages 59888–59915. Curran Associates, Inc.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’22*, Red Hook, NY, USA. Curran Associates Inc.
- Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. 2025. [Web-Walker: Benchmarking LLMs in web traversal](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10290–10305, Vienna, Austria. Association for Computational Linguistics.
- Lingnan Xu, Chong Feng, Kaiyuan Zhang, Liu Zhengyong, Wenqiang Xu, and Fanqing Meng. 2025. [Equipping retrieval-augmented large language models with document structure awareness](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 24608–24631. Association for Computational Linguistics.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. 2024. [Chain-of-note: Enhancing robustness in retrieval-augmented language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14672–14685, Miami, Florida, USA. Association for Computational Linguistics.
- Ziyuan Zhuang, Zhiyang Zhang, Sitao Cheng, Fangkai Yang, Jia Liu, Shujian Huang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. 2024. [EfficientRAG: Efficient retriever for multi-hop question answering](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3392–3411, Miami, Florida, USA. Association for Computational Linguistics.

A Implementation Details

A.1 Datasets

We provide detailed descriptions of the seven QA datasets used in our experiments, covering knowledge-inductive generation and short-form question answering.

ASQA (Stelmakh et al., 2022) is a long-form factoid QA dataset derived from AmbigQA. Each question is ambiguous and requires multiple short answers corresponding to different interpretations, together with a long-form answer that covers all valid interpretations.

QAMPARI (Amouyal et al., 2023) is a list-style factoid QA dataset based on Wikipedia. Each question expects a set of entity answers, typically requiring evidence from multiple passages across different documents.

ELI5 (Fan et al., 2019) is a long-form QA dataset collected from the Reddit forum “Explain Like I’m Five”. Most questions are how/why/what questions that require detailed, multi-passage answers.

HotpotQA (Yang et al., 2018) is a multi-hop QA dataset that requires reasoning over multiple

Wikipedia documents, typically involving two supporting facts.

2WikiMultiHopQA (Ho et al., 2020) is a multi-hop QA dataset constructed from Wikipedia and Wikidata, requiring cross-document reasoning over structured and unstructured knowledge.

TriviaQA (Joshi et al., 2017) contains question-answer pairs sourced from trivia quizzes and competitions, paired with supporting Wikipedia documents.

PopQA (Mallen et al., 2023) is a large-scale open-domain QA dataset covering a wide range of entity-centric factual questions, testing the retrieval of long-tail knowledge.

Dataset statistics. Table 5 summarizes the original split sizes of the seven datasets used in our experiments. Following RDR² (Xu et al., 2025), we sample at most 1,000 examples from each dataset for evaluation. ASQA contains 948 development examples, so we use the full development set. Across all datasets, only the question field is used for retrieval and generation, and Wikipedia 2018 is consistently used as the retrieval datastore. For training the router and retriever, we sample 2,200 instances from the training split of each knowledge-inductive dataset, namely ASQA, QAMPARI, and ELI5. Evaluation examples are not used for training.

A.2 Evaluation Metrics

We employ task-specific metrics to evaluate answer quality.

EM (Hit) is the exact match score that checks whether the predicted answer exactly matches any of the reference short answers.

D-F1 (Disambig-F1) measures answer informativeness on ASQA. It employs a RoBERTa model pre-trained on SQuAD 2.0 to evaluate how many disambiguated questions can be answered from the generated long-form response and computes character-level F1.

CLAIM assesses the factual coverage of generated answers. It utilizes sub-claims (automatically generated for each question in ELI5 via text-davinci-003) and determines whether the model’s output entails these sub-claims using a natural language inference (NLI) model, reporting the entailment accuracy.

F1-5 and **REC-5** are list-based precision and recall metrics for QAMPARI. Given the list-based answers, recall is considered 100% if the predic-

tion contains at least 5 correct entities; F1-5 is the corresponding precision-capped F1 score.

Accuracy and **F1** are the standard span-level accuracy and token-level F1 used for short-form QA datasets.

Additional metrics including ROUGE, MAUVE, and average answer length (LEN) are reported for long-form tasks to measure fluency and coverage.

A.3 Baseline Details

Here, we provide details of the Adaptive & Advanced RAG and Structure-Based RAG baselines.

Adaptive and Advanced RAG Methods. SELF-RAG (Asai et al., 2024) introduces a self-reflection mechanism via special tokens to integrate retrieval with generation, performing retrieval on demand and assessing the quality of generated outputs. OPEN-RAG (Islam et al., 2024) transforms an arbitrary dense LLM into a sparse mixture of experts model capable of self-reflection and handling complex reasoning tasks, including both single- and multi-hop queries. FRONT (Huang et al., 2024) introduces a two-stage training framework that teaches LLMs to select supporting quotes from retrieved sources and then condition the generation process on them, enabling attributed text generation.

Structure-Based RAG Methods. LongRefiner (Jin et al., 2025) transforms long documents into structured trees and employs a scoring strategy to select relevant nodes, which effectively improves the signal-to-noise ratio during retrieval. RDR² (Xu et al., 2025) develops hierarchical document structure trees and introduces a routing mechanism for dynamic navigation, enabling iterative passage selection and structure expansion to achieve flexible knowledge acquisition.

A.4 Training and Inference Details

All experiments are conducted on NVIDIA A100-PCIE-40GB GPUs.

Training details. We fine-tuned Llama-3.1-8B-Instruct for both document routing and knowledge retrieval tasks using LoRA (Hu et al., 2022). Specifically, we set `lora_rank=8`, `lora_alpha=16`, `gradient-accumulated batch size=8`, `learning rate=1e-5`, and trained for 5 epochs. The two models are trained independently.

Inference details. For the Llama-2-13b-chat reader model, the context window was set to 4,192 tokens. Context was populated by sequentially adding structured passages in descending order of

Table 5: Dataset statistics.

Dataset	Task type	Original split size
ASQA	Knowledge-inductive QA	Train 4,353 / Dev 948
QAMPARI	Knowledge-inductive QA	Train 61,911 / Dev 1,000 / Test 1,000
ELI5	Knowledge-inductive QA	Train 272,634 / Dev 9,812 / Test 24,512
TriviaQA	Single-hop QA	Train 87,622 / Dev 11,313 / Test 10,832
PopQA	Single-hop QA	Test 14,267
HotpotQA	Multi-hop QA	Train 90,447 / Dev 7,405
2WikiMultiHopQA	Multi-hop QA	Train 167,454 / Dev 12,576 / Test 12,576

their relevance scores. To ensure a fair comparison, maximum output length was set to 128 tokens for ASQA and 256 tokens for ELI5, QAMPARI, and other short-form datasets. For proprietary models including ChatGPT, GPT-4o, and DeepSeek-V3, we set temperature= 0.2 without length constraints. Unless otherwise specified, all reported results are obtained from a single run with fixed experimental settings.

A.5 TDG Statistics & Efficiency Details

TDG Statistics. Table 6 reports key statistics of the global topology layer of the Tiered Document Graph, constructed on the Wikipedia 2018 dump. The graph contains 5,821,544 document nodes and 59,313,862 directed edges, with an average out-degree of 10.19 (maximum 91,989). Of all documents, 94.35% have at least one outgoing link, and 5.65% are isolated nodes with no connections. For the local skeletons layer, a hierarchical Document Structure Tree (DST) is constructed for each document, decomposing it into a title, hierarchical section headings, and content paragraphs. On average, each DST contains 16.06 nodes (max 27,343) across a mean tree depth of 2.96 (max 25), comprising 3.50 title/heading nodes and 12.56 paragraph nodes per document.

Table 6: Global topology statistics of TDG.

Property	Value
# Document nodes	5,821,544
# Hyperlink edges	59,313,862
Avg. out-degree	10.19
Max out-degree	91,989
Documents with links (%)	94.35%
Isolated nodes (%)	5.65%

Offline Construction Cost. The TDG is constructed offline. DST construction for all 5.82M documents takes approximately 27 minutes using 8 CPU cores in parallel. Building the global graph (including Contriever embedding of all documents and edge insertion) requires approximately

15 hours on a single NVIDIA A100 GPU. Storage requirements are approximately 41 GB for the document structure trees, and 43 GB for the Neo4j graph (including pre-computed embeddings). This one-time preprocessing enables efficient inference for all subsequent queries without additional construction overhead.

Per-Stage Inference Efficiency. Under a parallel inference setting, each stage exploits within-query parallelism: graph expansion is parallelized over an average of 3.24 root documents per query (2.50 s/query); document routing over 15.22 multi-document paths (2.07 s/query); knowledge retrieval over 8 document subtrees (0.86 s/query); and reader generation proceeds as a single call (3.86 s/query). In terms of token consumption, TSA-RAG processes 1.523k/0.033k tokens for document routing and 0.722k/0.010k tokens for knowledge retrieval, yielding a more compact reader context (0.611k input tokens) compared to Naive RAG (0.753k) and RDR² (0.739k).

B More Details about Experimental Results

B.1 Extended Main Results

Table 7 presents the full main results on ASQA, QAMPARI and ELI5 with the additional generation-length metric (LEN) and ROUGE.

B.2 Extended Ablation Results

Table 8 presents the full ablation results on ASQA and QAMPARI with additional metrics, complementing the main ablation study in Table 3.

We describe the exact implementation of each ablation variant:

w/o Graph Expansion: We construct multi-document paths using only the initial seed documents retrieved by the dense retriever, where each path contains a single retrieved document. Subsequent routing and retrieval stages proceed as normal.

Table 7: Detailed experimental results on knowledge-inductive QA datasets including the generation length (LEN) metric. [†] denotes proprietary models. Bold indicates the best non-proprietary result. **Gray bold** indicates the best proprietary result when it exceeds all non-proprietary models. * denotes results reported in RDR².

Methods	ASQA					QAMPARI			ELI5		
	EM	D-F1	ROUGE	MAUVE	LEN	F1-5	REC-5	PRE	CLAIM	MAUVE	LEN
<i>No Retrieval</i>											
ChatGPT [†]	34.1	27.4	35.7	18.2	57.5	17.7	18.5	20.8	23.6	14.4	145.3
GPT-4o [†]	41.4	33.9	36.2	23.3	58.8	26.0	28.7	30.5	26.5	20.2	158.4
DeepSeek-V3 [†]	43.0	33.1	36.3	21.9	69.2	23.4	28.7	23.3	26.3	15.6	137.4
Llama-2-13b	24.7	19.3	35.1	13.4	65.9	14.9	16.4	14.3	14.7	21.9	140.2
<i>Naive RAG</i>											
ChatGPT [†]	42.8	34.4	38.0	57.0	51.1	22.2	22.1	27.9	22.3	12.2	141.5
GPT-4o [†]	47.0	36.5	38.4	39.9	68.4	23.7	23.2	31.0	24.5	17.9	154.7
DeepSeek-V3 [†]	48.8	37.4	37.5	36.7	74.2	23.2	24.6	27.3	26.6	14.9	132.4
Llama-2-13b	36.5	26.9	39.2	31.4	61.2	21.0	22.0	21.6	14.9	20.8	141.2
<i>Adaptive & Advanced RAG</i>											
SELF-RAG*	37.5	27.5	<u>39.2</u>	77.7	69.9	6.5	9.0	6.4	11.8	42.8	81.9
OPEN-RAG*	39.9	24.0	40.4	17.2	83.7	2.5	3.9	2.3	11.9	19.1	129.2
FRONT*	41.5	–	38.6	<u>76.1</u>	57.6	–	11.9	22.6	9.3	<u>34.4</u>	75.1
<i>Structure-Based RAG</i>											
LongRefiner	33.5	26.2	36.5	60.8	65.7	13.9	14.2	15.9	14.5	29.0	178.1
RDR ^{2*}	41.7	31.6	39.2	61.2	69.6	<u>23.2</u>	<u>24.3</u>	<u>25.0</u>	<u>15.9</u>	22.8	172.3
TSA-RAG	43.4	32.5	38.8	62.7	68.0	26.4	29.3	27.3	16.8	23.8	177.6

Table 8: Full ablation study on ASQA and QAMPARI with complete metrics.

Methods	ASQA					QAMPARI			
	EM(HIT)	D-F1	ROUGE	MAUVE	LEN	F1-5(F1)	REC-5(REC)	PRE	
TSA-RAG (Full)	43.4(17.3)	32.5	38.8	62.7	68.0	26.4(19.3)	29.3(17.8)	27.3	
w/o Graph Expansion	40.9(15.1)	30.6	38.7	54.8	70.3	22.9(16.8)	25.3(15.8)	24.6	
w/o Document Routing	41.3(15.4)	32.1	38.2	80.5	57.6	23.6(17.8)	26.7(17.0)	24.5	
w/o Knowledge Retrieval	37.8(13.8)	29.2	37.2	75.0	54.8	21.8(15.5)	24.2(14.3)	23.9	

Table 9: Statistics of retrieved chunk scaling on ASQA.

Methods	ASQA				
	EM(HIT)	D-F1	ROUGE	MAUVE	LEN
chunk1	39.7(14.9)	31.0	38.1	45.6	71.0
chunk2	41.3(15.7)	30.8	38.6	57.7	69.7
chunk3	41.9(15.3)	31.9	38.7	53.5	69.0
chunk4	42.7(16.8)	31.6	38.6	55.9	69.5
chunk5	43.4(17.3)	32.5	38.8	62.7	68.0

Table 10: Statistics of expansion hop scaling on ASQA.

Methods	ASQA				
	EM(HIT)	D-F1	ROUGE	MAUVE	LEN
hop0 (w/o expansion)	40.9(15.1)	30.6	38.7	54.8	70.3
hop1	42.1(15.6)	32.0	39.0	56.9	69.2
hop2	42.5(15.6)	31.7	39.1	57.7	69.1
hop3	42.8(16.9)	32.0	38.6	61.2	69.1
hop4	43.0(16.0)	32.2	38.6	60.2	69.1
hop5	43.4(17.3)	32.5	38.8	62.7	68.0

w/o Document Routing: Building upon graph expansion, we construct a document structure tree for each expanded document, comprising all structure and content nodes, and feed the full tree directly into the knowledge retrieval stage without any structural pruning.

w/o Knowledge Retrieval: After obtaining documents and their relevant structure nodes through document routing, we sequentially concatenate the constructed document structure subtrees until the

context window is filled and then proceed to answer generation, without fine-grained content node selection.

B.3 Scaling Analysis

Tables 9 and 10 provide the full statistics for the chunk scaling and graph expansion hop scaling experiments on ASQA, corresponding to the analysis in Section 4.5.

Table 11: Intermediate module analysis on 200 held-out ASQA examples.

Stage	Micro			Macro		
	P	R	F1	P	R	F1
Router (document)	64.50	90.85	75.44	70.19	88.73	76.00
Router (joint heading)	45.29	66.32	53.82	55.22	68.38	57.81
Router (conditional heading)	65.47	71.41	68.31	75.34	75.93	73.34
Retriever (paragraph)	79.31	81.47	80.37	82.29	84.23	81.11

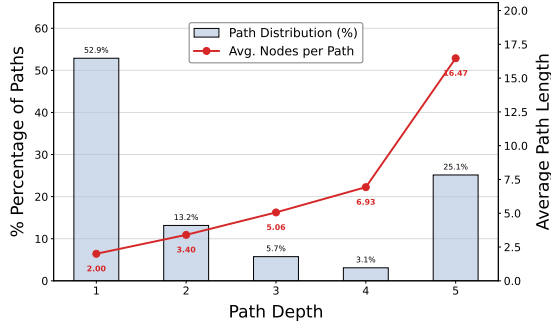


Figure 6: Path analysis of graph expansion on ASQA.

B.4 Intermediate Module Analysis

We evaluate whether the trained router and retriever align with the teacher annotations on 200 randomly sampled ASQA test examples. For each example, we annotate up to five expanded paths for routing and up to eight document subtrees for retrieval, yielding 1,000 routing instances and 1,111 retrieval instances. Reference labels are produced by DeepSeek-V3 using the same teacher prompts as training-data construction (Prompts D.1–D.2). Routing is evaluated at the document level, joint heading level, and conditional heading level, while retrieval is evaluated at the paragraph level. The joint heading setting requires both the document title and heading ID to match, while the conditional heading metric evaluates heading selection only within the intersection of predicted and reference documents. We report micro P/R/F1 by aggregating matches over all examples, and macro P/R/F1 by averaging per-example scores. Table 11 shows that the router preserves high document-level recall, and the retriever closely matches teacher-selected evidence paragraphs.

B.5 Path Analysis

We analyze the depth and length of multi-document paths constructed through graph expansion, corresponding to the number of expansion hops and the number of document nodes. As illustrated in Figure 6, the depth distribution exhibits a bimodal pattern: the majority of paths either terminate at Depth-1 (approx. 50%) or extend to the maximum

Table 12: Performance of deep vs. shallow paths on ASQA.

Methods	ASQA				
	EM(HIT)	D-F1	ROUGE	MAUVE	LEN
Naive RAG	36.5(13.5)	26.9	39.2	31.4	61.2
TSA-RAG	43.4(17.3)	32.5	38.8	62.7	68.0
depth $\leq$ 3	41.5(15.7)	31.4	38.5	54.7	69.2
depth $>$ 3	42.9(16.4)	32.0	38.5	51.3	70.0

Table 13: 2×2 ablation of structural representation and intermediate-module supervision.

Structure	Modules	ASQA EM	ASQA D-F1	Hotpot F1
Naive RAG	–	36.5	26.9	41.7
Flat	zero-shot	38.7	30.8	40.9
Flat	supervised	40.8	31.9	42.7
TDG	zero-shot	41.5	31.5	45.1
TDG	supervised	43.4	32.5	49.8

depth (approx. 25%). The average path length increases nearly linearly for depths 1–4, but rises sharply at depth 5. These patterns may be due to the selective nature of our hybrid scoring mechanism: paths originating from irrelevant first-hop nodes are rapidly truncated due to low relevance scores, while paths leading to high-quality documents are extended deeper.

To verify this, we partition paths into a shallow group (depth ≤ 3) and a deep group (depth > 3), processing each group independently through the router and retriever. As shown in Table 12, the deep group achieves correctness close to that of the full path set and substantially outperforms the shallow group, confirming the high value of deeper paths and validating our expansion algorithm’s effectiveness in exploiting long-range dependencies.

B.6 Disentangling Structural Representation and Intermediate-Module Supervision

Table 13 reports a 2×2 ablation crossing two structural representations (Flat and TDG) with zero-shot and supervised intermediate modules. Flat skips graph expansion and represents each seed document as a title root with all paragraphs as direct children. Zero-shot variants use the base Llama-3.1-8B-Instruct without loading the trained LoRA adapters. Supervised variants use the trained modules with teacher-generated labels. Readers are fixed within each dataset.

Holding the modules fixed, TDG improves over Flat by 2.8/2.6 ASQA EM and 4.2/7.1 HotpotQA F1 under zero-shot/supervised modules. Holding the structure fixed, supervision adds 2.1/1.9 ASQA EM and 1.8/4.7 HotpotQA F1 for Flat/TDG. Thus,

Table 14: Supplementary comparison with graph-based RAG systems.

Method	Hotpot F1	2Wiki F1	Avg. F1
Naive RAG	41.7	32.3	37.0
LongRefiner	45.8	35.0	40.4
RDR ²	44.8	36.7	40.8
GraphRAG	43.0	36.2	39.6
HippoRAG	44.8	37.4	41.1
TSA-RAG	49.8	37.2	43.5

both structural representation and intermediate-module supervision contribute to performance.

B.7 Comparison with Graph-Based RAG Systems

We additionally compare with GraphRAG and HippoRAG (Gutierrez et al., 2024) on HotpotQA and 2Wiki. For computational tractability, these two baselines are indexed separately for each dataset over a reduced passage pool formed by deduplicating the Contriever top-50 passages for all test questions and adding all benchmark-provided supporting and distractor contexts. The resulting pools contain 59,074 and 50,489 passages for HotpotQA and 2Wiki, respectively. We use Qwen3-30B-A3B for graph construction and Llama-3.1-8B-Instruct as the reader.

As shown in Table 14, TSA-RAG achieves the highest average F1 (43.5). It exceeds GraphRAG and HippoRAG by 6.8 and 5.0 F1 on HotpotQA and remains competitive with HippoRAG on 2Wiki (37.2 vs. 37.4). Overall, these results demonstrate TSA-RAG’s competitiveness with graph-oriented retrieval systems.

B.8 Robustness without Native Structural Signals

We evaluate Generic-TDG, which replaces native Wikipedia structures with generic semantic cues. Its global layer connects each document to its top-5 semantic nearest neighbors. Locally, raw text is split at low adjacent-sentence similarity, contiguous sentences are grouped into pseudo-sections, and each group receives a deterministic keyword-based pseudo-heading. This construction uses neither LLM parsing nor additional supervision. Flat, Generic-TDG, and Full TDG use the same trained router and retriever, and all methods use the same reader.

As shown in Table 15, Generic-TDG improves over Flat by 0.4 ASQA EM and 1.7 HotpotQA F1,

Table 15: Results without native hyperlinks or headings.

Method	Global/local structure	ASQA EM	ASQA D-F1	Hotpot F1
Naive RAG	none / chunks seed docs /	36.5	26.9	41.7
Flat	flat paragraphs semantic-kNN /	40.8	31.9	42.7
Generic-TDG	pseudo-sections hyperlinks /	41.2	32.1	44.4
Full TDG	native headings	43.4	32.5	49.8

while Full TDG remains strongest. This provides evidence that the Expand–Route–Retrieve pipeline can operate on induced structures, although native structural signals are more effective in this setting.

C Algorithm: Path-Consistent Graph Expansion

Algorithm 1 presents the full pseudocode for the Path-Consistent Graph Expansion procedure. Starting from a seed node $v_{\text{root}} \in \mathcal{V}_{\text{seed}}$, the algorithm iteratively evaluates all neighbors of the currently retained nodes using the hybrid expansion score s_{exp} (Equation 3) and selects the top- K candidates for the next hop, yielding an expanded subgraph $\mathcal{S}_{\text{sub}}$. The multiplicative formulation of s_{path} accumulates pairwise coherence along the expansion path, thereby penalizing paths with weak successive semantic alignment and mitigating semantic drift.

D Prompts

We present the complete prompt templates used for data annotation (Prompts D.1–D.2) and inference (Prompts D.3–D.4).

CoT Annotation. DeepSeek-V3 serves as the teacher model and is prompted with Chain-of-Thought (CoT) instructions to generate structural labels. The routing annotation prompt (Prompt D.1) and retrieval annotation prompt (Prompt D.2) are shown below.

Inference Prompts. The following prompts are used during inference by the trained $\mathcal{M}_{\text{route}}$ (Prompt D.3) and $\mathcal{M}_{\text{retrieval}}$ (Prompt D.4).

Algorithm 1 Path-Consistent Graph Expansion

Require: Global topology $\mathcal{G}$, query q , root v_{root} , balance α , beam width K , max hops D_{max}

Ensure: Expanded subgraph $\mathcal{S}_{\text{sub}}$

```

1:  $\mathcal{S}_{\text{sub}} \leftarrow \{v_{\text{root}}\}$   $\triangleright$  Initialize subgraph
2:  $\mathcal{Q} \leftarrow [(v_{\text{root}}, [v_{\text{root}}])]$   $\triangleright$  Queue of retained nodes and paths
3: for  $\text{hop} = 1$  to  $D_{\text{max}}$  do
4:    $\mathcal{C} \leftarrow \emptyset$   $\triangleright$  Candidate buffer
5:   for each  $(v_{\text{curr}}, p_{\text{curr}}) \in \mathcal{Q}$  do
6:     for each neighbor  $v_{\text{next}} \in \mathcal{N}(v_{\text{curr}})$  do
7:        $p_{\text{new}} \leftarrow p_{\text{curr}} \oplus v_{\text{next}}$ 
8:        $s_{\text{global}} \leftarrow \text{sim}(\mathbf{e}_{v_{\text{next}}}, \mathbf{e}_q)$ 
9:        $s_{\text{path}} \leftarrow \prod_{k=1}^{|p_{\text{new}}|-1} \text{sim}(\mathbf{e}_{p_{\text{new}}[k]}, \mathbf{e}_{p_{\text{new}}[k-1]})$ 
10:       $s_{\text{exp}} \leftarrow \alpha \cdot s_{\text{global}} + (1 - \alpha) \cdot s_{\text{path}}$ 
11:       $\mathcal{C} \leftarrow \mathcal{C} \cup \{(s_{\text{exp}}, v_{\text{next}}, p_{\text{new}})\}$ 
12:     end for
13:   end for
14:   if  $\mathcal{C} = \emptyset$  then break
15:   end if
16:   Sort  $\mathcal{C}$  by  $s_{\text{exp}}$  descending
17:    $\mathcal{Q} \leftarrow \text{Top-}K \text{ candidates from } \mathcal{C}$ 
18:    $\mathcal{S}_{\text{sub}} \leftarrow \mathcal{S}_{\text{sub}} \cup \{v \mid (\cdot, v, \cdot) \in \mathcal{Q}\}$ 
19: end for
20: return  $\mathcal{S}_{\text{sub}}$ 

```

Prompt D.3: Inference Instruction for Document Routing

Given a multi-document path composed of multiple structured documents, each consisting of a title, an introduction, and hierarchical headings, identify all headings that may contain answers to the specified question. List relevant headings with the tag [title] and [heading]. If none are relevant, reply exactly: "No relevant headings".

```

## Multi-Document Path
{multi_document_path}

## Question
{question}

## Response

```

Prompt D.4: Inference Instruction for Knowledge Retrieval

Given a document structure tree comprising a title, hierarchical headings and subordinate paragraphs, identify all paragraphs that answer the specified question. List each relevant paragraph with the tag [paragraph]. If none are relevant, reply exactly: "No relevant paragraphs".

```

## Document Structure Tree
{document_structure_tree}

## Question
{question}

## Response

```

E Case Study

We present detailed end-to-end cases to illustrate TSA-RAG’s retrieval process.

E.1 End-to-End Answer Comparison

Table 16 shows a full end-to-end comparison between No Retrieval, Naive RAG, and TSA-RAG on the question “*When did New Zealand win the America’s Cup?*”, which requires synthesizing evidence from multiple documents.

The results illustrate a clear progression: the no-retrieval Llama-2-13b-chat model hallucinates an entirely false answer; Naive RAG correctly identifies the 1995 win but misses the 2000 and 2017 victories due to single-document retrieval; TSA-RAG correctly identifies all three wins by performing navigation over the TDG.

E.2 Intermediate Steps

Cases E.1 and E.2 illustrate the input/output of the two navigation models for the same question. $\mathcal{M}_{\text{route}}$ (Case E.1) takes the multi-document path as input and identifies relevant documents together with internal headings. $\mathcal{M}_{\text{retrieval}}$ (Case E.2) then operates on the pruned subtree and extracts the specific evidence paragraphs.

Table 16: End-to-end comparison between No Retrieval, Naive RAG, and TSA-RAG.

User question	When did New Zealand win the America’s Cup?
Short answers	[[“29th America’s Cup”, “6–13 May 1995”, “1995”], [“20 February – 2 March 2000”, “30th America’s Cup”, “2000”], [“35th staging of the America’s Cup yacht race”, “17–26 June 2017”, “2017 America’s Cup”, “2017”]]
Gold long answer	The Royal New Zealand Yacht Squadron yacht club has won the America’s Cup a number of times. The club first won it at the 29th America’s Cup in 1995 , then won its first defense at the 30th America’s Cup in 2000 . The club later won its second defense at the 2017 America’s Cup , the 35th staging of the race.
No Retrieval	Sure, I’d be happy to help! New Zealand has never won the America’s Cup. The America’s Cup is a sailing competition that has been held since 1851 ...
Naive RAG	New Zealand won the America’s Cup once, in 1995 . Team New Zealand, skippered by Russell Coutts, won the challenger series and defeated Dennis Connor’s Stars & Stripes team 5-0 to win the cup. This was the first time New Zealand had won the America’s Cup.
TSA-RAG	New Zealand won the America’s Cup on three occasions: 1. In 1995 , Team New Zealand defeated Team Dennis Conner 5-0 in San Diego. 2. In 2000 , Team New Zealand successfully defended by defeating Prada Challenge 5-0 in Auckland. Additionally, in 2017 , Team New Zealand won the America’s Cup for the third time by defeating Oracle Team USA 7-1 in Bermuda.

Case E.1: Document Routing Input/Output

```
## Multi-Document Path
[Root Doc]
0: =Team New Zealand=
1: Team New Zealand or 'TNZ' is a sailing team...
2: Team New Zealand became a household name...
3: ==History==
5: ===Founding===
7: ===1995: 'Black Magic'===
10: ===2000: "Still New Zealand's Cup"===
...
45: ===35th America's Cup===
50: ==Yachts used by Team New Zealand==
[Doc1]
0: =2013 America's Cup=
...
[Doc2]
0: =America's Cup World Series=
...
[Doc3]
0: =2017 America's Cup=
1: The '2017 America's Cup' was the 35th...
2: It was Oracle's second defence of the...
3: ==Venue==
...

## Question
When did new zealand win the americas cup?

## Response
[title]0: =Team New Zealand=
[heading]7: ===1995: 'Black Magic'===
[heading]10: ===2000: "Still New Zealand's Cup"===
[heading]45: ===35th America's Cup===

[title]0: =2017 America's Cup=
[heading]0: =2017 America's Cup=
```

Case E.2: Knowledge Retrieval Input/Output

```
## Document Structure Tree
0: =Team New Zealand=
1: Team New Zealand or 'TNZ' is a sailing team...
2: Team New Zealand became a household name...
wins in the America's Cup in 1995 and 2000...
In 2017, skippered by Glenn Ashby, they went on...
3: ==History==
7: ===1995: 'Black Magic'===
8: In 1995, TNZ beat Team Dennis Conner...
10: ===2000: "Still New Zealand's Cup"===
11: TNZ ('NZL 60') beat Prada Challenge 5-0
in the 2000 match held on...
45: ===35th America's Cup===
49: ...Team New Zealand beat Oracle Team USA...
and thereby won the Cup.

## Question
When did new zealand win the americas cup?

## Response
[paragraph]2: Team New Zealand...
[paragraph]8: In 1995, TNZ...
[paragraph]11: TNZ ('NZL 60')...
[paragraph]49: Having won the...
```

The router correctly identifies the three historically relevant headings (1995, 2000, and 2017) across documents, and the retriever extracts the four specific paragraphs that directly contain the answer evidence from the document “Team New Zealand”.

Prompt D.1: Teacher CoT Annotation for Routing Labels

You are an expert in information retrieval and reading comprehension. Given a tree-structured document collection generated through knowledge graph propagation, your task is to systematically identify all headings that may lead to answers for the specified question. The documents are organized in BFS order, each presented in a structured format comprising title, introduction, and hierarchical headings.

Extraction Guidelines:

1. Document Selection:

- * For each document, judge whether it is relevant to the question. If it is clearly about a different topic, just skip this document.

- * Consider a document relevant if:

- Its title or introduction contains key entities/concepts from the question
- It is connected to other relevant documents in the tree

2. Heading Evaluation:

- * For each relevant document, examine the internal headings thoroughly and extract all relevant headings.

- * Mark a heading as relevant if:

- It directly matches question keywords/phrases
- It contains synonyms or standard terminology related to the question
- Its subordinate content (when expanded) could contain answers

- * For hierarchical headings, extract the most specific relevant subheadings whenever possible. If all subheadings under a parent heading are relevant, return the parent; otherwise, return only the relevant subheadings. Do not return both the parent and its subheadings.

- * If the introduction directly contains answers to the question, return the title heading.

3. Output Format:

- * Strictly use the following JSON format:

```
[ {"doc_title": str, "heading_id": integer, "explanation": str} ]
```

- * If the collection contains no headings relevant to the question, return "No relevant headings" directly without any further explanation.

- * You should not return any other text or explanation outside the JSON format.

Document Collection

```
{document_collection}
```

Question

```
{question}
```

Prompt D.2: Teacher CoT Annotation for Retrieval Labels

You are an expert in information retrieval and reading comprehension. Given a document structure tree, your task is to identify all paragraphs that contain answers to a specified question. The tree is presented in a structured format comprising a title, hierarchical headings and subordinate paragraphs.

Extraction Guidelines:

1. Paragraph Identification:

- * Include a textual paragraph if it directly answers the question, whether fully or partially.
- * If multiple paragraphs are collectively necessary to form a complete answer, include all relevant ones.
- * Ensure extracted paragraphs are precise (minimizing irrelevant text) while remaining comprehensive (including all key information).

2. Output Format:

- * Strictly use the following JSON format:
[{"content_id": integer, "explanation": str}]
- * If no paragraphs contain answers, return "No relevant paragraphs" directly without any further explanation.
- * You should not return any other text or explanation outside the JSON format.

Document Structure Tree

{document_tree}

Question

{question}