# STAGE-RAG: Continuous Entity State Evolution over Dual-Space Subgraphs for Multi-Hop Reasoning

**Anonymous ACL submission**


## Abstract

Maintaining logical coherence in multi-hop Retrieval-Augmented Generation (RAG) remains a challenging problem, as systems must integrate fragmented evidence without suffering reasoning drift. Iterative frameworks enhance information coupling through query refinement, yet often depend on lossy compressed representations that obscure subtle semantics. Conversely, global graph-based methods ensure structural connectivity but carry topological noise and high computational costs. To overcome these issues, we propose **STAGE-RAG**, a **STA**te-aware **G**raph **E**volution framework that conceptualizes multi-hop reasoning as continuous Bayesian entity-state evolution on a bi-relational chunk subgraph. By tracking entity states composed of identities and dynamic belief scores, our method implicitly guides reasoning via state-weighted propagation. A dual-path retrieval strategy constructs local subgraphs to prune irrelevant nodes while retaining fine-grained logical dependencies. Evaluations on HotpotQA, MuSiQue, 2WikiMultiHopQA and MoreHopQA indicate that STAGE-RAG consistently surpasses state-of-the-art baselines in both accuracy and efficiency.


## 1 Introduction

Large Language Models (LLMs) are advancing unprecedentedly (Luo et al., 2025b; Qin et al., 2026), demonstrating extraordinary capabilities in complex tasks, including long-context summarization (Zhang et al., 2024), autonomous code synthesis (Grishina et al., 2025) and sophisticated question answering (Ma et al., 2025a). However, constrained by static parametric memory, LLMs struggle with outdated knowledge (Feng et al., 2024) and insufficient domain expertise (Ling et al., 2025), resulting in incorrect outputs (Huang et al., 2025).

In response to these limitations, Retrieval-Augmented Generation (RAG) enhances LLMs by integrating relevant information from external knowledge sources (Lewis et al., 2020). While effective for simple queries, standard RAG struggles with multi-hop question answering (MHQA), as shown in Figure 1(a), which requires synthesizing evidence across documents (Trivedi et al., 2023). This complex reasoning process faces two primary bottlenecks: first, irrelevant retrieval noise frequently obscures critical bridge entities due to the "lost in the middle" phenomenon (Liu et al., 2024). Moreover, without modeling evidence evolution, the reasoning trajectory gradually deviates from the original query as hops accumulate, leading to semantic drift and coherence breakdown (Yang et al., 2024).

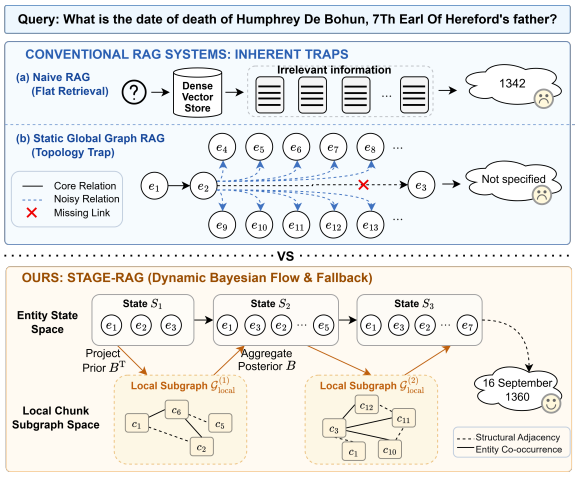


Figure 1: Comparison of conventional RAG systems (top) and STAGE-RAG (bottom). Conventional methods suffer from document noise and topological traps that obscure critical edges. In contrast, STAGE-RAG maintains robust reasoning through continuous Bayesian entity state updates.

Existing methods address these problems via iterative reasoning or knowledge graph (KG) construction. Iterative retrieval mimics the Chain-of-Thought (CoT) process (Wei et al., 2022; Kojima et al., 2022) by decomposing queries into sub-questions (Trivedi et al., 2023; Asai et al., 2024)

and solving them step-by-step to incrementally gather supporting evidence. However, unlike internal reasoning in CoT, iterative retrieval relies on external sources at each step. This makes it prone to reasoning drift, as cumulative errors divert the focus from the original query (Zhang et al., 2025a). Graph-based RAG offers a structured alternative but faces a trilemma of cost, noise, and complexity (Zhou et al., 2025), as shown in Figure 1(b). Conventional methods build global KGs through triplet extraction (Peng et al., 2025), which incurs token costs and disrupts source text integrity (Zhang et al., 2025b). To improve efficiency, (Zhu et al., 2025) uses raw sentences as nodes to construct global graphs, though this often introduces redundant connections that impair retrieval precision (Zhang et al., 2025d). While hybrid architectures integrating both triplets and chunks improve coverage (Gutiérrez et al., 2025; Ghassel et al., 2025), they suffer from prohibitive indexing latency and excessive graph complexity (Zhao et al., 2025).

To address these challenges, we propose **STAGE-RAG** (**STA**te-aware **G**raph **E**volution for RAG), a novel framework that synergizes continuous entity-state evolution with dynamic local subgraphs, as illustrated in Figure 1. Unlike paradigms relying on static retrieval or discrete iterations, STAGE-RAG formalizes multi-hop reasoning through three integrated components. First, the Entity-Based State Evolution mechanism establishes a consistent memory for complex inference trajectories. Through continuous Bayesian updates, it fuses prior entity weights with the observation likelihood of retrieved chunks to compute refined posterior probabilities. This synthesis of past states and current evidence highlights critical nodes, dampening cascading errors and mitigating the "lost-in-the-middle" challenge. Second, a State-Aware Dual-Path Retrieval strategy integrates explicit semantic anchors with implicit entity states. The explicit path anchors the reasoning chain in factual origins. Simultaneously, the implicit path aggregates belief scores to uncover latent evidence essential to reasoning evolution, even if unmentioned in the sub-question. Finally, these paths converge into a dynamically instantiated bi-relational local subgraph. This maps retrieved evidence into a structured topology interconnected by structural adjacency and entity co-occurrence. By tracking this evolving belief distribution, STAGE-RAG seamlessly bridges the semantic-structural gap, ensuring logical coherence across disjoint document spaces.

In summary, the main contributions of our work are as follows:

- We propose **STAGE-RAG**, which reframes multi-hop reasoning as a continuous evolution of entity states, moving beyond conventional retrieve-and-concatenate approaches. By explicitly tracing belief trajectories, this work provides a structured and interpretable theoretical foundation for mitigating reasoning drift.
- We introduce a **State-Aware Dual-Path Retrieval** strategy, which dynamically constructs multi-granular local subgraphs to decompose the global reasoning task into structured local decisions. This design ensures the reasoning focus consistently evolves around core evidence, significantly enhancing the accuracy and robustness of path selection.
- Extensive experiments on MuSiQue, HotpotQA, 2WikiMultiHopQA and MoreHopQA datasets demonstrate that STAGE-RAG achieves an optimal accuracy-efficiency balance across various LLM backbones.

## 2 Related Work

### 2.1 RAG for Multi-Hop Reasoning

Multi-Hop Question Answering (MHQA) requires synthesizing fragmented evidence, serving as a critical yardstick for RAG systems. To optimize evidence quality, RankRAG (Yu et al., 2024) unifies context ranking and answer generation within a single LLM, while SETR (Lee et al., 2025a) fine-tunes LLMs for set-wise passage selection guided by Information Requirement Identification. However, these methods treat retrieval as static, neglecting the evolving semantic focus required by multi-step reasoning. To introduce dynamic awareness, Q-Dream (Ye et al., 2025) models interdependence between sub-questions during decomposition, and ReSCORE (Lee et al., 2025b) aligns the retriever with LLM knowledge through iterative refinement. S2G-RAG (Li et al., 2026) further introduces explicit evidence sufficiency judging to stabilize iterative retrieval trajectories. ReAgent (Xinjie et al., 2025) employs a hybrid backtracking mechanism to rectify incorrect reasoning chains. Nevertheless, these frameworks remain constrained by discrete reasoning. Relying on compressed representations or reactive correction, they fail to sustain a continuous, high-fidelity memory, leading to cascading failures as reasoning state drifts across hops.

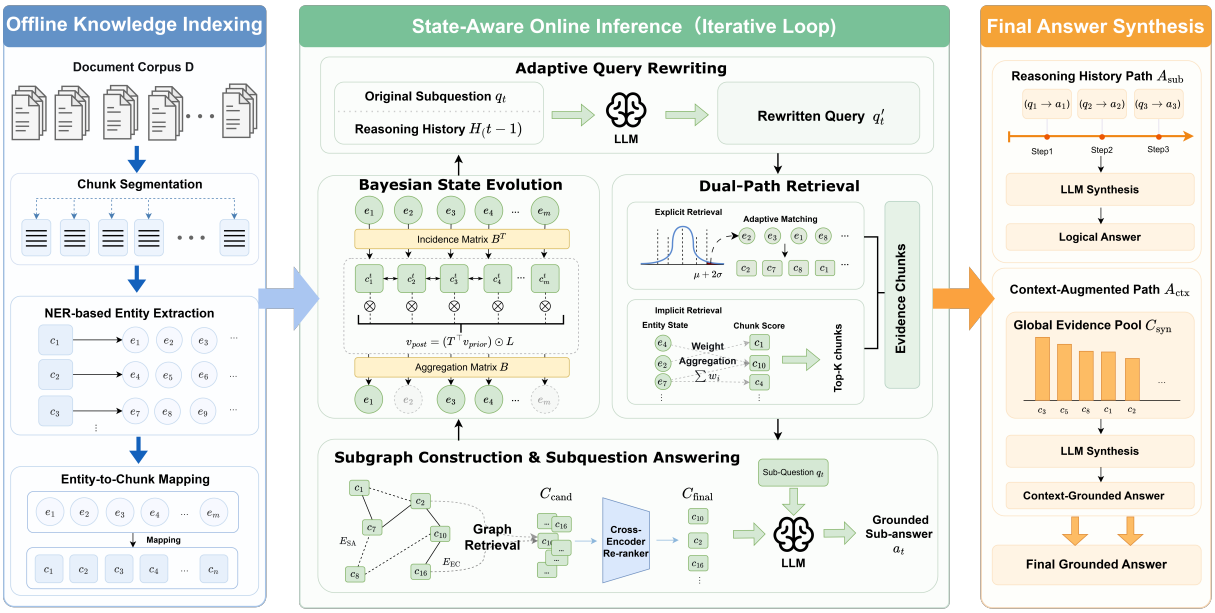


Figure 2: Architecture of STAGE-RAG. The framework establishes an offline hierarchical knowledge index to map global semantic entities to constituent text chunks. This foundation drives the online state-aware iterative inference loop, which leverages dynamic entity beliefs and dual-path retrieval to bridge abstract reasoning with concrete textual evidence.

### 2.2 Graph-based RAG

To enhance retrieval with structural semantics, recent graph-based methods model complex associations (Edge et al., 2024). Systems like HippoRAG (Jimenez Gutierrez et al., 2024), ToG-2 (Ma et al., 2025b) and HyperGraphRAG (Luo et al., 2025a) utilize entity nodes and relational edges for associative retrieval across disjoint documents. In contrast, frameworks such as ChainRAG (Zhu et al., 2025) and SGMem (Wu et al., 2025) organize chunks or sentences as nodes to preserve broader context and mitigate information loss from triplet extraction. To leverage both granularities, LightRAG (Guo et al., 2025) and HippoRAG2 (Gutiérrez et al., 2025) implement hierarchical structures indexing coarse document communities and fine-grained entity relationships for multi-level retrieval. Recent methods such as LogicRAG (Chen et al., 2026) further introduce adaptive graph pruning and inference-time reasoning structures to improve retrieval efficiency. While these graph structures enhance connectivity, they rely on static global indexing over the corpus, introducing topological noise and computational overhead as irrelevant high-degree nodes induce reasoning drift during traversal (Zhao et al., 2025). Our STAGE-RAG instead constructs dynamic local subgraphs, pruning irrelevant pathways based on evolving states to mitigate topological noise and reasoning drift in large-scale graphs.

## 3 Methodology

### 3.1 State Formalization and Overview

Instead of treating multi-hop reasoning as a series of independent retrieval steps, we formulate it as a sequential Bayesian inference process over an evolving entity belief distribution. During the multi-hop reasoning process, the state at hop $t$ is anchored to a global entity repository $\mathcal{E}$:

$$S_t = \{(e_i, w_i^{(t-1)}, \phi_i) \mid e_i \in \mathcal{E}\} \tag{1}$$

where $e_i$ denotes an entity, $w_i^{(t-1)} \in [0, 1]$ serves as a belief score, reflecting the posterior probability that $e_i$ is relevant given evidence up to hop $t-1$. $\phi_i$ tracks provenance (e.g., query or diffusion). This evolution from $S_{t-1}$ to $S_t$ provides a mathematical foundation for mitigating semantic drift.

The framework leverages an offline hierarchical knowledge index in Section 3.2 to execute an iterative online inference loop in Section 3.3. At hop $t$, ambiguities in $q_t$ are first resolved via adaptive rewriting. Next, a state-aware dual-path retrieval strategy instantiates a local subgraph using explicit anchors and implicit belief-weighted evidence. The state is then updated via continuous Bayesian flow, projecting beliefs onto the subgraph to calculate posterior scores $w_i^{(t)}$. This reinforces high-confidence paths while pruning noise, culminating in a synthesized final answer in Section 3.4.

### 3.2 Knowledge Indexing and Entity Mapping

We design a hierarchical knowledge indexing scheme to construct coherent logical paths across heterogeneous documents. Given a corpus $D$ consisting of a set of documents, we partition it into text chunks $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$ and apply a pretrained Named Entity Recognition (NER) model to extract a global entity set $\mathcal{E} = \{e_1, e_2, \ldots, e_m\}$. These entities serve as semantic anchors for cross-document logical associations. To navigate from abstract entities to concrete evidence, we establish an inverted index $\mathcal{M} : \mathcal{E} \to 2^{\mathcal{C}}$, defined as:

$$\mathcal{M}(e) = \{c_i \in \mathcal{C} \mid e \in \mathcal{V}(c_i)\}, \tag{2}$$

where $\mathcal{V}(c_i) \subseteq \mathcal{E}$ denotes the entities identified within the parent document of chunk $c_i$. Finally, we utilize a pretrained encoder $f_\phi$ to project both individual entities $e_i \in \mathcal{E}$ and text chunks $c_j \in \mathcal{C}$ into a unified high-dimensional vector space $\mathbb{R}^d$:

$$\mathbf{h}_{e_i} = f_\phi(e_i), \quad \mathbf{h}_{c_j} = f_\phi(c_j), \tag{3}$$

where $\mathbf{h}_{e_i}$ and $\mathbf{h}_{c_j}$ are dense embeddings for subsequent semantic matching.

### 3.3 State-Aware Inference Loop

The online inference of STAGE-RAG is an iterative process. At each hop $t$, the system navigates the global entity space $\mathcal{E}$ to aggregate evidence through following synchronized stages: rewriting, retrieval, and state evolution.

#### 3.3.1 Adaptive Question Rewriting

To mitigate the compositional bottleneck of dense retrievers, we first decouple evidence acquisition from symbolic execution. Given a complex multi-hop query $Q$, an LLM decomposes it into a logically dependent sequence of retrieval-centric sub-questions $Q = \{q_1, q_2, \ldots, q_T\}$. Purely logical operations are deferred to the final synthesis, ensuring each $q_t$ acts solely as a factual probe.

During iterative retrieval, sub-questions $q_t$ ($t > 1$) often encounter semantic gaps due to unresolved pronouns (e.g., "his mother", "that film") or implicit dependencies on prior outcomes. To guarantee self-contained representations, we adopt a history-aware adaptive rewriting mechanism that transforms $q_t$ into a standalone query $q'_t$:

$$q'_t = \begin{cases} \mathcal{R}(q_t, \mathcal{H}_{t-1}), & \text{if } \mathcal{G}(q_t, \mathcal{H}_{t-1}) = \text{True}, \\ q_t, & \text{otherwise.} \end{cases} \tag{4}$$

where $\mathcal{H}_{t-1} = \{(q_1, a_1), \ldots, (q'_{t-1}, a_{t-1})\}$ denotes the reasoning history up to hop $t-1$. An LLM-driven gating operator $\mathcal{G}$ detects unresolved anaphoric references in $q_t$. If triggered, the rewriting operator $\mathcal{R}$ grounds implicit pronouns using preceding answers. This dynamic resolution ensures every projected query is structurally complete, preventing semantic drift at each hop.

#### 3.3.2 State-Aware Dual-Path Retrieval

For each optimized sub-question $q'_t$, we implement a **state-aware dual-path retrieval strategy**. This strategy integrates deterministic semantic anchors with probabilistic entity belief weights to instantiate a task-sensitive local subgraph.

**Explicit Anchor Path.** To anchor the reasoning process to factual origins, we initially isolate the query focus. An LLM extracts the subject entity $\hat{e}_t$ directly from $q'_t$. Using the encoder $f_\phi$, we generate its dense representation $\mathbf{h}_{\hat{e}_t} = f_\phi(\hat{e}_t)$ to identify a set of target entities $\mathcal{E}_{q'_t} \subset \mathcal{E}$ through a statistically adaptive strategy. Instead of rigid thresholds, $\mathcal{E}_{q'_t}$ strictly prioritizes exact string alignment by including all entities satisfying $\mathrm{EM}(e, \hat{e}_t)$. If none exist, it captures statistical outliers satisfying $s(\mathbf{h}_e, \mathbf{h}_{\hat{e}_t}) \geq \mu + 2\sigma$, where $s$ denotes the cosine similarity, and $\mu$, $\sigma$ are the mean and standard deviation of similarities over $\mathcal{E}$.

Leveraging the pre-constructed inverted index mapping $\mathcal{M}$, we subsequently retrieve the explicit evidence set:

$$\mathcal{C}_{\text{exp}} = \bigcup_{e \in \mathcal{E}_{q'_t}} \mathcal{M}(e). \tag{5}$$

**Implicit State-Aware Path.** To complement explicit retrieval, we leverage the current reasoning state $S_{t-1}$ to identify latent evidence critical to logical evolution, even if absent from $q'_t$. For each candidate chunk $c_j \in \mathcal{C}$, we calculate a state-relevance score by aggregating the belief weights $w_i^{(t-1)}$ of its active constituent entities:

$$\text{Score}(c_j) = \sum_{(e_i, w_i^{(t-1)}, \phi_i) \in S_{t-1},\, e_i \in \mathcal{V}(c_j)} w_i^{(t-1)}, \tag{6}$$

The implicit retrieval set $\mathcal{C}_{\text{imp}}$ is then constructed by selecting a subset of exactly $k$ chunks that maximizes the cumulative relevance:

$$\mathcal{C}_{\text{imp}} = \arg\max \mathcal{C}' \in \binom{\mathcal{C}}{k} \sum_{c_j \in \mathcal{C}'} \text{Score}(c_j). \tag{7}$$

#### 3.3.3 Local Subgraph Construction

Upon acquiring candidate chunks for $q'_t$, we instantiate a task-specific local subgraph $\mathcal{G}^{(t)}_{\text{local}} = (V_t, E_t)$ to model the inter-document dependencies.

The vertex set $V_t$ comprises the retrieved chunks $\mathcal{C}_{\text{exp}} \cup \mathcal{C}_{\text{imp}}$. The edge set models dual-dimensional text relations. Structural adjacency ($E_{\text{SA}}$) preserves narrative continuity by linking chunks within a sliding window $\omega$. Let $\Delta_{ij} = |pos(c_i) - pos(c_j)|$ denote the positional offset. The adjacency weight is assigned inversely proportional to this offset:

$$w_{\text{SA}}(c_i, c_j) = \Delta_{ij}^{-1} \cdot \mathbb{I}(\Delta_{ij} \leq \omega) \tag{8}$$

where $\mathbb{I}(\cdot)$ is the indicator function. Entity co-occurrence ($E_{\text{EC}}$) facilitates cross-document transitions. To prevent ubiquitous entities from causing edge proliferation, we apply an Information Content filter $\text{IC}(e) = \ln(|\mathcal{V}_t|/\text{df}(e))$, where $\text{df}(e)$ indicates the node frequency. Let $\mathcal{E}^*_{ij}$ denote shared entities between chunks $c_i$ and $c_j$with frequencies above $\mu_{\text{ic}} - \sigma_{\text{ic}}$, estimated from the global information distribution over $V_t$. Scaled by factor $\gamma$, the dynamic edge weight aggregates the normalized information content of these filtered entities:

$$w_{\text{EC}}(c_i, c_j) = \gamma \sum_{e \in \mathcal{E}^*_{ij}} \frac{\text{IC}(e)}{\max_{e'} \text{IC}(e')} \tag{9}$$

To identify high-fidelity evidence, we recall a seed set $\mathcal{S}_t$ via dense similarity against $q'_t$, expanding it to its one-hop neighborhood $\mathcal{C}_{\text{cand}} = \mathcal{S}_t \cup \mathcal{N}(\mathcal{S}_t)$ to capture latent dependencies. A Cross-Encoder reranker refines $\mathcal{C}_{\text{cand}}$ into the top-$K$ chunks $\mathcal{C}_{\text{final}}$, which an LLM then synthesizes into a grounded sub-answer $a_t$, updating the reasoning history:

$$\mathcal{H}_t = \mathcal{H}_{t-1} \cup \{(q'_t, a_t)\}. \tag{10}$$

#### 3.3.4 Entity State Evolution

To ensure mathematically principled evidence aggregation under retrieval uncertainty, we formulate multi-hop reasoning as a continuous Bayesian inference process over local subgraphs, driving a probability flow between entity and chunk spaces.

**Prior Diffusion.** At each hop $t$, we form the prior state vector $\mathbf{s}_{\text{prev}} \in \mathbb{R}^M$ by taking the union of historical and newly extracted entities. To project these entity-level beliefs onto local subgraph $\mathcal{G}^{(t)}_{\text{local}}$, which consists of $N$ chunk nodes, we construct an incidence matrix $\mathbf{B} \in \{0,1\}^{M \times N}$ where $\mathbf{B}_{k,j} = 1$ if entity $e_k$ is present in chunk $c_j$. The beliefs are projected onto the chunks and propagated via Markovian diffusion to capture latent dependencies:

$$\mathbf{v}_{\text{prior}} = \mathbf{T}^\top \text{Norm}(\mathbf{B}^\top \mathbf{s}_{\text{prev}}) \tag{11}$$

where $\mathbf{T}$ denotes the transition matrix of $\mathcal{G}^{(t)}_{\text{local}}$ equipped with self-loops to preserve node inertia.

**Bayesian Posterior Update.** Observational evidence from $q'_t$ is quantified via a likelihood vector $\boldsymbol{\mathcal{L}} = \exp(s(\mathbf{C}, \mathbf{q}'_t)/\tau)$, where $\mathbf{C}$ comprises the chunk embeddings and $\tau$ is the temperature.

Following Bayes' theorem, the posterior belief over the chunks is updated as:

$$\mathbf{v}_{\text{post}} = \text{Norm}(\mathbf{v}_{\text{prior}} \odot \boldsymbol{\mathcal{L}}) \tag{12}$$

**State Distillation.** We aggregate the posterior upward to directly compute the updated entity belief scores:

$$\mathbf{s}_{\text{new}} = \text{MaxNorm}(\mathbf{B}\mathbf{v}_{\text{post}}) \tag{13}$$

Since $\mathbf{v}_{\text{post}}$ inherently encapsulates the historical prior, $\mathbf{s}_{\text{new}}$ comprehensively reflects the memory-aware beliefs. To ensure computational efficiency and maintain reasoning focus, we prune the state into $\mathcal{S}_t$ by retaining entities satisfying $w_i^{(t)} \geq \theta$.

### 3.4 Final Answer Synthesis

Upon completing $T$ hops, we construct a global evidence pool $\mathcal{C}_{\text{syn}}$ by deduplicating and re-ranking the aggregated chunks $\bigcup_{t=1}^{T} \mathcal{C}^{(t)}_{\text{final}}$ against $Q$. To capture dual reasoning dimensions, the final synthesis yields two complementary outputs:

**Sub-answer Path:** Focuses on deductive logic by conditioning on the question $Q$ and the reasoning history $\mathcal{H}_T = \{(q'_t, a_t)\}_{t=1}^{T}$:

$$A_{\text{sub}} = \text{LLM}(Q, \mathcal{H}_T) \tag{14}$$

**Context-augmented Path:** Ensures factual grounding by conditioning directly on the sorted evidence pool $\mathcal{C}_{\text{syn}}$:

$$A_{\text{ctx}} = \text{LLM}(Q, \mathcal{C}_{\text{syn}}) \tag{15}$$

## 4 Experiments

### 4.1 Experimental Settings

**Datasets.** We utilize four MHQA datasets to validate our architecture. HotpotQA(Yang et al., 2018) is a multi-hop reasoning dataset requiring integration of information across documents. 2WikiMultiHopQA(Ho et al., 2020) is a Wikipedia-based

benchmark specifically designed to evaluate cross-document evidence synthesis and reasoning chains. MuSiQue(Trivedi et al., 2022) provides a more rigorous test of reasoning depth, typically involving 2-4 hops. MoreHopQA(Schnitzler et al., 2024) is a recent benchmark evaluating long-chain compositional reasoning, requiring extensive multi-hop steps and cross-document evidence aggregation.

**Compared Methods.** We compare our method against several publicly available baselines: Vanilla LLM, which directly generates answers without external retrieval; Naive RAG (Lewis et al., 2020), a standard **chunk-based** retrieval method; ChainRAG (Zhu et al., 2025), which constructs a global **sentence-based** graph for iterative reasoning; LightRAG (Guo et al., 2025) and HyperGraphRAG (Luo et al., 2025a), which utilize **entity-based** graphs for retrieval; HippoRAG2 (Gutiérrez et al., 2025), which builds **hybrid-based** architecture incorporating both text chunks and entity nodes.

**Metrics and Models.** Lexical similarity between the generated and ground-truth answers is evaluated using Exact Match (EM) and F1-score. We conduct experiments using multiple LLMs, including the open-source Qwen3 series (8B, 14B, 30B) (Yang et al., 2025) as well as GPT-4o-mini (Hurst et al., 2024), to evaluate effectiveness and generalizability across different model architectures and scales. All vector retrieval tasks are performed using the Qwen3-Embedding-0.6B model (Zhang et al., 2025c), while the BGE-Reranker (Xiao et al., 2024) is employed as the cross-encoder for evidence refinement.

**Implementation Details.** All experiments are conducted on 8 $\times$ NVIDIA RTX 4090 GPUs. Implemented via PyTorch and HuggingFace, the framework uses vLLM (Kwon et al., 2023) for efficient inference. For fair comparison, all LLMs employ deterministic decoding at temperature 0. For retrieval, the top-$k$ for supplying context is set to 5, the sliding window size $\omega$ is set to 2, the edge scaling factor $\gamma$ is 0.5, and the seed set size is fixed at 7. For the Bayesian posterior update, the temperature hyperparameter $\tau$ is 0.1. For new entities, we set the initial anchor weight to 0.1, while the pruning threshold $\theta$ is established at 0.5 to filter low-confidence reasoning paths. These values reflect stable regions detailed in Appendix D rather than per-dataset tuning. Further details are provided in Appendix A.

### 4.2 Main Results

Table 1 summarizes the comparative performance of **STAGE-RAG** and several state-of-the-art baselines across four multi-hop reasoning benchmarks. Overall, our method consistently achieves the highest scores across all metrics and datasets, demonstrating its competitive capability in handling complex information synthesis.

**Beyond Parametric Knowledge.** Although LLMs may have encountered benchmark corpora during pre-training, their zero-shot performance remains limited in complex multi-hop tasks. For instance, Vanilla Qwen3-30B achieves an average F1 of only 21.45%, while our state-aware retrieval boosts this to 59.94% (+38.49). This indicates that internal parametric knowledge is insufficient for deep evidence synthesis, whereas our architecture provides the necessary factual grounding.

**Overall Performance.** STAGE-RAG shows consistent improvements across all LLM backbones, demonstrating reduced reliance on massive parameter scales. While methods such as ChainRAG rely heavily on larger models to stabilize reasoning, and graph-based approaches like LightRAG are constrained by the extraction quality of smaller LLMs, our framework bypasses these bottlenecks. We attribute this advantage to the state-aware evolution process, where dynamic entity updates provide more stable reasoning context and localized subgraph construction reduces the impact of globally noisy graph structures. As a result, STAGE-RAG maximizes the intrinsic capabilities of smaller models, outperforming HippoRAG2 by 8.88% F1 points on Qwen3-8B, while remaining effective on stronger backbones.

### 4.3 Ablation Study

To evaluate the contribution of each module within STAGE-RAG, we conduct an ablation study across MuSiQue, HotpotQA and 2WikiMultiHopQA based on Qwen3-8B, systematically removing the Adaptive Query Rewriting (w/o AQR), Entity State Tracking (w/o EST), Local SubGraph (w/o LSG), Structural Adjacency Edges (w/o ESA), and Entity Co-occurrence Edges (w/o EEC).

As shown in Figure 3 and 4, the removal of AQR causes the most severe performance degradation, suggesting its strict necessity in preventing semantic drift during complex multi-hop reasoning. Similarly, discarding EST leads to a notable 5.07

| Model | Method | MuSiQue | | HotpotQA | | 2WikiMHQA | | MoreHopQA | | Avg | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| Qwen3-8B | Vanilla LLM | 6.00 | 13.01 | 17.00 | 26.34 | 23.00 | 27.61 | 4.00 | 5.83 | 12.50 | 18.20 |
| | Naive RAG | 16.00 | 24.78 | 37.00 | 50.73 | 40.00 | 48.82 | 4.50 | 6.48 | 24.38 | 32.70 |
| | ChainRAG | 26.00 | 34.79 | 45.50 | 58.22 | 47.00 | 57.63 | 44.00 | 44.25 | 40.63 | 48.72 |
| | LightRAG | 19.00 | 26.22 | 33.50 | 45.42 | 32.50 | 41.71 | 9.00 | 10.31 | 23.50 | 30.92 |
| | HyperGraphRAG | 30.50 | 38.47 | 47.00 | 60.69 | 51.50 | 61.66 | 25.00 | 26.26 | 38.50 | 46.77 |
| | HippoRAG2 | 27.00 | 36.87 | 43.50 | 56.03 | 52.50 | 62.36 | 32.50 | 34.25 | 38.88 | 47.38 |
| | **Ours** | **38.50** | **47.34** | **48.00** | **62.37** | **57.00** | **67.01** | **47.00** | **48.33** | **47.63** | **56.26** |
| Qwen3-14B | Vanilla LLM | 2.00 | 8.77 | 20.00 | 27.23 | 24.50 | 28.85 | 4.00 | 5.86 | 8.67 | 17.68 |
| | Naive RAG | 21.00 | 30.42 | 36.50 | 51.76 | 45.00 | 54.96 | 6.00 | 8.25 | 21.17 | 36.35 |
| | ChainRAG | 27.50 | 39.69 | 44.00 | 58.60 | 51.50 | 61.23 | 32.00 | 33.95 | 34.50 | 48.37 |
| | LightRAG | 15.00 | 25.94 | 28.50 | 44.20 | 22.00 | 39.90 | 4.50 | 5.60 | 16.00 | 28.91 |
| | HyperGraphRAG | 25.50 | 35.53 | 44.00 | 59.04 | 33.00 | 50.89 | 12.00 | 12.54 | 27.17 | 39.50 |
| | HippoRAG2 | 25.00 | 34.72 | 41.50 | 53.80 | 53.00 | 63.38 | **41.00** | **42.73** | 35.83 | 48.66 |
| | **Ours** | **38.50** | **48.39** | **47.50** | **63.36** | **58.50** | **68.07** | **41.00** | 42.71 | **42.33** | **55.63** |
| Qwen3-30B | Vanilla LLM | 6.50 | 14.21 | 23.50 | 32.13 | 27.00 | 32.03 | 5.00 | 7.42 | 11.67 | 21.45 |
| | Naive RAG | 19.50 | 29.58 | 35.50 | 52.25 | 49.50 | 58.66 | 15.00 | 16.67 | 23.33 | 39.29 |
| | ChainRAG | 28.00 | 40.21 | 46.00 | 61.60 | 55.50 | 65.06 | 33.00 | 35.25 | 35.67 | 50.28 |
| | LightRAG | 30.50 | 38.69 | 44.50 | 57.91 | 50.50 | 60.94 | 20.50 | 21.12 | 31.83 | 44.67 |
| | HyperGraphRAG | 38.50 | 47.12 | **50.00** | **62.56** | 56.00 | 66.03 | 25.50 | 26.75 | 38.00 | 50.62 |
| | HippoRAG2 | 33.00 | 42.15 | 44.50 | 56.29 | 57.00 | 65.56 | 44.00 | 45.25 | 40.50 | 52.31 |
| | **Ours** | **39.50** | **47.95** | 47.50 | 62.14 | **62.00** | **71.93** | **56.00** | **57.75** | **47.67** | **59.94** |
| GPT-4o-mini | Vanilla LLM | 12.00 | 18.64 | 29.00 | 38.40 | 25.00 | 30.62 | 17.00 | 17.00 | 19.33 | 26.17 |
| | Naive RAG | 27.00 | 36.65 | 41.00 | 55.39 | 48.50 | 59.05 | 22.50 | 23.00 | 30.17 | 43.52 |
| | ChainRAG | 35.50 | 45.25 | 45.00 | 59.17 | 51.00 | 60.08 | 33.00 | 34.00 | 37.83 | 49.63 |
| | LightRAG | 22.50 | 32.92 | 29.50 | 46.34 | 33.50 | 48.20 | 10.00 | 12.07 | 20.67 | 34.88 |
| | HyperGraphRAG | 32.50 | 40.65 | 47.50 | 59.48 | 49.00 | 61.43 | 28.50 | 28.70 | 36.17 | 47.57 |
| | HippoRAG2 | 34.00 | 41.72 | 43.50 | 56.73 | 55.00 | 64.96 | 45.00 | 46.45 | 40.83 | 52.47 |
| | **Ours** | **42.00** | **51.61** | **49.50** | **63.67** | **57.50** | **67.96** | **47.00** | **47.00** | **46.17** | **57.56** |

Table 1: Performance comparison of retrieval augmented generation methods across backbones. For ChainRAG and STAGE-RAG, we report optimal results between $A_{\text{sub}}$ and $A_{\text{ctx}}$ paths. Deployment boundaries for these dual synthesis trajectories are analyzed in Section B.1. Best results are bolded, second best underlined.

F1 decline, validating that dynamic state updates are crucial for establishing reliable contextual anchors. Among the graph components, completely omitting the local subgraph (w/o LSG) naturally results in the most significant drop. When dissecting the specific edge types, eliminating EEC causes a much larger decline than ESA, which indicates that while positional continuity provides complementary support for certain cases, entity-centric topological bridges serve as the primary driver for successful cross-document transitions.

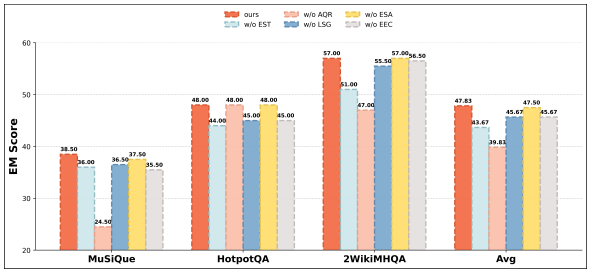


Figure 3: EM (%) ablation results across datasets.

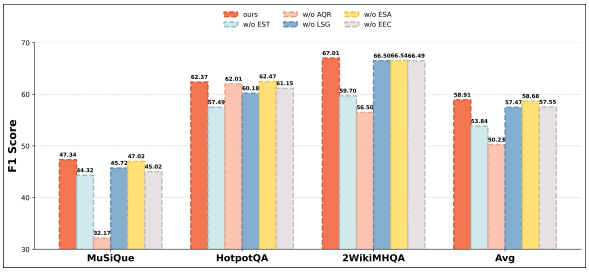


Figure 4: F1 (%) ablation results across datasets.

## 4.4 Component Analysis and Robustness

To validate STAGE-RAG's architectural rationality and resilience, we investigate alternative component designs, shown in Table 2, and conduct stress tests regarding incomplete information extraction based on Qwen3-8B across MuSiQue, HotpotQA, and 2WikiMultiHopQA datasets.

**Bayesian vs. LLM Scoring.** As detailed in Table 2, replacing our Bayesian tracker with LLM entity

| Method | Avg | | |
|---|---|---|---|
| | F1 | Token | LLM Calls |
| **Ours** | **58.91** | **7.7k** | **11.49** |
| LLM Scoring | 56.57 | 9.1k | 12.77 |
| Paragraph Chunking | 57.44 | 8.2k | 11.51 |

Table 2: Average performance of component variants.

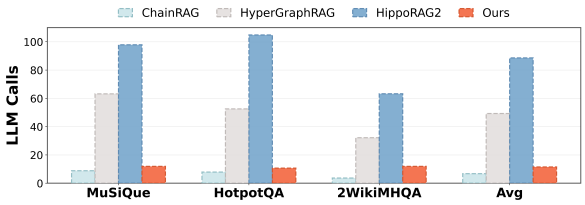


Figure 6: Comparison of Qwen3-8B call times.

| Method | Tokens | Latency (s) | F1 |
|---|---|---|---|
| ChainRAG | **3.1k** | **5.62** | 50.21 |
| HyperGraphRAG | 27.6k | 146.95 | 53.61 |
| HippoRAG2 | 10.1k | 88.63 | 51.75 |
| Ours | 7.7k | 6.60 | **58.91** |

Table 3: Average efficiency comparison across datasets.

scoring over the retrieved chunks degrades average F1 to 56.57%. It also exacerbates overhead, increasing LLM calls by 1.28 and token consumption by 1.4k. This confirms that LLM-based scoring introduces latency and susceptibility to context noise, whereas our Bayesian updating ensures a deterministic, efficient mechanism.

**Impact of Context Granularity.** Segmenting corpus by paragraphs rather than our semantic chunking degrades average F1 and increases token usage, shown in Table 2. Coarse paragraph blocks frequently encapsulate distractor facts that inject irrelevant noise into entity states during dual-space projection, proving bounded semantic chunks are essential for maximizing the signal-to-noise ratio.

**Robustness to Extraction Noise.** We simulate realistic NER recall failures by randomly dropping extracted entities at proportions up to 40%, as illustrated in Figure 5. STAGE-RAG exhibits graceful degradation, maintaining 57.79% F1 even when 20% of entities are deleted. Interestingly, performance peaks at a 10% drop rate, reaching 59.88% F1. This demonstrates a self-regularizing effect where the topological fallback mechanism bypasses missing nodes via structural adjacency edges while filtering peripheral noisy entities to stabilize reasoning pathways.

### 4.5 Efficiency Analysis

We evaluate the computational efficiency of different multi-hop frameworks by analyzing the average LLM calls, total token usage, and inference latency. As shown in Figure 6 and Table 3, STAGE-RAG substantially reduces computational costs compared to heavy graph-enhanced approaches like HyperGraphRAG and HippoRAG2, which suffer from a massive number of LLM queries. While ChainRAG shows slightly lower latency and fewer LLM calls, our approach secures a vastly superior tradeoff. By delegating state evolution to deterministic matrix diffusion rather than redundant model generation, STAGE-RAG introduces a marginal latency overhead of less than 1 second while delivering an +8.70% absolute F1 gain over ChainRAG. This confirms our mechanism successfully achieves an optimal balance between reasoning accuracy and efficiency.

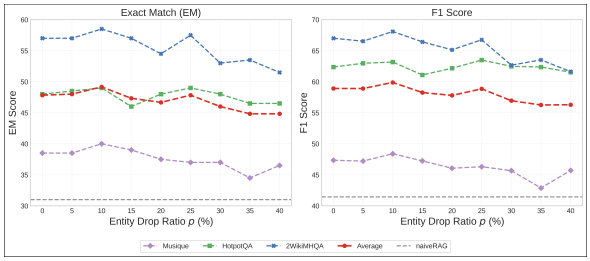


Figure 5: EM (%) and F1 (%) scores under varying entity drop ratios $p \in [0\%, 40\%]$.

## 5 Conclusion

In this work, we introduce STAGE-RAG, a novel state-aware graph-based framework designed to mitigate reasoning drift and address the "lost in the middle" challenges inherent in multi-hop question answering. By augmenting local relational subgraphs with a Bayesian-driven dynamic evolution strategy, our approach maintains logical coherence and evidence persistence throughout complex reasoning trajectories. Extensive experiments across multiple challenging benchmarks demonstrate that STAGE-RAG strikes an optimal balance between retrieval accuracy and computational efficiency, outperforming existing structural RAG systems. Future work will explore extending STAGE-RAG to global search scenarios and investigating adaptive mechanisms for real-world, scalable deployment in broader information synthesis tasks.

## Limitations

Although our experiments demonstrate that STAGE-RAG effectively improves multi-hop retrieval and reasoning under constrained inference budgets, the current framework relies on query-specific local subgraphs rather than a fully global corpus-level topology. This localized design improves efficiency and controllability, but may overlook long-range latent associations across highly disconnected documents during early retrieval stages. In addition, the framework depends on the quality of upstream entity extraction and query decomposition. Errors introduced in these stages can propagate through the iterative state evolution process and affect subsequent retrieval and reasoning trajectories. Finally, the current system operates under a constrained hop setting to maintain stable latency and token costs, which may limit its ability to handle open-ended problems requiring substantially deeper reasoning chains. We therefore do not claim strong performance for unconstrained exploratory reasoning scenarios in the current system.

## References


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*.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2025. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3639–3664, Vienna, Austria. Association for Computational Linguistics.

Shengyuan Chen, Chuang Zhou, Zheng Yuan, Qinggang Zhang, Zeyang Cui, Hao Chen, Yilin Xiao, Jiannong Cao, and Xiao Huang. 2026. You don't need pre-built graphs for rag: Retrieval augmented generation with adaptive reasoning structures. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 30270–30278.

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

Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov. 2024. Don't hallucinate, abstain: Identifying LLM knowledge gaps via multi-LLM collaboration. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14664–14690, Bangkok, Thailand. Association for Computational Linguistics.

Abdellah Ghassel, Ian Robinson, Gabriel Tanase, Hal Cooper, Bryan Thompson, Zhen Han, Vassilis N. Ioannidis, Soji Adeshina, and Huzefa Rangwala. 2025. Hierarchical lexical graph for enhanced multi-hop retrieval. *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*.

Anastasiia Grishina, Vadim Liventsev, Aki Härmä, and Leon Moonen. 2025. Fully autonomous programming using iterative multi-agent debugging with large language models. *ACM Transactions on Evolutionary Learning*, 5(1):1–37.

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

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From RAG to memory: Non-parametric continual learning for large language models. In *Forty-second International Conference on Machine Learning*.

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*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in Neural Information Processing Systems*, 37:59532–59569.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Dahyun Lee, Yongrae Jo, Haeju Park, and Moontae Lee. 2025a. Shifting from ranking to set selection for retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17606–17619, Vienna, Austria. Association for Computational Linguistics.

Dosung Lee, Wonjun Oh, Boyoung Kim, Minyoung Kim, Joonsuk Park, and Paul Hongsuck Seo. 2025b. ReSCORE: Label-free iterative retriever training for multi-hop question answering with relevance-consistency supervision. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 341–359, Vienna, Austria. 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, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Minghan Li, Junjie Zou, Xinxuan Lv, Chao Zhang, and Guodong Zhou. 2026. S2g-rag: Structured sufficiency and gap judging for iterative retrieval-augmented qa. *arXiv preprint arXiv:2604.23783*.

Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, and 1 others. 2025. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *ACM Computing Surveys*, 58(3):1–39.

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

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

Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, and 1 others. 2025b. Large language model agent: A survey on methodology, applications and challenges. *arXiv preprint arXiv:2503.21460*.

Chuangtao Ma, Yongrui Chen, Tianxing Wu, Arijit Khan, and Haofen Wang. 2025a. Large language models meet knowledge graphs for question answering: Synthesis and opportunities. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24578–24597, Suzhou, China. Association for Computational Linguistics.

Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. 2025b. Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation. In *The Thirteenth International Conference on Learning Representations*.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2025. Graph retrieval-augmented generation: A survey. *ACM Transactions on Information Systems*, 44(2):1–52.

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S Yu. 2026. Large language models meet nlp: A survey. *Frontiers of Computer Science*, 20(11):2011361.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Xavier Schmitt, Sylvain Kubler, Jérémy Robert, Mike Papadakis, and Yves LeTraon. 2019. A replicable comparison study of ner software: Stanfordnlp, nltk, opennlp, spacy, gate. In *2019 sixth international conference on social networks analysis, management and security (SNAMS)*, pages 338–343. IEEE.

Julian Schnitzler, Xanh Ho, Jiahao Huang, Florian Boudin, Saku Sugawara, and Akiko Aizawa. 2024. Morehopqa: More than multi-hop reasoning. *arXiv preprint arXiv:2406.13397*.

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

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou,

and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yaxiong Wu, Yongyue Zhang, Sheng Liang, and Yong Liu. 2025. Sgmem: Sentence graph memory for long-term conversational agents. *arXiv preprint arXiv:2509.21212*.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 641–649, New York, NY, USA. Association for Computing Machinery.

Zhao Xinjie, Fan Gao, Xingyu Song, Yingjian Chen, Rui Yang, Yanran Fu, Yuyang Wang, Yusuke Iwasawa, Yutaka Matsuo, and Irene Li. 2025. ReAgent: Reversible multi-agent reasoning for knowledge-enhanced multi-hop QA. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4067–4089, Suzhou, China. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024. Do large language models latently perform multi-hop reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10210–10229, Bangkok, Thailand. 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.

Linhao Ye, Lang Yu, Zhikai Lei, Qin Chen, Jie Zhou, and Liang He. 2025. Optimizing question semantic space for dynamic retrieval-augmented multi-hop question answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17814–17824, Vienna, Austria. Association for Computational Linguistics.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *Advances in Neural Information Processing Systems*, 37:121156–121184.

Ningning Zhang, Chi Zhang, Zhizhong Tan, Xingxing Yang, Weiping Deng, and Wenyong Wang. 2025a. Credible plan-driven rag method for multi-hop question answering. *arXiv preprint arXiv:2504.16787*.

Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Hao Chen, Yilin Xiao, Chuang Zhou, Junnan Dong, and 1 others. 2025b. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*.

Yang Zhang, Hanlei Jin, Dan Meng, Jun Wang, and Jinghua Tan. 2024. A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. *arXiv preprint arXiv:2403.02901*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, and 1 others. 2025c. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Yaoze Zhang, Rong Wu, Pinlong Cai, Xiaoman Wang, Guohang Yan, Song Mao, Ding Wang, and Botian Shi. 2025d. Leanrag: Knowledge-graph-based generation with semantic aggregation and hierarchical retrieval. *ArXiv*, abs/2508.10391.

Yibo Zhao, Jiapeng Zhu, Ye Guo, Kangkang He, and Xiang Li. 2025. Eˆ 2graphrag: Streamlining graph-based rag for high efficiency and effectiveness. *arXiv preprint arXiv:2505.24226*.

Yingli Zhou, Yaodong Su, Youran Sun, Shu Wang, Taotao Wang, Runyuan He, Yongwei Zhang, Sicong Liang, Xilin Liu, Yuchi Ma, and 1 others. 2025. In-depth analysis of graph-based rag in a unified framework. *arXiv preprint arXiv:2503.04338*.

Rongzhi Zhu, Xiangyu Liu, Zequn Sun, Yiwei Wang, and Wei Hu. 2025. Mitigating lost-in-retrieval problems in retrieval augmented multi-hop question answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22362–22375, Vienna, Austria. Association for Computational Linguistics.

# A Implementation Details

## A.1 Data Preparation and NER

The evaluation settings for MuSiQue, HotpotQA, and 2WikiMultiHopQA strictly adhere to the LongBench settings (Bai et al., 2025), utilizing a chunk capacity of 300 with a 20 token overlap for retrieval preprocessing. Conversely, we evaluate MoreHopQA via a random sampling strategy encompassing 3-5 hop queries. As the context passages in this dataset are substantially shorter, we establish a tighter chunk capacity of 100 with a 20 token overlap.

For Named Entity Recognition (NER), we utilize spaCy's en_core_web_sm model (Schmitt et al., 2019). This configuration is selected to prioritize computational efficiency and inference throughput, striking a balance between entity extraction speed and reasonable accuracy for large-scale corpora.

Table 4: Statistics of the MHQA benchmarks used in our experiments.

| Dataset | Samples | Passage | Chunks |
|---|---|---|---|
| MuSiQue | 200 | 2218 | 12586 |
| HotpotQA | 200 | 1722 | 10302 |
| 2WikiMHQA | 200 | 1986 | 6150 |
| MoreHopQA | 200 | 400 | 1031 |

### A.2 Baseline Configurations

No modifications were made to the prompt settings used in the original baseline experiments. Comprehensive hyperparameter configurations for the graph-based baselines are detailed in Table 5.

## B Additional Results

### B.1 Analysis of Answer Synthesis Paths

Table 6 evaluates the intermediate reasoning path $A_{\text{sub}}$ against the context augmented path $A_{\text{ctx}}$. While supplying a global evidence pool via $A_{\text{ctx}}$ proves sufficient and occasionally superior for shallower retrieval tasks like HotpotQA, the $A_{\text{sub}}$ trajectory excels in rigorous deduction on datasets including MuSiQue and 2WikiMultiHopQA. A critical anomaly emerges on MoreHopQA, where $A_{\text{ctx}}$ performance drops precipitously, yielding merely 5.00 Exact Match on the Qwen3 8B architecture compared to 47.50 using $A_{\text{sub}}$. This collapse occurs because extended logical chains cause the aggregated global context to become saturated with distractors, overwhelming the attention mechanism of the language model during a single generation pass. Consequently, system design must adhere to a clear deployment boundary: $A_{\text{ctx}}$ should be prioritized exclusively for shallow queries where raw evidence density drives the final answer. Conversely, $A_{\text{sub}}$ must serve as the primary synthesis engine whenever queries demand extended deductive steps, effectively isolating intermediate reasoning to shield the language model from dense topological noise.

Table 5: Hyperparameter settings.

| Parameter | LightRAG | HyperGraphRAG |
|---|---|---|
| Retrieval Mode | Hybrid | Hybrid |
| Top-$k$ Retrieval | 40 | 60 |
| Summary Tokens | 1200 | – |
| Entity Gleaning | 0 | 0 |
| Entity Extract Gleaning | 0 | 0 |
| Entity Summary Tokens | – | 500 |

### B.2 Results on weaker LLMs

To evaluate robustness under constrained capacities, we utilize the weaker Qwen2.5-3B model(Qwen et al., 2025) detailed in Table 7. Smaller language models possess limited attention spans, making them highly susceptible to dense topological noise. Consequently, heavy graph approaches like HyperGraphRAG and LightRAG degrade significantly, frequently underperforming a vanilla Naive RAG setup. In stark contrast, STAGE-RAG demonstrates remarkable resilience and achieves state-of-the-art performance across all three benchmarks. On HotpotQA and MuSiQue, our method outperforms the strongest baseline ChainRAG by substantial margins of 8.50 EM and 4.50 EM respectively. Furthermore, STAGE-RAG perfectly matches HippoRAG2 on 2WikiMultiHopQA. This comprehensive dominance confirms our lightweight Bayesian state-tracking successfully filters peripheral distractors, delivering a high signal-to-noise context that compensates for the reasoning deficits of smaller models.

### B.3 Token distribution of STAGE-RAG

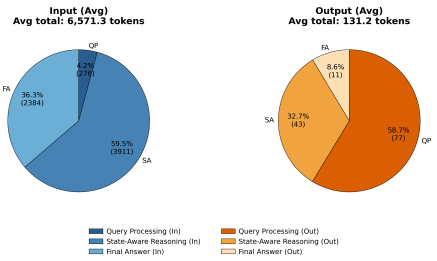


Figure 7: Average token consumption breakdown across the three operational phases of STAGE-RAG on multi-hop benchmarks. "In" and "Out" denote input and output tokens respectively.

Figure 7 illustrates the token distribution across the three core phases of STAGE-RAG (QP, SA, and FA). The data reveals a stark asymmetry between input and output tokens: the total computational cost is heavily dominated by input tokens, particularly during the State-Aware Reasoning & Evolution (SA) phase, which averages 3,600 to

Table 6: Performance breakdown of complementary synthesis paths ($A_{\text{sub}}$ and $A_{\text{ctx}}$) across multi-hop benchmarks.

| Model | Method | MuSiQue | | HotpotQA | | 2WikiMHQA | | MoreHopQA | |
|---|---|---|---|---|---|---|---|---|---|
| | | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| Qwen3-8B | ChainRAG $A_{\text{sub}}$ | 26.00 | 34.79 | 41.00 | 55.45 | 47.00 | 57.63 | 44.00 | 44.25 |
| | ChainRAG $A_{\text{ctx}}$ | 23.00 | 34.85 | 45.50 | 58.22 | 44.50 | 52.40 | 14.00 | 15.94 |
| | **Ours** $A_{\text{sub}}$ | **38.50** | **47.34** | 45.50 | 56.94 | **57.00** | **67.01** | **47.50** | **48.83** |
| | **Ours** $A_{\text{ctx}}$ | 35.50 | 43.70 | **48.00** | **62.37** | 42.50 | 52.39 | 5.00 | 7.00 |
| Qwen3-14B | ChainRAG $A_{\text{sub}}$ | 21.00 | 30.42 | 41.00 | 54.81 | 51.50 | 61.23 | 32.00 | 33.95 |
| | ChainRAG $A_{\text{ctx}}$ | 27.50 | 39.69 | 44.00 | 58.60 | 45.50 | 54.56 | 19.00 | 21.33 |
| | **Ours** $A_{\text{sub}}$ | **38.50** | **48.39** | 42.50 | 56.49 | **58.50** | **68.07** | **41.00** | **42.71** |
| | **Ours** $A_{\text{ctx}}$ | 35.50 | 48.04 | **47.50** | **63.36** | 49.50 | 60.53 | 6.00 | 8.25 |
| Qwen3-30B | ChainRAG $A_{\text{sub}}$ | 28.00 | 40.21 | 39.00 | 52.00 | 55.50 | 65.06 | 33.00 | 35.25 |
| | ChainRAG $A_{\text{ctx}}$ | 29.00 | 39.42 | 46.00 | 60.60 | 52.00 | 61.21 | 23.00 | 24.83 |
| | **Ours** $A_{\text{sub}}$ | **39.50** | **47.95** | 46.00 | 58.06 | **62.00** | **71.93** | **55.00** | **57.25** |
| | **Ours** $A_{\text{ctx}}$ | 35.50 | 45.14 | **47.50** | **62.14** | 58.50 | 69.83 | 16.00 | 18.46 |
| GPT-4o-mini | ChainRAG $A_{\text{sub}}$ | 35.00 | 45.01 | 44.00 | 57.42 | 51.00 | 60.08 | 33.00 | 34.00 |
| | ChainRAG $A_{\text{ctx}}$ | 35.50 | 45.25 | 45.00 | 59.17 | 47.00 | 57.45 | 27.00 | 27.33 |
| | **Ours** $A_{\text{sub}}$ | **42.00** | **51.61** | 44.50 | 56.90 | **57.50** | **67.96** | **47.00** | **47.00** |
| | **Ours** $A_{\text{ctx}}$ | 34.00 | 45.70 | **49.50** | **63.67** | 51.00 | 62.83 | 22.00 | 22.25 |

Table 7: Performance comparison on multi-hop QA benchmarks using the weaker Qwen2.5-3B model.

| Method | MuSiQue | | HotpotQA | | 2WikiMHQA | |
|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 |
| Vanilla LLM | 3.00 | 8.39 | 16.50 | 22.75 | 20.00 | 23.68 |
| Naive RAG | 16.50 | 23.28 | 29.50 | 42.10 | 39.00 | 46.15 |
| ChainRAG | 19.00 | 28.90 | 32.50 | 44.73 | 31.00 | 36.62 |
| LightRAG | 10.50 | 17.19 | 21.00 | 29.96 | 25.00 | 31.89 |
| HyperGraphRAG | 12.00 | 18.86 | 28.50 | 36.63 | 23.50 | 29.06 |
| HippoRAG2 | 17.00 | 25.08 | 30.00 | 41.05 | **42.00** | **50.75** |
| **Ours** | **23.50** | **32.41** | **41.00** | **50.88** | **42.00** | **50.75** |

4,100 tokens due to the iterative processing of retrieved contexts and belief state updates. In contrast, the output tokens remain remarkably concise across all stages; notably, the Final Answer Synthesis (FA) phase yields only about 10 to 13 tokens. This demonstrates that STAGE-RAG effectively distills dense multi-hop reasoning trajectories into precise, succinct final answers without introducing unnecessary linguistic redundancy.

### B.4 Impact of Different Chunk Size

Table 8: Performance comparison across varying retrieval chunk sizes using the Qwen3-8B model.

| Chunk | MuSiQue | | HotpotQA | | 2WikiMHQA | |
|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 |
| 100 | 36.50 | 45.68 | 42.50 | 57.14 | 55.00 | 63.96 |
| 300 | 38.50 | 47.34 | 48.00 | 62.37 | **57.00** | **67.01** |
| 500 | **39.50** | **49.18** | **49.50** | **63.71** | 56.00 | 66.44 |

Table 8 evaluates the influence of varying retrieval chunk capacities on reasoning efficacy. Restricting text segments to 100 tokens yields suboptimal outcomes across all datasets, indicating that fragmented contexts fail to encapsulate the logical dependencies required for multi-hop deduction. Conversely, expanding the capacity to 300 and 500 tokens demonstrates system robustness. While larger retrieval windows typically introduce text noise that degrades language model performance, our architecture maintains stable accuracy at these expanded capacities. The metrics between the 300 and 500 token configurations remain competitive, and the 300 token setting secures peak accuracy on the 2WikiMultiHopQA dataset. This resilience validates that the state update mechanism isolates and prunes peripheral distractors within longer texts, balancing efficiency and precision without requiring meticulous parameter tuning.

### B.5 Performance Analysis across Varying Hop Counts

Table 9 provides a granular performance breakdown across varying reasoning lengths on the

Table 9: Detailed performance breakdown on MuSiQue categorized by the number of reasoning hops. The best results are highlighted in bold, and the second best are underlined.

| Model | Method | MuSiQue | | |
|---|---|---|---|---|
| | | 2-Hops | 3-Hops | ≥4-Hops |
| Qwen3-8B | NaiveRAG | 25.70 | 21.11 | 26.00 |
| | ChainRAG $A_{\text{sub}}$ | 39.48 | 22.76 | 22.67 |
| | ChainRAG $A_{\text{ctx}}$ | 39.37 | 24.66 | 19.33 |
| | **Ours $A_{\text{sub}}$** | **54.77** | 26.61 | **32.67** |
| | **Ours $A_{\text{ctx}}$** | 49.97 | **30.57** | 19.33 |
| Qwen3-14B | NaiveRAG | 33.19 | 23.52 | 22.67 |
| | ChainRAG $A_{\text{sub}}$ | 44.68 | 25.96 | 29.33 |
| | ChainRAG $A_{\text{ctx}}$ | 43.22 | 27.46 | 25.33 |
| | **Ours $A_{\text{sub}}$** | **58.28** | 19.88 | 31.33 |
| | **Ours $A_{\text{ctx}}$** | 54.99 | **29.24** | **32.67** |
| Qwen3-30B | NaiveRAG | 32.28 | 21.41 | 26.00 |
| | ChainRAG $A_{\text{sub}}$ | 45.06 | 25.95 | 32.67 |
| | ChainRAG $A_{\text{ctx}}$ | 45.82 | 23.09 | 22.67 |
| | **Ours $A_{\text{sub}}$** | **54.57** | 27.70 | **39.81** |
| | **Ours $A_{\text{ctx}}$** | 51.75 | **30.14** | 22.67 |
| GPT-4o-mini | NaiveRAG | 37.73 | 31.90 | 39.33 |
| | ChainRAG $A_{\text{sub}}$ | 49.10 | **38.58** | 23.33 |
| | ChainRAG $A_{\text{ctx}}$ | 48.47 | 34.92 | 42.67 |
| | **Ours $A_{\text{sub}}$** | **55.71** | 35.24 | **56.95** |
| | **Ours $A_{\text{ctx}}$** | 49.88 | 37.03 | 29.33 |

MuSiQue benchmark. STAGE-RAG consistently maintains a substantial lead over all baselines at shallower two-hop queries, confirming the efficiency of our foundational retrieval mechanism. As the reasoning chain extends to four or more hops, a stark divergence emerges between the synthesis paradigms. The context-augmented path $A_{\text{ctx}}$ experiences severe degradation across most models due to the exponential accumulation of distractor paragraphs overwhelming the attention capacity. Conversely, the sub-answer path $A_{\text{sub}}$ exhibits exceptional resilience at extreme hop counts, achieving a commanding 56.95 F1 on GPT-4o-mini compared to 42.67 for the strongest baseline. This robust performance at extended reasoning depths empirically validates our core hypothesis, demonstrating that isolating intermediate logic into step-by-step sub-answers successfully prevents dense topological noise from paralyzing the deductive process.

## C Case Study

The qualitative trajectories detailed in Table 10 and Table 11 show the stability of STAGE-RAG against semantic drift and localized noise. In the two-hop scenario, the baseline framework fails at the initial step by retrieving a distractor segment regarding an incomplete schooling history, which misdirects the subsequent retrieval. Similarly, in the three-hop problem, the baseline extracts an irrelevant biological term, interrupting the remaining deductive chain. In contrast, STAGE-RAG tracks the entity states across varying reasoning depths. By integrating state-aware Bayesian updates, the framework maintains the correct entity belief weights when confronted with evolutionary and cultural noise, filtering out peripheral ambiguities to ensure that each generated sub-question anchors to the target logical trajectory.

## D Hyperparameter Sensitivity Analysis

To evaluate the stability of the continuous Bayesian flow and retrieval mechanisms within STAGE-RAG, we analyze three critical hyperparameters: retrieval chunk size ($k$), state pruning threshold ($\theta$), and likelihood temperature ($\tau$). Detailed F1 results are plotted in Figure 8.

**State Pruning Threshold $\theta$.** The threshold $\theta$ governs the retention of high-confidence entities transitioning into the subsequent hop. Unlike an inverted U-shape trend, Figures 8(a) and 8(b) display system robustness across a broad operational range. Performance curves remain nearly flat for $0.0 \leq \theta \leq 0.8$, indicating that lower thresholds do not propagate harmful long-tail topological noise. A sharp degradation in both $A_{\text{sub}}$ and $A_{\text{ctx}}$ is only observed under highly aggressive pruning at $\theta \geq 0.9$, which prematurely discards latent bridge entities crucial for complex chaining. Consequently, setting $\theta = 0.5$ safely distills the reasoning state without severing valid logical trajectories.

**Retrieval Capacity $k$.** The parameter $k$ dictates the size of the refined evidence set $\mathcal{C}_{\text{final}}$ post cross encoder scoring. Figures 8(c) and 8(d) indicate that a restrictive limit $k \leq 2$ truncates the contextual scope, frequently missing latent dependencies and degrading F1 scores. However, performance rapidly plateaus and remains highly stable for $k \geq 3$. This suggests the framework is resilient against denser background noise, effectively shielding the language model from distractors at larger retrieval sizes. Setting $k = 5$ therefore optimally maximizes semantic recall while preserving computational efficiency.

**Likelihood Temperature $\tau$.** The parameter $\tau$ modulates the sharpness of the observation likelihood $\boldsymbol{\mathcal{L}}$. As shown in Figures 8(e) and 8(f), the system exhibits exceptional stability against variations in $\tau$. Across several orders of magnitude

Table 10: Case study detailing reasoning trajectories on a two-hop spatial query.

| Method | Hop 1 Details | Hop 2 Details | Final Answer |
|---|---|---|---|
| **Query:** What company is the school where Lawrence Landweber was educated a part of? | | | |
| ChainRAG | **Sub-Q:** Which school did Lawrence Landweber attend?<br>**Context:** Lawrence attended The Kubert School for two years, before dropping out to focus on his stand-up. He received his bachelor's degree in 1963 at Brooklyn College.<br>**Answer:** The Kubert School | **Sub-Q:** What company is associated with The Kubert School?<br>**Context:** Notable alumni in business include Deloitte Touche Tohmatsu CEO Barry Salzberg (B.S. 1974), Adobe Inc.<br>**Answer:** None | **None** |
| **Ours** | **Sub-Q:** Where did Lawrence Landweber receive his education?<br>**Context:** Lawrence Hugh Landweber is John P. Morgridge Professor Emeritus of computer science at the University of Wisconsin–Madison. He received his bachelor's degree in 1963 at Brooklyn College.<br>**State:** Update: [Lawrence Landweber: 1.00].<br>**Ans:** Brooklyn College | **Sub-Q:** What company is Brooklyn College a part of?<br>**Context:** Brooklyn College is a public university in Brooklyn in New York City. It is part of the City University of New York system.<br>**State:** Update: [Brooklyn: 0.98].<br>**Ans:** City University of New York | **City University of New York** |

ranging from 0.01 to 100, the F1 scores remain almost entirely constant. Rather than becoming vulnerable to isolated noise at low values or neutralizing discriminative power at high values, the Bayesian update mechanism proves highly robust. Setting $\tau = 0.1$ thus ensures effective evidence integration while capitalizing on this profound insensitivity to precise temperature tuning.

## E Detailed Computational Efficiency Metrics

To provide a comprehensive view of the computational overhead discussed in Section 4.5, we present the detailed breakdown of inference latency, LLM API calls, and token consumption across the MuSiQue, HotpotQA, and 2WikiMultiHopQA datasets.

Table 12 details the average latency (in seconds) and the number of LLM calls required per query. While standard linear-chain reasoning (ChainRAG) exhibits the lowest baseline metrics, heavy graph-based paradigms (HyperGraphRAG and HippoRAG2) suffer from a severe explosion in API interactions, largely due to their reliance on LLMs for extensive edge verification and topological traversal. STAGE-RAG effectively bypasses this bottleneck by executing the continuous state updates via deterministic matrix diffusion, maintaining the LLM calls at a highly manageable level.

Table 13 further illustrates the token economy of the evaluated systems. Graph-based and dense-retrieval baselines (including LightRAG) consume an exorbitant number of tokens, particularly in output generation, as they frequently require the LLM to output extensive reasoning paths or subgraph structures. In contrast, STAGE-RAG explicitly restricts the reasoning trajectory to a maximum constraint of 3-4 essential hops and utilizes a lightweight JSON schema for state tracking. This architectural efficiency results in an overwhelming reduction in both input and output token consumption, proving the real-world scalability of our framework.

## F Prompt Engineering Details

To support the reproducibility of our experimental results, we include the precise instruction prompts employed in the main components of our framework. Prompts are designed to produce structured JSON outputs for all modules except sub-question answering and final answer integration, enabling convenient downstream processing through automated scripts.

### F.1 Question Evaluation and Decomposition Prompts

To mitigate the compositional bottleneck discussed in Section 3.3.1, we employ a dual-stage prompt-

Table 11: Case study detailing reasoning trajectories on an extended three-hop relational query.

| Method | Hop 1 Details | Hop 2 Details | Hop 3 Details | Final Answer |
|---|---|---|---|---|
| **Query:** Who is the voice of the character in SpongeBob SquarePants with the same name as the creature that annelid larvae live like? | | | | |
| ChainRAG | **Sub-Q:** What is the name of the character in SpongeBob SquarePants that shares its name with the creature that annelid larvae live like? | **Sub-Q:** What is the name of the voice actor for the character Trochophore in SpongeBob SquarePants? | – | **Not specified** |
| | **Context:** The current view is that annelids are grouped with molluscs, brachiopods and several other phyla that have lophophores and/or trochophore larvae as members of Lophotrochozoa. | **Context:** Plankton is a character in SpongeBob SquarePants, often interacting with SpongeBob. | – | |
| | **Ans:** Trochophore | **Ans:** Not specified | – | |
| **Ours** | **Sub-Q:** What is the name of the creature that annelid larvae live like? | **Sub-Q:** What is the name of the character in SpongeBob SquarePants named plankton? | **Sub-Q:** Who is the voice actor for Sheldon Plankton? | **Doug Lawrence** |
| | **Context:** The fertilized eggs develop into trochophore larvae, which live as plankton. | **Context:** 'The Algae Always Greener' and 'Plankton's Army' focused on Sheldon Plankton. | **Context:** Lawrence started out by voicing a variety of minor characters. This included Plankton, + 'it's Doug [Lawrence].' | |
| | **State:** Update: [SpongeBob SquarePants: 1.00]. | **State:** Update: [Australia: 0.98] (noise), [SpongeBob SquarePants: pruned]. | **State:** Update: [SpongeBob SquarePants: 0.72]. | |
| | **Ans:** plankton | **Ans:** Sheldon Plankton | **Ans:** Doug Lawrence | |

| Average Latency (Seconds) | | | | |
|---|---|---|---|---|
| **Method** | **MuSiQue** | **HotpotQA** | **2WikiMHQA** | **Average** |
| ChainRAG | 7.34 | 5.90 | 3.62 | 5.62 |
| HyperGraphRAG | 192.28 | 155.99 | 92.57 | 146.95 |
| HippoRAG2 | 125.68 | 80.35 | 50.48 | 85.50 |
| Ours | 8.18 | 6.66 | 4.96 | 6.60 |
| **Average LLM Calls** | | | | |
| **Method** | **MuSiQue** | **HotpotQA** | **2WikiMHQA** | **Average** |
| ChainRAG | 8.79 | 7.87 | 3.62 | 6.76 |
| HyperGraphRAG | 63.15 | 52.53 | 32.14 | 49.27 |
| HippoRAG2 | 97.87 | 104.81 | 63.21 | 88.63 |
| Ours | 11.91 | 10.65 | 11.90 | 11.49 |

Table 12: Detailed breakdown of inference latency and LLM calls per query across datasets.

ing mechanism illustrated in Figure 10 and Figure 12. The initial judge prompt evaluates whether the incoming query necessitates multi-hop reasoning. For questions requiring complex deduction, the subsequent decomposition prompt instructs the LLM to break the query down into a logical sequence of minimal subqueries. To prioritize reasoning efficiency and prevent unnecessary computational overhead, this decomposition process strictly constrains the reasoning chain to a maximum of three to four essential steps.

### F.2 Adaptive Question Rewriting Prompt

To execute the history-driven gating operator $\mathcal{G}$ and rewriting operator $\mathcal{R}$ defined in Equation 4, we deploy the dual instruction configuration illustrated in Figure 9. The instruction detailed in the upper panel evaluates whether the current sub-question $q_t$ contains unresolved anaphoric references dependent on the reasoning history $\mathcal{H}_{t-1}$. When the gating mechanism dictates resolution is necessary, the instruction presented in the lower panel integrates the historical context to generate the standalone query $q'_t$.

### F.3 Reasoning Synthesis and Answer Generation Prompts

At each hop, upon following the retrieval and reranking of the local subgraph to isolate the final evidence set, the instruction shown in the top panel of Figure 11 generates the grounded intermediate answer $a_t$ used to construct the reasoning history. This process prioritizes conciseness to maintain a clean deductive trajectory.

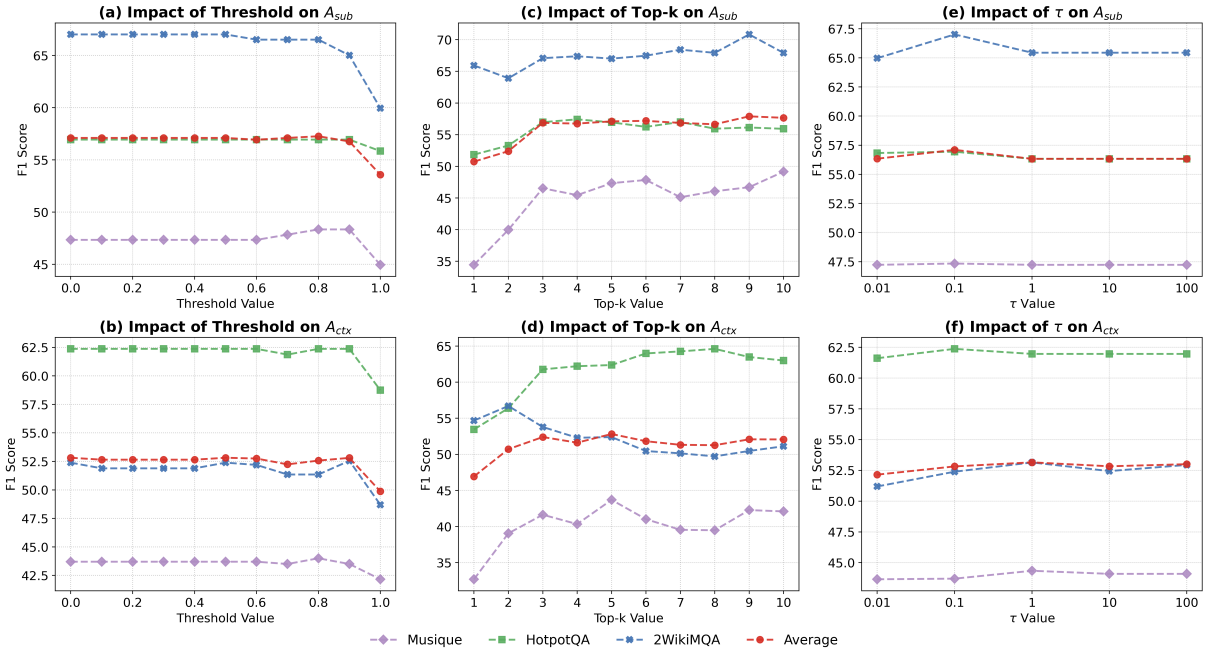


Figure 8: Parameter sensitivity analysis on the multi-hop QA datasets using the Qwen3-8B backbone. The subplots illustrate the impact of varying the Threshold, Top-$k$, and $\tau$ on the F1 performance of $A_{\text{sub}}$ (top row) and $A_{\text{ctx}}$ (bottom row) across different task categories. The curves represent the performance on individual datasets alongside the overall average. The relatively flat curves across wide ranges of these values indicate the system's robustness to hyperparameter variations.

| Method | MuSiQue | | | HotpotQA | | | 2WikiMHQA | | |
|---|---|---|---|---|---|---|---|---|---|
| | Input | Output | Total | Input | Output | Total | Input | Output | Total |
| ChainRAG | 3,113.0 | 76.2 | 3,189.2 | 2,679.0 | 66.3 | 2,745.3 | 3,089.7 | 70.1 | 3,159.8 |
| LightRAG | 210,493.9 | 64,029.6 | 274,523.6 | 176,805.2 | 53,182.5 | 229,987.7 | 109,926.6 | 33,086.2 | 143,012.9 |
| HyperGraphRAG | 267,842.5 | 84,583.4 | 352,425.9 | 225,139.2 | 70,118.5 | 295,257.6 | 140,948.0 | 38,378.9 | 179,326.9 |
| HippoRAG2 | 84,629.4 | 43,949.8 | 128,579.3 | 70,962.8 | 37,199.6 | 108,162.3 | 43,227.0 | 21,768.0 | 64,995.0 |
| Ours | 7,939.5 | 133.7 | 8,073.2 | 7,305.7 | 125.7 | 7,431.4 | 7,468.8 | 134.1 | 7,602.9 |

Table 13: Average token consumption (Input, Output, and Total) per query across different datasets.

For the terminal synthesis phase defined in Section 3.4, the framework utilizes two distinct generation paths to ensure absolute factual grounding. The instruction illustrated in the middle panel conditions the model on the complete reasoning history $\mathcal{H}_T$ to derive the path specific answer $A_{\text{sub}}$. Simultaneously, the instruction presented in the bottom panel leverages the prioritized global evidence pool $\mathcal{C}_{\text{syn}}$ to formulate $A_{\text{ctx}}$. This dual path synthesis ensures that the final response integrates both the step by step logical deduction and the raw source evidence.

**Dependency Resolution Prompt**

```
### Previous Q&A History:
        {history_text}
### Current Question:
        {sub_q}
Does the current question refer to the answer of the
previous question? Answer yes or no.
```

**Question Rewriting Prompt**

```
 Rewrite the following question to be self-contained
by replacing pronouns or references with the actual
entities they refer to.
### Previous Q&A History:
        {history_text}
### Current Question:
        {sub_q}
Rewritten question:
```

Figure 9: Adaptive query formulation prompts. Top: dependency resolution prompt for gating-based historical reference detection. Bottom: rewriting prompt for standalone subquery generation.

**Multi-Hop Judge Prompt**

```
You are a helpful AI assistant that determines if a
question requires multiple steps to answer.

Guidelines for identifying multi-hop questions:
  1. The question requires finding and connecting
multiple pieces of information
  2. The answer cannot be found in a single direct
statement
  3. You need to find intermediate information to reach
the final answer

Output format should be a JSON object with only one
fields: "is_multi_hop": boolean (true/false)

Example:
  Question: "Who is the paternal grandmother of
Marie Of Brabant, Queen Of France?"
  Output: {"is_multi_hop": false}
  Question: "Who is Archibald Acheson, 4Th Earl Of
Gosford's paternal grandfather?"
  Output: {"is_multi_hop": false}
  Question: "Which song came out first, Joel The
Lump Of Coal or Jugband Blues?"
  Output: {"is_multi_hop": true}
  Question: "Who was the wife of the person who
founded Microsoft?"
  Output: {"is_multi_hop": true}
```

Figure 10: The Judge Prompt utilized to determine the reasoning complexity of incoming queries.

**Sub-question Answering Prompt**

```
Task: Answer the question based on the specific
context provided.

  CRITICAL GUIDELINES:
  1. CONCISENESS: Provide the answer with the
fewest words possible.
  2. FORMAT: Do not output any intro words, just the
entity name.
  Context: {context}
  Question: {question}

  Answer:
```

**$A_{\text{sub}}$ Generation Prompt**

```
Based on the answers to the sub-questions, use the
fewest words possible to answer the original
question.
Only give me the answer and do not output any other
words.
  Original Question: {original_question}
  Sub-questions and their answers:{sub_answers}
  Provide the shortest possible answer:
```

**$A_{\text{ctx}}$ Generation Prompt**

```
Based on the following context, use the fewest words
possible to answer the original question.
Only give me the answer and do not output any other
words.
   Context: {context}
   Question: {question}
   Answer:
```

Figure 11: Prompts for intermediate and terminal answer synthesis. Top: subquery answering prompt. Middle and bottom: complementary synthesis prompts for $A_{\text{sub}}$ and $A_{\text{ctx}}$, respectively.

**Question Decompostion Prompt**

You are a helpful AI assistant that breaks down complex questions into a logical chain of minimal necessary sub-questions.

Guidelines:
1. **Identify Bridge Entities**: For multi-hop reasoning, identify the "bridge" entities that connect the start of the question to the answer (e.g., Movie -> Producer -> Character -> Actor).
2. **Sequential Dependency**: Ensure sub-questions follow a strict logical order where step N often depends on the answer from step N-1.
- If a step relies on finding a specific person/object first, formulate the question to explicitly look for that entity.
3. **Comparison Logic**: For comparison questions (A vs B):
- Identify the specific attribute being compared.
- Ask about that attribute for A.
- Ask about that attribute for B.
- (Optional) Add a final comparison step if the calculation is complex.
4. **Avoid Redundancy**: Do not ask for information that is already provided in the context of the original question.
5. **Constraint**: Keep the total number of sub-questions minimal necessary (usually 2-4 steps). Do not exceed 4 unless absolutely necessary.
Output format must be a JSON array of strings.

Logical Progression Types & Examples:
Type A: Parallel
Original: "What are the causes and consequences of climate change?"
Output: [
"What are the main causes of climate change?",
"What are the major consequences of climate change?"
]
Type B: Comparative
Original: "Which magazine was first published earlier, Time or Vogue?"
Output: [
"When was the magazine Time first published?",
"When was the magazine Vogue first published??"
]
Type C: Sequential Bridging
Original: "What is the birth city of the director of the film 'Inception'?"
Output: [
"Who is the director of the film 'Inception'?",
"What is the birth city of the director of 'Inception'?"
]
Type D: Deep Chaining
Original: "Who plays the wife of the producer of Here Comes the Boom in Grown Ups?"
Output: [
"Who is the producer of the film Here Comes the Boom?",
"What character does the producer of Here Comes the Boom play in the film Grown Ups?",
"Who is the wife of that character in the film Grown Ups?",
"Which actress plays the wife of that character in Grown Ups?"
]

Question: {question}
Break down this question into minimal necessary sub-questions:

Figure 12: The Decomposition Prompt designed to break down complex queries into minimal sequential subqueries.