ECG-RAG: EVIDENCE CHAIN-GUIDED RETRIEVAL FOR MULTI-HOP QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-Augmented Generation (RAG) is widely used to improve the factual grounding of Large Language Models (LLMs) by leveraging external knowledge. While effective for simple queries, standard RAG systems struggle with multi-hop questions where the necessary evidence is scattered across multiple passages. Recent graph-based methods capture relations among entities and text units to support broader context retrieval for complex reasoning tasks. However, existing dense and graph-based retrievers often optimize local relevance or neighborhood coverage, without explicitly ensuring that the retrieved evidence forms a complete support chain from the question to the answer. In this paper, we propose Evidence Chain-Guided Retrieval-Augmented Generation (ECG-RAG), a retrieval framework that explicitly targets evidence-chain completeness for multi-hop RAG. To support this goal, ECG-RAG adopts a hierarchical hypergraph that aligns atomic entities, pairwise relations, high-level multi-entity relations, and source passages in a unified retrieval space. At inference time, ECG-RAG parses each query into an evidence-chain goal and activates anchor nodes in the pairwise-relation layer. It then uses ant-inspired exploration to dynamically prioritize relation transitions that advance this goal, thereby deriving a compact query-specific local relation graph for subsequent path search. A*-style search is then applied over this local graph to assemble evidence paths that cover the target chain. The selected paths are projected back to grounded source passages and organized into path-structured evidence context for answer generation. Extensive experiments on six QA benchmarks show that ECG-RAG achieves the best overall QA performance, reaching 60.4 EM and 66.6 F1 with average gains of 2.1 EM and 0.7 F1 over the strongest baseline. Retrieval diagnostics on 2WikiMultiHopQA show 93.6% Evidence Recall, with all annotated supporting evidence covered in the final generation context for 85.0% of the questions.

1 INTRODUCTION

Retrieval-augmented generation (RAG) improves the factual grounding of large language models (LLMs) by retrieving external evidence at inference time and conditioning generation on the retrieved context (Lewis et al., 2020; Gao et al., 2023). This paradigm is effective for fact-seeking questions whose answers can be supported by a few directly relevant passages.

Multi-hop question answering poses a more demanding retrieval problem. Unlike simple queries, multi-hop questions require several facts to be connected in a specific reasoning order (Trivedi et al., 2022; Ho et al., 2020; Yang et al., 2018). These facts are often distributed across passages and linked through intermediate entities or relations that may not be explicitly mentioned in the question (Press et al., 2023; Zhang et al., 2025). Thus, the value of a retrieved passage depends not only on its local relevance, but also on whether it connects to other evidence needed to support the final answer.

Recent graph-based RAG methods introduce structured retrieval spaces over entities, passages, relations, or communities to capture cross-passage associations (Edge et al., 2024; Guo et al., 2025; Jimenez Gutierrez et al., 2024; Xu et al., 2025). Although these structures retrieve beyond isolated chunks, many dense and graph-based retrievers remain driven by local matching, structural proximity, neighborhood expansion, or subgraph coverage (Jimenez Gutierrez et al., 2024; Chen et al., 2026; Li et al., 2025a). Such objectives improve retrieval breadth but do not explicitly optimize

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

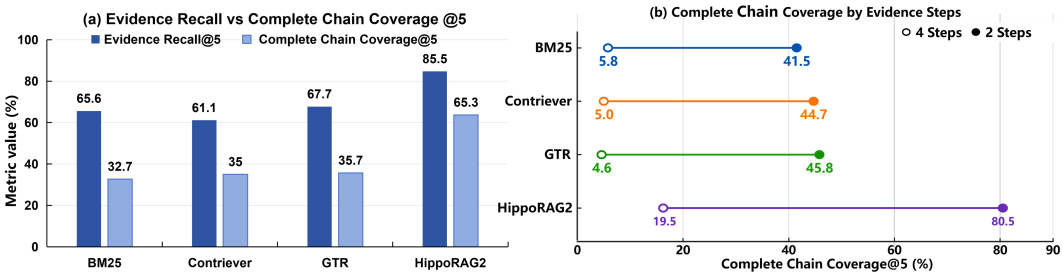

Figure 1: Preliminary analysis on 2WikiMultiHopQA. Existing retrievers often achieve moderate evidence recall but substantially lower complete chain coverage, especially for questions requiring more evidence steps.

evidence-chain completeness, so the generator may still receive related fragments that fail to form a complete support chain.

We empirically observe this gap on 1,000 samples from 2WikiMultiHopQA (Ho et al., 2020). As shown in Figure 1, existing retrievers achieve moderate to high Evidence Recall@5, yet substantially lower Complete Chain Coverage@5. Evidence Recall@5 measures the fraction of gold supporting evidence units in the top-5 results, whereas Complete Chain Coverage@5 requires the full gold evidence chain to be retrieved. BM25 (Robertson & Walker, 1994) reaches 65.6% Evidence Recall@5 but only 32.7% Complete Chain Coverage@5, and GTR shows a similar gap of 67.7% versus 35.7%. The gap also persists for graph-based retrieval: HippoRAG2 (Gutiérrez et al., 2025) reaches 85.5% Evidence Recall@5 but 65.3% Complete Chain Coverage@5, dropping from 80.5% on questions with two evidence steps to 19.5% on questions with four evidence steps. These results suggest that retrieving individually relevant evidence is insufficient for multi-hop QA when the complete support chain is not recovered.

To address this problem, we propose Evidence Chain-Guided Retrieval-Augmented Generation (ECG-RAG), a framework for recovering complete support chains in multi-hop RAG. ECG-RAG models retrieval as goal-guided search over evidence chains. It adopts a hierarchical hypergraph index that organizes atomic entities, pairwise relations, high-level relations, and grounding links to source passages. At inference time, ECG-RAG derives an evidence-chain goal, activates anchor relations in the pairwise-relation layer, uses ant-inspired exploration (Dorigo et al., 1996) to estimate query-specific transition preferences and derive a compact local relation graph, and applies A*-style path search (Hart et al., 1968) to assemble relation paths that cover the target chain. The selected paths are finally projected back to source passages and organized into path-structured context for answer generation.

We evaluate ECG-RAG on six QA benchmarks covering simple, multi-hop, and explanatory QA. ECG-RAG achieves the best overall performance, reaching 60.4 EM and 66.6 F1, with average gains of 2.1 EM and 0.7 F1 over the strongest baseline. Retrieval diagnostics on 2WikiMultiHopQA show 93.6% Evidence Recall and 85.0% Complete Chain Coverage in the final generation context. These results indicate that explicitly modeling evidence-chain completeness improves both support-chain recovery and downstream QA performance.

Our contributions are summarized as follows:

- We reveal a key limitation of existing multi-hop RAG retrieval objectives: high evidence recall does not necessarily imply complete support-chain coverage. We therefore identify evidence-chain completeness as a more faithful retrieval objective for multi-hop question answering.
- We propose ECG-RAG, a goal-guided evidence-chain retrieval method for recovering complete support chains. Guided by a query-derived chain goal, ant-inspired exploration estimates query-specific transition preferences and derives a compact local relation graph; A*-style search then uses this graph and these preferences to assemble chain-aligned evidence paths that cover the required support chain for answer generation.

- Experiments on six QA benchmarks show that ECG-RAG achieves the best overall QA performance among competitive dense, graph-based, and hypergraph-based RAG baselines. Retrieval diagnostics further confirm that ECG-RAG recovers more complete evidence chains.

2 Related Work

Retrieval-Augmented Generation. Retrieval-augmented generation (RAG) enhances large language models (LLMs) by retrieving external evidence and conditioning generation on the retrieved context (Lewis et al., 2020; Gao et al., 2023). Recent work further makes retrieval more adaptive through query decomposition, iterative retrieval-generation, and task-adaptive representations (Trivedi et al., 2023; Li et al., 2025b). Despite these advances, many retrieval objectives remain centered on local query-passage relevance. For multi-hop questions, retrieving individually relevant passages may still fail to recover the distributed evidence required for complete reasoning (Press et al., 2023; Zhang et al., 2025). ECG-RAG addresses this limitation by making evidence-chain completeness an explicit retrieval objective.

Graph-Structured Retrieval for RAG. Graph-structured RAG methods organize corpora into graph or hierarchical retrieval spaces to capture cross-passage associations beyond isolated text chunks (Xiang et al., 2026). Representative approaches span entity- and community-centric graphs, hierarchical summaries, memory-inspired graph retrieval, relational-path retrieval, and hypergraph-based retrieval spaces (Edge et al., 2024; Sarthi et al., 2024; Guo et al., 2025; Jimenez Gutierrez et al., 2024; Gutiérrez et al., 2025; Xu et al., 2025; Chen et al., 2026; Feng et al., 2026; Hu et al., 2026; Hou et al., 2026). These methods improve how RAG expands, summarizes, prunes, or traverses graph-structured context, typically emphasizing query relevance, structural proximity, neighborhood or community coverage, or salient graph structures (Li et al., 2025a; Zhou et al., 2026).

Path-Based Retrieval and Evidence-Chain Optimization. Recent work has explored relational paths for structured retrieval and evidence chains for improving retrieved-context quality. PathRAG retrieves salient relational paths from an indexing graph to reduce redundant graph context (Chen et al., 2026). NeocorRAG mines evidence chains from an initially retrieved document set and uses them to refine retrieval quality (Peng et al., 2026). In contrast, ECG-RAG takes evidence-chain completeness itself as the retrieval objective for multi-hop QA. It starts from a query-specific evidence-chain goal and organizes retrieval around recovering the complete support chain required by the question. This shifts the focus from retrieving relevant evidence fragments to explicitly ensuring the completeness of the evidence required for multi-hop reasoning.

3 Method

As shown in Figure 2, ECG-RAG has three stages: H^3G index construction, evidence-chain-guided retrieval, and path-structured context construction. It builds a grounded hierarchical hypergraph offline, retrieves relation paths guided by a query-derived evidence-chain goal, and maps the selected paths back to source passages for answer generation.

3.1 Index Construction

Following the H^3G-style formulation introduced by IGMiRAG (Hou et al., 2026), ECG-RAG builds an offline hierarchical index that represents corpus knowledge at multiple granularities and grounds it to source passages. Given a corpus $\mathcal{D}$, documents are segmented into source passages $\mathcal{P}$. An LLM-based extractor identifies three types of knowledge units: atomic entities $\mathcal{N}$, pairwise relations $\mathcal{L}$, and high-level relations $\mathcal{H}$. The graph component of the H^3G index is defined as

$$\mathcal{G}_{\mathrm{H^3G}} = (\mathcal{V}, \mathcal{E}), \qquad \mathcal{V} = \mathcal{N} \cup \mathcal{L} \cup \mathcal{H}, \qquad \mathcal{E} = \mathcal{E}_{LR} \cup \mathcal{E}_{HR}. \tag{1}$$

Here, $\mathcal{E}_{LR}$ connects entity vertices with pairwise-relation vertices, and $\mathcal{E}_{HR}$ connects entity vertices with high-level-relation vertices. Each vertex $v \in \mathcal{V}$ stores textual metadata $v = \langle n_v, d_v, \mathbf{a}_v \rangle$, where n_v, d_v, and $\mathbf{a}_v$ denote its name, semantic description, and auxiliary attributes. For relation vertices, the attributes record their participating entities.

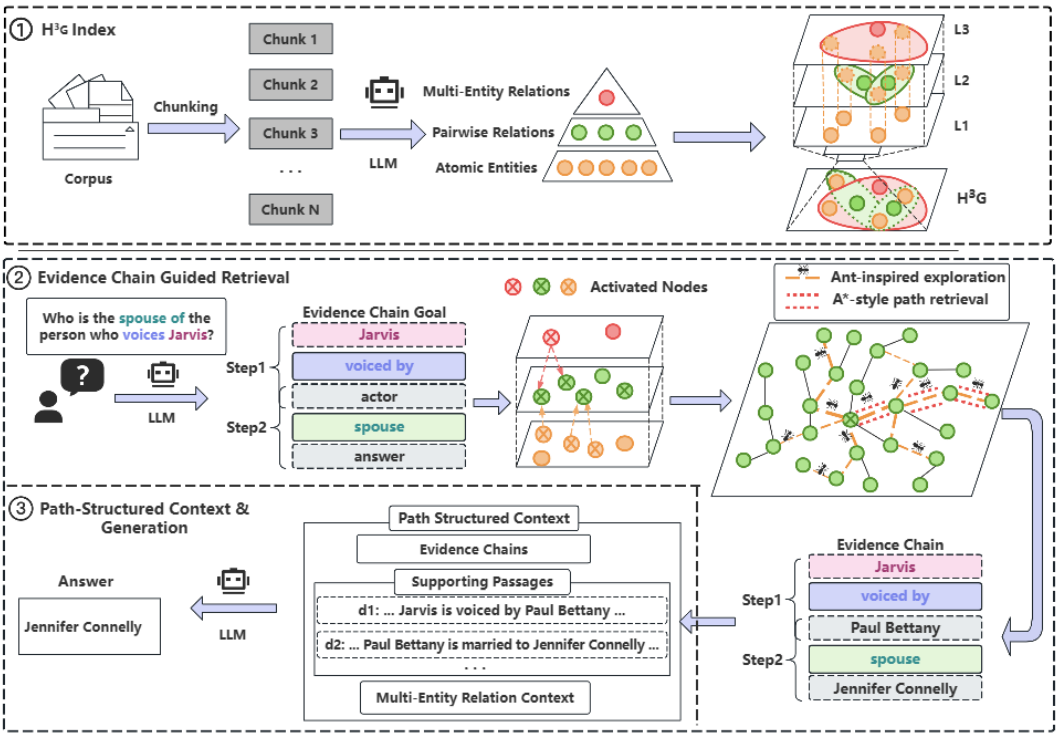

Figure 2: Overview of ECG-RAG. It builds a hierarchical hypergraph index offline and, online, derives an evidence-chain goal to guide relation exploration and A*-style path retrieval for path-structured answer generation.

To preserve source grounding, ECG-RAG maintains links $F_R \subseteq \mathcal{V} \times \mathcal{P}$, where $(v, p) \in F_R$ indicates that vertex v is extracted from or supported by passage p. The online stage uses the pairwise-relation layer $\mathcal{L}$ as the primary search space because each $l \in \mathcal{L}$ represents a grounded local factual transition. Entity and high-level-relation vertices provide structural context for activation and expansion, while F_R maps selected relation paths back to textual evidence for answer generation.

3.2 Evidence-Chain-Guided Online Retrieval

Given a question q and the H^{3}G index $\mathcal{G}_{\mathrm{H^3G}}$, ECG-RAG retrieves relation paths that cover the evidence-chain goal. Instead of ranking passages independently, retrieval operates on the pairwise-relation layer $\mathcal{L}$, whose nodes represent grounded local factual transitions.

The retrieval process consists of four steps: evidence-chain goal analysis, anchor relation activation, ant-inspired relation exploration, and A*-style path retrieval. Together, these steps transform the question into goal-aligned relation paths for subsequent passage projection and context construction.

Evidence-chain goal analysis. Multi-hop questions provide retrieval cues at different levels: entity mentions suggest starting points, relational phrases describe factual transitions, and answer constraints indicate what information must ultimately be resolved (Trivedi et al., 2023; Luo et al., 2024; Zhang et al., 2025). ECG-RAG converts these cues into an explicit evidence-chain retrieval goal.

Given a question q, an LLM-based query analyzer produces $\mathcal{A}_q = (\mathcal{K}_q^e, \mathcal{K}_q^w, \mathcal{T}_q)$, where $\mathcal{K}_q^e$ contains key entities, $\mathcal{K}_q^w$ contains retrieval keywords and relation cues, and $\mathcal{T}_q$ specifies the factual transitions to be recovered. We represent $\mathcal{T}_q$ as an ordered list $(g_1, \ldots, g_m)$, where $g_t = (s_t, r_t, o_t)$ denotes a relation r_t to be established between arguments s_t and o_t. Each argument can be an explicit entity or an unresolved semantic slot, which is grounded during retrieval and reused by dependent transitions. Dependencies indicate when a later factual step relies on information grounded by an earlier one.

For questions requiring comparison or selection, such answer-derivation operations are handled during generation rather than included in $\mathcal{T}_q$ as retrieval targets. ECG-RAG uses $\mathcal{K}_q^e$ and $\mathcal{K}_q^w$ to activate candidate relation anchors, while $\mathcal{T}_q$ guides subsequent relation exploration and path retrieval toward the required support chain.

Anchor relation activation. Given $\mathcal{A}_q$, ECG-RAG activates a compact anchor set in the pairwise-relation layer $\mathcal{L}$, where each node represents a grounded local factual transition. Candidate pairwise-relation nodes are recalled through complementary activation channels. Relation cues match query-derived relation expressions with pairwise-relation descriptions through lexical and dense retrieval (Robertson & Walker, 1994; Gao et al., 2023); entity cues activate query-relevant entities and collect their associated pairwise relations; and high-level relation cues retrieve high-level relation nodes and project their connected pairwise relations back to $\mathcal{L}$.

Let $\mathcal{R}_q$ denote the ranked lists returned by these channels, and let $\mathcal{C}_L(q)$ be their union. For each $l \in \mathcal{C}_L(q)$, ECG-RAG computes an activation score by weighted rank fusion (Cormack et al., 2009):

$$s_{\mathrm{act}}(l, q) = \sum_{R \in \mathcal{R}_q : l \in R} \frac{\alpha_R}{k_0 + \mathrm{rank}_R(l)}. \tag{2}$$

Here, $\mathrm{rank}_R(l)$ is the rank of l in list R, α_R is the channel weight, and k_0 is a smoothing constant. The anchor set is selected as $\mathcal{S}_L(q) = \mathrm{Top}\mathrm{M}_{l \in \mathcal{C}_L(q)}\, s_{\mathrm{act}}(l, q)$. These anchors initialize ant-inspired relation exploration and provide starting relations for subsequent A*-style path retrieval.

Ant-inspired relation exploration. The activated anchors provide entry points into the relation layer, but the multi-hop transitions required to complete the evidence-chain goal remain unknown. Expanding all neighboring relations may introduce substantial noise. ECG-RAG therefore performs ant-inspired exploration (Dorigo et al., 1996) to estimate query-specific transition preferences and construct a compact local relation graph for subsequent path retrieval.

Given the activated anchor set $\mathcal{S}_L(q)$, ECG-RAG launches multiple ants from the anchor relations. Each ant maintains a partial relation path P, a covered-step set C, and a slot-binding map B. The binding map preserves intermediate entities or values grounded by previous steps, enforcing consistency across dependent transitions. Each walk is additionally constrained by a maximum depth D and exploration energy E, initialized as E_0.

At relation node l_i, candidate neighbors are obtained through H^{3}G projection. Two pairwise-relation nodes are adjacent if they share at least one participating entity, corresponding to a relation–entity–relation transition in H^{3}G. Previously visited nodes are removed, yielding $\mathcal{N}_q(l_i, P)$.

Each candidate $l_j \in \mathcal{N}_q(l_i, P)$ is evaluated by combining local contextual support and evidence-chain progress:

$$\eta_q(l_j, C, B) = 1 - \big(1 - u_q(l_j)\big)\big(1 - a_q(l_j, C, B)\big), \tag{3}$$

where $u_q(l_j) \in [0, 1]$ measures contextual support based on query relevance, path consistency, and grounding evidence. The progress term is defined as

$$\begin{aligned} \mathcal{M}_q(l_j, C, B) &= \{t \in \{1, \ldots, m\} \setminus C : g_t \text{ is eligible given } C \wedge \sigma_q(l_j, g_t; B) \geq \delta\}, \\ a_q(l_j, C, B) &= \frac{|\mathcal{M}_q(l_j, C, B)|}{m}. \end{aligned} \tag{4}$$

A step is eligible only after its prerequisites have been covered. The compatibility score $\sigma_q(l_j, g_t; B)$ evaluates relation and argument consistency under the current bindings. Explicit arguments are matched through exact, alias, or embedding-based matching, while unresolved slots are grounded from matched arguments of l_j. Newly grounded entities or values are added to B and reused by dependent transitions.

Each ant samples the next relation according to

$$p_q(l_j \mid l_i, C, B) = \frac{\tilde{\tau}_q(l_i, l_j)^\alpha \eta_q(l_j, C, B)^\beta}{\sum_{l_k \in \mathcal{N}_q(l_i, P)} \tilde{\tau}_q(l_i, l_k)^\alpha \eta_q(l_k, C, B)^\beta}, \tag{5}$$

where $\tilde{\tau}_q(l_i, l_j) = \max(\Delta\tau_q(l_i, l_j), \epsilon)$. The floor ϵ preserves exploration of previously unvisited transitions, while $\Delta\tau_q$ accumulates query-specific transition preferences.

After moving to l_j, the exploration energy is updated as $E \leftarrow E - \frac{E_0/D}{1+\eta_q(l_j,C,B)}$, and the selected transition is reinforced by

$$\Delta\tau_q(l_i,l_j) \leftarrow \Delta\tau_q(l_i,l_j) + r_q(l_i,l_j,C,B), \qquad r_q(l_i,l_j,C,B) = \frac{\eta_q(l_j,C,B) + a_q(l_j,C,B)}{2}. \tag{6}$$

The covered-step set and binding map are updated using $\mathcal{M}_q(l_j,C,B)$. A walk terminates when it reaches depth D, satisfies $E < E_{\min}$, has no valid neighbor, or covers the evidence-chain goal.

After exploration, the visited relation nodes and selected transitions form the query-specific local relation graph $\mathcal{G}_q^L = (\mathcal{L}_q, \mathcal{E}_q^L)$. The accumulated transition preferences are normalized into τ_q and used for subsequent goal-guided path retrieval.

Goal-guided relation path retrieval. Given the local relation graph $\mathcal{G}_q^L$ and the normalized transition map $\tau_q : \mathcal{E}_q^L \to [0,1]$, ECG-RAG performs A*-style heuristic best-first search (Hart et al., 1968) to assemble relation paths that cover the factual steps in the evidence-chain goal. The search prioritizes partial relation paths according to transition preference and goal progress.

A search state is defined as $x = (P, l, C, B)$, where $P = (l_0, \ldots, l_t)$ is a partial relation path, $l = l_t$ is the current relation node, C contains the factual steps grounded by P, and B is the current slot-binding map. A state is target-reaching when all m factual transitions in $\mathcal{T}_q$ have been covered, i.e., $|C| = m$.

Each anchor $l_0 \in \mathcal{S}_L(q)$ initializes a state $x_0 = ((l_0), l_0, C_0, B_0)$, where (C_0, B_0) is obtained by matching l_0 against the factual steps under an empty binding map. States are ranked in a priority queue by $f(x) = g(x) + h(x)$, with

$$g(x) = \sum_{i=0}^{t-1} \big(1 - \tau_q(l_i, l_{i+1})\big), \qquad h(x) = 1 - \frac{|C|}{m}. \tag{7}$$

Thus, paths with stronger transition preferences and greater factual-step coverage receive higher expansion priority.

At each iteration, ECG-RAG pops the state with the lowest $f(x)$. If the state is target-reaching, its path is retained. Otherwise, it is expanded to unvisited neighbors l' of l in $\mathcal{G}_q^L$, producing successor states $x' = (P \circ l', l', C', B')$. The coverage and bindings are updated by matching l' against eligible factual steps under the current binding map. Only steps whose dependencies and argument constraints are satisfied can be added to C'. Before queue insertion, dominated successors with the same current node, covered-step set, and bindings are pruned if they have higher cost without additional grounding evidence.

Under a fixed expansion budget, the search retains up to K terminal candidate paths. Target-reaching paths are prioritized over incomplete candidates; when fewer than K target-reaching paths are available, the remaining slots are filled with partial paths according to goal progress and path cost. The selected paths form $\mathcal{P}_q = (P_1, \ldots, P_K)$ and are jointly used for path-structured context construction.

Path-structured context construction. ECG-RAG projects selected relation paths back to grounded passages and packs the resulting evidence in path order. For a path $P = (l_0, \ldots, l_t)$, let $r(l_i)$ be the relation-node text and $\mathcal{D}_{\text{src}}(l_i)$ its grounded source passages:

$$T(P) = \text{concat}_{i=0}^{t}\big(r(l_i), \mathcal{D}_{\text{src}}(l_i)\big). \tag{8}$$

Given $\mathcal{P}_q = (P_1, \ldots, P_K)$, ECG-RAG constructs $\mathcal{C}_q = \text{concat}(T(P_1), \ldots, T(P_K), T_{\text{aux}})$, where T_{aux} contains auxiliary evidence within the context budget. The answer is generated as $\hat{a} = \text{LLM}(q, \mathcal{C}_q)$. This preserves relation-path structure while supplying textual evidence.

4 Experiments

We conduct extensive experiments to answer the following research questions: **RQ1**: How effective is ECG-RAG compared with standard RAG, graph-enhanced RAG, and hypergraph-enhanced

Table 1: **QA performance** on six RAG benchmarks, reported in EM and F1 (%). Mix and Pathology use LLM-judged scores; the other benchmarks use normalized exact match and token-level F1. Best and second-best results are highlighted.

Methods	Simple QA		Multi-Hop QA						Explanatory QA				Overall	
	PopQA		MuSiQue		2Wiki		Hotpot		Mix		Pathology		Avg.	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
GPT-4o-mini	20.7	24.5	12.4	21.7	32.2	37.4	30.8	41.4	57.5	57.5	72.1	69.4	37.6	42.0
Naive RAG														
RAG (Top-1)	41.3	51.9	21.4	31.4	28.8	35.1	41.4	55.6	69.6	68.4	76.5	74.8	46.5	52.9
RAG (Top-3)	46.6	58.6	27.4	38.4	37.9	45.1	48.5	62.7	73.9	72.2	76.8	75.5	51.8	58.8
RAG (Top-5)	48.9	60.6	28.6	40.3	41.1	48.9	50.2	65.1	74.5	73.0	77.3	75.9	53.4	60.6
Graph-enhanced RAG														
LightRAG	49.1	60.3	22.6	32.4	35.2	41.4	40.3	54.3	74.5	72.8	75.5	73.6	49.5	55.8
PathRAG	33.1	44.3	19.3	31.1	40.2	47.8	41.1	55.8	56.8	56.1	61.9	61.8	42.1	49.5
NodeRAG	50.2	62.7	27.8	40.1	40.9	49.7	50.1	64.0	75.3	73.5	76.6	75.6	53.5	60.9
HippoRAG2	45.7	59.3	30.2	**45.0**	53.7	64.2	52.5	67.6	64.7	62.9	63.6	61.8	51.7	60.1
Hypergraph-enhanced RAG														
Hyper-RAG	49.1	59.9	22.6	32.6	45.3	54.0	41.0	54.9	71.4	69.7	75.9	75.2	50.9	57.7
Cog-RAG	30.1	44.3	15.5	26.6	27.5	40.7	28.1	44.1	73.7	72.2	78.9	**78.0**	42.3	51.0
IGMiRAG	49.8	62.4	33.0	**45.0**	57.5	67.4	54.0	69.1	**76.5**	74.4	**79.2**	77.4	58.3	65.9
ECG-RAG (Ours)	**56.8**	**67.7**	**33.7**	41.9	**60.9**	**68.1**	**57.6**	**69.7**	75.9	**75.7**	77.2	76.4	**60.4**	**66.6**

RAG baselines? **RQ2**: Does ECG-RAG improve evidence-chain completeness in retrieved contexts, especially for questions requiring more evidence steps? **RQ3**: How does each key component of ECG-RAG contribute to the final QA performance and evidence-chain coverage? **RQ4**: How does ECG-RAG's evidence-chain retrieval perform under different path budgets and graph sparsity levels?

4.1 Experimental Setup

Datasets. We evaluate ECG-RAG on six QA benchmarks covering simple factual, multi-hop, and explanatory QA. PopQA (Mallen et al., 2023) represents simple factual QA, where answers are typically supported by direct evidence. MuSiQue (Trivedi et al., 2022), 2WikiMultiHopQA (Ho et al., 2020), and HotpotQA (Yang et al., 2018) cover multi-hop QA, where evidence is distributed across passages. Mix (Qian et al., 2025) and Pathology (Xiong et al., 2024) cover explanatory QA, where answers are longer and require richer evidence support. Dataset statistics and evaluated sample sizes are reported in Appendix A.

Baselines. We compare ECG-RAG with representative retrieval paradigms. The retrieval-free baseline directly prompts GPT-4o-mini without external evidence. Passage-based RAG (Lewis et al., 2020) retrieves top-k passages, instantiated as RAG (Top-1), RAG (Top-3), and RAG (Top-5). Graph-enhanced baselines include LightRAG (Guo et al., 2025), PathRAG (Chen et al., 2026), NodeRAG (Xu et al., 2025), and HippoRAG2 (Gutiérrez et al., 2025); hypergraph-enhanced baselines include Hyper-RAG (Feng et al., 2026), Cog-RAG (Hu et al., 2026), and IGMiRAG (Hou et al., 2026). All retrieval-based methods use the same answer generator. Additional baseline settings are provided in Appendix B.

Evaluation metrics. For simple and multi-hop QA, we report Exact Match (EM) and token-level F1 against reference answers. For explanatory QA, whose answers are open-ended and long-form, we use an LLM-as-a-judge protocol (Zheng et al., 2023): the evaluator receives the question, reference answer, and generated response, then assigns EM, precision, and recall; F1 is computed from judged precision and recall. The same evaluation prompt is applied to all methods, and all scores are reported as percentages. Overall scores are computed as the unweighted average over the six benchmarks.

Implementation details. We use GPT-4o-mini as the answer generator with temperature 0, and use `text-embedding-3-small` for semantic matching. All methods share the same prepro-

Table 2: Ablation study of ECG-RAG on 2WikiMultiHopQA. The left table reports end-to-end QA performance, and the right table reports evidence-chain coverage. ER denotes Evidence Recall, CCC denotes Complete Chain Coverage, and Gap denotes ER–CCC.

(a) 2WikiMultiHopQA QA

Models	EM (%)	F1 (%)
ECG-RAG	**60.9**	**68.1**
Component Ablation		
w/o Relation Exploration	58.5	66.5
w/o Goal-Guided Search	56.3	64.3
w/o Path-Structured Context	59.1	66.9

(b) 2WikiMultiHopQA Coverage

Models	ER (%)	CCC (%)	Gap
ECG-RAG	**93.6**	**85.0**	**8.6**
Component Ablation			
w/o Relation Exploration	92.3	82.6	9.7
w/o Goal-Guided Search	89.9	76.5	13.4
w/o Path-Structured Context	92.8	83.3	9.5

cessing, evaluation split, and generation setting whenever applicable. ECG-RAG builds the H^3G index offline and performs evidence-chain-guided retrieval at inference time. Hyperparameters and efficiency statistics are reported in Appendices B and C.

4.2 Main Results (RQ1)

Table 1 reports QA performance on six benchmarks. ECG-RAG achieves the strongest average performance, reaching 60.4 EM and 66.6 F1, outperforming the strongest baseline by 2.1 EM and 0.7 F1 and RAG (Top-5) by 7.0 EM and 6.0 F1.

On multi-hop QA, ECG-RAG achieves the best EM on all three benchmarks and the best F1 on 2WikiMultiHopQA and HotpotQA, showing its clearest advantage when answering requires connecting distributed evidence into a complete support chain. It also achieves the best EM and F1 on PopQA and the best F1 on Mix, while IGMiRAG and Cog-RAG perform slightly better on Pathology, indicating less consistent gains on explanatory QA.

Paired bootstrap tests confirm significant EM gains over IGMiRAG on 2WikiMultiHopQA, HotpotQA, and the three-dataset macro average (Appendix H). Under the LLaMA-3.3-70B setting, ECG-RAG remains best overall, outperforming IGMiRAG by 1.5 EM and 1.0 F1 on average (Appendix D).

4.3 Evidence-Chain Completeness Analysis (RQ2)

The preliminary analysis in Figure 1 shows that high evidence recall does not necessarily imply complete evidence-chain coverage. We conduct fine-grained evidence-chain diagnostics on 1,000 samples from 2WikiMultiHopQA (Ho et al., 2020), whose explicit supporting-fact annotations enable precise measurement of evidence-unit coverage and complete-chain recovery. ECG-RAG achieves 93.6% Evidence Recall and 85.0% Complete Chain Coverage overall, with 89.2% CCC for questions with two evidence steps and 72.2% for questions with four evidence steps.

Under the same top-k passage-level diagnostic (Appendix G), ECG-RAG progressively closes the gap as the retrieval budget increases and becomes strongest overall at top-15. At top-15, ECG-RAG and HippoRAG2 achieve nearly identical overall Evidence Recall (90.48% vs. 90.21%, $p = 0.7021$), whereas ECG-RAG achieves significantly higher Complete Chain Coverage (77.20% vs. 73.10%, $+4.10$ points, $p = 0.0081$). The difference becomes substantially larger for questions requiring four evidence steps, where ECG-RAG improves CCC from 26.56% to 56.02% ($+29.46$ points, $p < 0.0001$). This pattern is consistent with goal-guided retrieval prioritizing uncovered evidence-chain components as the evidence requirement becomes more complex.

4.4 Ablation Study (RQ3)

We ablate relation exploration, goal-guided path retrieval, and path-structured context construction by replacing them with static anchor expansion, ranked-candidate selection, and flat passage packing, respectively. Table 2 reports QA and final-context evidence-chain coverage on 2WikiMultiHopQA; MuSiQue results appear in Appendix E.

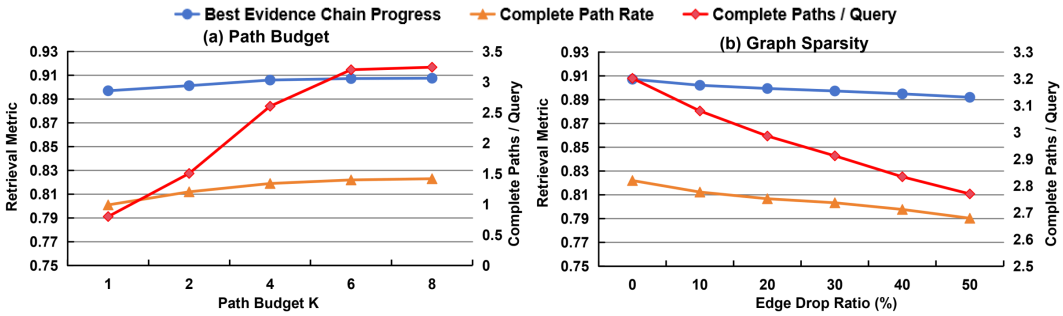

Figure 3: Sensitivity and robustness of ECG-RAG's evidence-chain retrieval under path budget and graph sparsity.

Goal-guided path retrieval. Removing goal-guided path retrieval causes the largest drop, reducing 2WikiMultiHopQA performance from 60.9 EM and 68.1 F1 to 56.3 EM and 64.3 F1. Complete Chain Coverage also drops from 85.0% to 76.5%, while the ER–CCC gap increases from 8.6 to 13.4 points. The larger degradation in both QA performance and Complete Chain Coverage supports the utility of goal-guided path assembly over the ranked-candidate selection used in this ablation.

Ant-inspired relation exploration. Removing ant-inspired exploration reduces 2WikiMultiHopQA performance to 58.5 EM and 66.5 F1, and lowers Complete Chain Coverage from 85.0% to 82.6%. Since this variant still uses activated anchors and path retrieval, the drop indicates that query-adaptive relation exploration improves the local transition graph before path assembly.

Path-structured context. Removing path-structured context lowers 2WikiMultiHopQA performance to 59.1 EM and 66.9 F1, and reduces Complete Chain Coverage to 83.3%. This variant keeps the retrieval stage but packs selected passages as flat context. Under the same context budget, flat packing can deduplicate or truncate evidence without preserving path boundaries, suggesting that path-structured context helps retain usable relation-path evidence for generation.

4.5 Evidence-Chain Retrieval under Path Budget and Graph Sparsity (RQ4)

Figure 3 evaluates path budget K and local graph sparsity using goal progress, goal-complete path rate, and complete paths per query; full numbers are in Appendix I.

Path budget. We vary K from 1 to 8 with fixed search depth and expansion budget. At $K = 1$, ECG-RAG reaches 0.897 average goal progress and an 80.1% goal-complete path rate. Increasing K to 6 mainly raises complete paths per query from 0.801 to 3.202, while the rate changes only slightly to 82.2%; further gains at $K = 8$ are marginal. Thus, a moderate path budget provides sufficient complete-chain candidates for context construction.

Graph sparsity. Inspired by the graph sparsity analysis in PathRAG (Chen et al., 2026), we remove 10% to 50% of local relation graph edges with three seeds per non-zero setting. Even at 50% edge removal, goal progress drops only from 0.907 to 0.892 and the goal-complete path rate from 82.2% to 79.0%, showing robustness to local-graph perturbation.

5 Conclusion

We study multi-hop RAG from the perspective of evidence-chain completeness, where retrieving relevant evidence fragments alone is insufficient unless they form the complete support chain required by the question. ECG-RAG moves beyond optimizing local relevance alone by explicitly targeting complete support-chain recovery and operationalizing this objective through goal-guided relation-path retrieval over an H^3G-style index. Across six QA benchmarks, diagnostics, and ablations, ECG-RAG improves QA performance and more reliably recovers complete support chains, with its clearest gains on multi-hop questions requiring distributed evidence.

AI USE STATEMENT

We used ChatGPT to improve the grammar, wording, and readability of the manuscript. All AI-assisted revisions were reviewed by the authors, who take full responsibility for the final content of the paper.

ETHICS STATEMENT

We evaluate ECG-RAG on six existing question-answering benchmarks: PopQA, MuSiQue, 2WikiMultiHopQA, HotpotQA, Mix, and Pathology. We are committed to responsible research practices, with particular attention to data privacy, transparency, and the appropriate use of benchmark data. This study does not involve research with human participants.

REPRODUCIBILITY STATEMENT

We describe the ECG-RAG framework in the Method section and specify the models and evaluation protocols in the Experimental Setup section. Dataset and index statistics are reported in Appendix A, while baseline settings and implementation details are provided in Appendix B. Evaluation metrics and prompt templates are presented in Appendices F and K, respectively.

REFERENCES

Boyu Chen, Zirui Guo, Zidan Yang, Yuluo Chen, Junze Chen, Zhenghao Liu, Chuan Shi, and Cheng Yang. Pathrag: Pruning graph-based retrieval augmented generation with relational paths. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30183–30191, 2026. doi: 10.1609/aaai.v40i36.40268. URL https://ojs.aaai.org/index.php/AAAI/article/view/40268.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 758–759, 2009.

Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41, 1996. doi: 10.1109/3477.484436.

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

Yifan Feng, Hao Hu, Shihui Ying, Xingliang Hou, Shiquan Liu, Mingyuan Yang, Junchang Li, Shaoyi Du, Nanning Zheng, Han Hu, and Yue Gao. Hyper-rag: Combating llm hallucinations using hypergraph-driven retrieval-augmented generation. *Nature Communications*, 17 (1):5778, 2026. doi: 10.1038/s41467-026-71411-1. URL https://doi.org/10.1038/s41467-026-71411-1.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1), 2023.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 10746–10761, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.568. URL https://aclanthology.org/2025.findings-emnlp.568/.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21497–21515. PMLR, 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. doi: 10.1109/TSSC.1968.300136.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Xingliang Hou, Yuyan Liu, Qi Sun, Haoxiu Wang, Hao Hu, Shaoyi Du, and Zhiqiang Tian. IGMi-RAG: Intuition-guided retrieval-augmented generation with adaptive mining of in-depth memory. *arXiv preprint arXiv:2602.07525*, 2026. URL https://arxiv.org/abs/2602.07525.

Hao Hu, Yifan Feng, Ruoxue Li, Rundong Xue, Xingliang Hou, Zhiqiang Tian, Yue Gao, and Shaoyi Du. Cog-rag: Cognitive-inspired dual-hypergraph with theme alignment retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31032–31040, 2026. doi: 10.1609/aaai.v40i37.40363. URL https://ojs.aaai.org/index.php/AAAI/article/view/40363.

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

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

Mufei Li, Siqi Miao, and Pan Li. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/11e1900e680f5fe1893a8e27362dbe2c-Abstract-Conference.html.

Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, and Yongbin Li. StructRAG: Boosting knowledge intensive reasoning of LLMs via inference-time hybrid information structurization. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5975754c7650dfee0682e06e1fec0522-Abstract-Conference.html.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/3e2aeb66481dd63a32421bf032b70384-Abstract-Conference.html.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 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)*, pp. 9802–9822, 2023.

Shiyao Peng, Qianhe Zheng, Zhuodi Hao, Zichen Tang, Rongjin Li, Qing Huang, Jiayu Huang, Jiacheng Liu, Yifan Zhu, and Haihong E. NeocorRAG: Less irrelevant information, more explicit evidence, and more effective recall via evidence chains. In *Proceedings of the ACM Web Conference 2026*, 2026. doi: 10.1145/3774904.3792093. URL https://doi.org/10.1145/3774904.3792093.

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

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM Web Conference 2025*, pp. 2366–2377, 2025. doi: 10.1145/3696410.3714805. URL https://doi.org/10.1145/3696410.3714805.

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

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

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 10014–10037, 2023.

Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. When to use graphs in RAG: A comprehensive analysis for graph retrieval-augmented generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/6c9e01d6cefbbf4cdd265032550e767f-Abstract-Conference.html.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 6233–6251, 2024.

Tianyang Xu, Haojie Zheng, Chengze Li, Haoxiang Chen, Yixin Liu, Ruoxi Chen, and Lichao Sun. Noderag: Structuring graph-based rag with heterogeneous nodes. *arXiv preprint arXiv:2504.11544*, 2025.

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

Nan Zhang, Prafulla Kumar Choubey, Alexander Fabbri, Gabriel Bernadett-Shapiro, Rui Zhang, Prasenjit Mitra, Caiming Xiong, and Chien-Sheng Wu. SiReRAG: Indexing similar and related information for multihop reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f9668d223e713943634dce9c66e8f2c1-Abstract-Conference.html.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Zhuoping Zhou, Davoud Ataee Tarzanagh, Sima Didari, Wenjun Hu, Baruch Gutow, Oxana Verkholyak, Masoud Faraki, Heng Hao, Hankyu Moon, and Seungjai Min. Query-aware flow diffusion for graph-based RAG with retrieval guarantees. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/584f32ccf76d73b85fa2053e795df697-Abstract-Conference.html.

APPENDIX

Within this supplementary material, we provide additional details and results:

- Appendix A: Dataset and Index Statistics.
- Appendix B: Baseline and Implementation Details.
- Appendix C: Efficiency Analysis.
- Appendix D: Additional Results with LLaMA-3.3-70B.
- Appendix E: Additional Ablation Results.
- Appendix F: Evaluation Metrics.
- Appendix G: Additional Evidence-Chain Coverage Results.
- Appendix H: Statistical Significance Analysis.
- Appendix I: Additional Sensitivity and Robustness Results.
- Appendix J: ECG-RAG Pipeline Example.
- Appendix K: Prompt Templates.

A DATASET AND INDEX STATISTICS

We evaluate ECG-RAG on six benchmarks that cover three retrieval and generation settings: simple factual QA, multi-hop QA, and explanatory QA. Table 3 reports both corpus statistics and H^3G index statistics for each benchmark. The corpus statistics include the number of source passages or documents, chunks, and tokens. The index statistics include the number of extracted entities, pairwise relations, and high-level relations used by ECG-RAG during retrieval.

Table 3: Corpus and H^3G index statistics for the six evaluation benchmarks.

Statistics	PopQA	MuSiQue	2Wiki	HotpotQA	Mix	Pathology
# Passages / Documents	8,676	11,656	6,119	9,811	61	1
# Chunks	1,780	2,191	1,056	2,096	980	1,099
# Tokens	1,184,847	1,362,612	684,059	1,306,249	615,355	905,760
# Entities	41,315	43,827	22,041	39,899	18,792	20,556
# Pairwise Relations	16,697	21,941	9,843	19,941	8,820	10,510
# High-level Relations	6,654	8,941	4,359	7,566	3,373	4,158
# Queries	1,000	1,000	1,000	1,000	100	100

PopQA (Mallen et al., 2023) is used for simple factual QA. Its questions are typically centered on direct factual knowledge about entities, making it useful for evaluating whether structured retrieval preserves performance on relatively shallow evidence needs. MuSiQue (Trivedi et al., 2022), 2WikiMultiHopQA (Ho et al., 2020), and HotpotQA (Yang et al., 2018) are used for multi-hop QA. These datasets require connecting evidence across multiple passages, and therefore directly test whether retrieval can recover complete support chains rather than isolated relevant facts.

Mix (Qian et al., 2025) and Pathology (Xiong et al., 2024) are used for explanatory QA. Compared with short-answer QA, these datasets require longer responses and richer contextual support. Mix contains evidence from multiple domains, whereas Pathology focuses on a specialized biomedical document. These two datasets test whether path-structured evidence remains useful when answer generation requires coherent explanations rather than short entity answers.

For PopQA, MuSiQue, 2WikiMultiHopQA, and HotpotQA, we evaluate 1,000 sampled queries from each benchmark. For Mix and Pathology, we evaluate 100 queries following their long-form explanatory QA setting. All methods are evaluated on the same sampled instances within each benchmark under the same generation settings.

B Baselines and Implementation Details

B.1 Baseline Details

We compare ECG-RAG with representative retrieval-free, passage-based, graph-enhanced, and hypergraph-enhanced RAG methods. Details of the baselines are provided below.

Retrieval-free prompting. This baseline directly prompts the answer generator with the question only, without retrieving external evidence. It measures the parametric knowledge and reasoning ability of the generator in the absence of retrieved context.

RAG (Lewis et al., 2020). This baseline follows the standard retrieve-then-generate paradigm. The corpus is segmented into text chunks, and each chunk is embedded into a vector space. At inference time, the system retrieves the top-k chunks according to semantic similarity with the input question and concatenates them as the generation context. We report RAG (Top-1), RAG (Top-3), and RAG (Top-5).

LightRAG (Guo et al., 2025). LightRAG is a graph-enhanced RAG method that builds a graph-based index over entities and relations extracted from the corpus. During retrieval, it combines low-level entity/relation retrieval with high-level graph retrieval, allowing the system to use both local factual associations and broader graph context for answer generation.

PathRAG (Chen et al., 2026). PathRAG is a graph-based RAG method that retrieves relational paths from a graph-structured index. It first identifies candidate entities and relations relevant to the query, then applies path pruning to remove weakly related or redundant paths. The remaining paths are used to construct the final context for generation.

NodeRAG (Xu et al., 2025). NodeRAG represents heterogeneous knowledge units as nodes in a graph and performs retrieval through node-level expansion. It first retrieves global candidates using lexical or semantic matching, and then expands over the graph to collect query-relevant neighboring nodes. The retrieved nodes are aggregated as context for the answer generator.

HippoRAG2 (Gutiérrez et al., 2025). HippoRAG2 is a memory-inspired graph retrieval method. It constructs an OpenIE-style graph from extracted triples and connects query-linked concepts to associated passages through graph traversal. During retrieval, it applies Personalized PageRank and related graph-search signals to identify passages that are associatively connected to the query.

Hyper-RAG (Feng et al., 2026). Hyper-RAG extends graph-enhanced retrieval by modeling both pairwise and higher-order relations with hyperedges. During retrieval, it separately retrieves entity candidates and relation candidates, and then propagates activation through the hypergraph to identify multi-entity associations relevant to the query.

Cog-RAG (Hu et al., 2026). Cog-RAG is a hypergraph-enhanced RAG method that models both topic-level and entity-level associations. It constructs topic and entity hypergraphs from the corpus, retrieves relevant topic or entity structures according to the query, and uses the resulting hypergraph context to support answer generation.

IGMiRAG (Hou et al., 2026). IGMiRAG constructs a Hierarchical Heterogeneous Hypergraph (H^3G) to organize atomic entities, pairwise relations, high-level relations, and grounding links to source passages. At inference time, it parses the query into retrieval preferences and performs intuition-guided matching over the H^3G to retrieve task-relevant knowledge from different layers.

B.2 Implementation Details

Generation settings. We use GPT-4o-mini as the answer generator for all methods and set the temperature to 0 to reduce generation variance. All methods are evaluated on the same sampled queries within each benchmark. For retrieval-based methods, the retrieved evidence is provided to the same answer generator whenever applicable. For retrieval-free prompting, only the question is provided.

Embedding and semantic matching. For semantic retrieval, query-evidence matching, and relation matching in ECG-RAG, we use `text-embedding-3-small`. All corpus chunks and extracted retrieval units are embedded with the same embedding model when semantic matching is

Table 4: Implementation details and hyperparameters for baseline methods.

Baselines	RAG	LightRAG	PathRAG	NodeRAG	HippoRAG2	Hyper-RAG	Cog-RAG	IGMiRAG
Chunk Token Size	1,200	1,200	1,200	1,024	–	1,200	1,200	780 / 1,024
Chunk Overlap Token Size	100	100	100	–	–	100	100	–
Query Mode	–	hybrid	hybrid	–	–	hyper	cog	–
Top-k for Retrieve	1, 3, 5	60	40	10	5	60	60	adaptive
Max Token for Local Context	–	1,000	2,500	–	–	300	300	adaptive
Max Token for Global Context	–	2,400	2,100	–	–	1,600	1,600	adaptive
Max Token for Text Context	–	2,500	2,100	–	–	1,600	1,600	adaptive
Synonym Threshold	–	–	–	–	0.8	–	–	–
Damping Factor α of PPR	–	–	–	0.5	0.5	–	–	–
Max Iterations of PPR	–	–	–	2	–	–	–	–
Linking Top-k	–	–	–	–	5	–	–	–
Retrieval Top-k	–	–	–	–	200	–	–	–
Passage Node Weight	–	–	–	–	0.05	–	–	–
QA Top-k	–	–	–	–	5	–	–	–
Answer Generator	GPT-4o-mini							
Embedding Model	`text-embedding-3-small`							
Temperature	0							

required. Lexical matching and exact entity matching are used as additional retrieval signals when supported by the corresponding method.

Baseline hyperparameters. For all baseline methods, we follow the default hyperparameters and recommended operational modes specified in their official implementations whenever available. Following IGMiRAG (Hou et al., 2026), LightRAG, PathRAG, NodeRAG, Hyper-RAG, and Cog-RAG use their recommended retrieval modes and context budgets. For HippoRAG2 (Gutiérrez et al., 2025), we follow its recommended graph construction and Personalized PageRank retrieval settings while using the same embedding model and answer generator as the other methods for a consistent evaluation setup. The main implementation settings are summarized in Table 4.

ECG-RAG configuration. ECG-RAG builds the H^3G index offline and performs evidence-chain-guided retrieval at inference time. The index contains atomic entities, pairwise relations, high-level relations, and grounding links to source passages. Unless otherwise specified, ECG-RAG uses GPT-4o-mini as the answer generator with temperature 0 and `text-embedding-3-small` for semantic matching. It retrieves the top-60 query-relevant graph nodes as the initial anchor candidate pool and selects up to 16 start relation anchors by default, runs ant exploration with three ants per start relation and maximum depth 4, and performs A*-style path search with maximum depth 4 and expansion budget 300. ECG-RAG retains $K = 6$ relation paths for path-structured context construction; this default is selected based on the path-budget sensitivity analysis in Appendix I, where performance largely saturates after $K = 6$. The ablation variants replace ant exploration with static expansion, goal-guided path search with ranked-candidate selection, and path-structured packing with flat passage packing.

Context construction. For simple factual QA and multi-hop QA, the final context contains the selected path-structured evidence and a limited amount of auxiliary evidence within the context budget. For explanatory QA, ECG-RAG uses the same retrieval pipeline but allows longer generated responses, since the answers require richer contextual support. Dataset-specific context budgets are used only when required by different answer formats.

Evaluation settings. For simple factual QA and multi-hop QA, we report Exact Match (EM) and token-level F1 against the reference answers. For explanatory QA, where answers are open-ended and long-form, we use the same LLM-as-a-judge protocol for all methods. The metric definitions and evidence-chain diagnostics are described in Appendix F.

C Efficiency Analysis

Answer-context tokens. We first measure the final answer-context length supplied to the generator. As a reference, we compare this quantity with the generation-stage Avg. Tokens reported by IGMiRAG (Hou et al., 2026) for several structure-enhanced RAG methods. Because the account-

ing definitions are not identical across implementations, these values are intended as a context-size reference rather than a strict API-cost ranking.

Table 5: Reference comparison of generation-stage token usage (k tokens per query). Baseline values follow the Avg. Tokens reported by IGMiRAG, while ECG-RAG reports the final answer-context length supplied to the generator. Avg. denotes the unweighted average over the six benchmarks.

Method	PopQA	MuSiQue	2Wiki	HotpotQA	Mix	Pathology	Avg.
LightRAG	14.6	15.0	13.4	13.0	13.9	15.7	14.3
PathRAG	11.6	11.9	11.4	12.2	12.7	13.8	12.2
NodeRAG	4.7	5.8	6.5	5.0	5.1	5.5	5.4
Hyper-RAG	12.3	13.2	12.9	12.7	12.2	14.5	13.0
Cog-RAG	29.3	31.5	30.4	29.5	27.7	33.8	30.3
IGMiRAG	3.1	5.8	4.7	5.0	8.7	11.0	6.4
ECG-RAG	7.42	9.21	8.98	8.65	8.20	14.30	9.46

ECG-RAG uses 9.46k final answer-context tokens per query on average across the six benchmarks. Under this reference comparison, its context size is lower than the reported averages of LightRAG, PathRAG, Hyper-RAG, and Cog-RAG, while higher than NodeRAG and IGMiRAG. This indicates that ECG-RAG's gains in evidence-chain completeness and QA performance are not obtained by expanding the generation context to unusually large sizes.

Goal-analysis overhead. We measure the API usage of the evidence-chain goal analyzer by rerunning only this component on the same evaluated queries under the same analyzer configuration. The analyzer consumes 1.38k tokens per query on average, with approximately 1.19k input tokens and 0.20k output tokens. Dataset-level averages range from 1.29k to 1.49k tokens per query. We report this query-side analysis cost separately from answer-context length because the two quantities measure different token scopes and should not be summed.

For additional context, IGMiRAG reports 1.37k additional query-processing tokens per query for its retrieval-strategy parser (Hou et al., 2026), comparable in magnitude to ECG-RAG's 1.38k-token goal-analysis overhead. We treat this only as a contextual reference because the detailed accounting protocols are not identical.

Query runtime. We also report average wall-clock query runtime from our evaluation runs on PopQA and the three multi-hop QA benchmarks. These measurements cover ECG-RAG's online query processing under our evaluation setup. Since absolute latency depends on hardware, API serving conditions, caching, and concurrency, we use them only to characterize our runs rather than to make cross-paper latency comparisons.

Table 6: Average wall-clock query runtime of ECG-RAG under our evaluation setup.

Task	Dataset	Time / Query (s)
Simple QA	PopQA	0.84
Multi-Hop QA	2Wiki	0.92
	HotpotQA	1.20
	MuSiQue	2.73
	Multi-Hop Avg.	**1.62**

Across the three multi-hop benchmarks, ECG-RAG requires 1.62 seconds per query on average. Runtime is below 1.3 seconds on 2WikiMultiHopQA and HotpotQA, and increases to 2.73 seconds on MuSiQue. This shows that the evidence-chain-guided retrieval procedure remains within low single-digit seconds per query in our primary multi-hop evaluation setting.

Table 7: Additional QA results using LLaMA-3.3-70B as the indexing and answer generation model.

Methods	Simple QA		Multi-Hop QA						Explanatory QA				Overall	
	PopQA		MuSiQue		2Wiki		Hotpot		Mix		Pathology		Avg.	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
LLaMA-3.3-70B	33.8	37.0	17.7	26.2	38.9	45.7	37.6	48.0	61.9	60.0	72.4	70.6	43.7	47.9
Naive RAG														
RAG (Top-1)	39.8	51.0	21.4	28.9	32.7	36.5	45.3	57.6	64.9	65.7	74.1	73.1	46.4	52.1
RAG (Top-3)	45.6	58.3	29.0	38.5	43.1	48.3	54.3	65.9	71.7	70.0	74.8	73.2	53.1	59.0
RAG (Top-5)	47.1	59.9	32.2	42.8	45.5	52.0	55.7	67.9	72.6	71.9	75.3	**74.3**	54.7	61.5
Graph-enhanced RAG														
LightRAG	48.7	58.7	29.9	39.7	42.9	48.3	51.1	64.0	73.4	71.1	74.5	73.0	53.4	59.1
PathRAG	45.9	59.7	26.9	36.7	44.9	49.6	51.3	64.6	56.0	57.0	65.2	62.3	48.4	55.0
NodeRAG	47.5	60.4	31.8	42.2	49.8	55.6	53.4	67.4	74.3	72.2	75.3	73.6	55.4	61.9
HippoRAG2	44.5	58.6	35.0	49.2	59.8	68.1	57.4	71.3	62.6	61.2	61.3	59.7	53.4	61.4
Hypergraph-enhanced RAG														
Hyper-RAG	47.1	60.2	27.3	37.1	56.6	63.3	52.7	66.0	66.5	65.9	70.9	67.7	53.5	60.0
Cog-RAG	36.8	43.8	26.8	36.6	39.1	44.9	50.8	63.3	61.4	59.0	66.7	65.1	46.9	52.1
IGMiRAG	47.7	60.9	40.3	**51.3**	66.3	73.4	60.9	73.8	**74.4**	73.0	**75.7**	73.9	60.9	67.7
ECG-RAG (Ours)	**54.3**	**66.0**	**40.5**	49.4	**67.6**	**74.3**	**62.6**	**74.4**	73.9	**74.1**	75.3	73.9	**62.4**	**68.7**

D Additional Results with LLaMA-3.3-70B

To examine whether the performance trend depends on the LLM configuration, we report additional results using LLaMA-3.3-70B for indexing and answer generation. As shown in Table 7, ECG-RAG achieves the strongest average performance under this setting, reaching 62.4 EM and 68.7 F1. Compared with the strongest baseline, IGMiRAG, ECG-RAG improves the average score by 1.5 EM and 1.0 F1. ECG-RAG also achieves the highest EM on all three multi-hop benchmarks, although its MuSiQue F1 remains below IGMiRAG. These results support ECG-RAG's effectiveness under an alternative indexing and generation model configuration.

E Additional Ablation Results

Table 8 reports additional end-to-end ablation results on MuSiQue. The trend is consistent with the 2WikiMultiHopQA ablation results in Table 2: removing goal-guided path retrieval causes the largest performance drop, while relation exploration and path-structured context provide additional gains.

Table 8: Additional ablation results on MuSiQue.

Models	EM (%)	F1 (%)
ECG-RAG	**33.7**	**41.9**
Component Ablation		
w/o Relation Exploration	30.7	39.3
w/o Goal-Guided Search	28.5	35.8
w/o Path-Structured Context	32.3	40.2

F Evaluation Metrics

QA metrics. For simple factual QA and multi-hop QA, we report Exact Match (EM) and token-level F1 against the reference answers. EM measures whether the normalized prediction exactly matches the normalized reference answer, while F1 measures token overlap between the prediction and reference. For explanatory QA, where answers are open-ended and long-form, we use an LLM-as-a-judge protocol. Given the question, reference answer, and model response, the evaluator assigns EM, precision, and recall scores; F1 is computed from the judged precision and recall. The same evaluation prompt is used for all methods, with GPT-4o-mini as the evaluator and temperature set to 0. Overall scores are computed as the unweighted average over the six benchmarks.

Evidence-chain retrieval metrics. We use evidence-chain metrics to evaluate whether the retrieved context contains complete support chains rather than only isolated supporting facts. Evidence Recall measures the fraction of gold supporting evidence units covered by the retrieved evidence. Complete Chain Coverage is a binary question-level metric that equals 1 if all gold evidence units in the support chain are covered by the retrieved context, and 0 otherwise. For analyses over ranked retrieval lists, we report Evidence Recall@k and Complete Chain Coverage@k for $k \in \{5, 10, 15\}$. For ECG-RAG diagnostics, we additionally compute Evidence Recall and Complete Chain Coverage over the final generation context supplied to the answer generator.

For path-level retrieval analysis, we report average goal progress, the goal-complete path rate, and the average number of goal-complete paths. Average goal progress measures the fraction of evidence-chain steps covered by the best retrieved path for each question and ranges from 0 to 1. The goal-complete path rate is the fraction of questions for which at least one retrieved path covers all factual evidence-chain steps. The average number of goal-complete paths measures how many retrieved paths per question fully cover these factual steps.

G Additional Evidence-Chain Coverage Results

Coverage protocol. For 2WikiMultiHopQA, a gold evidence unit is a supporting fact specified by a Wikipedia title and sentence id. In the top-k diagnostic, coverage is computed at the canonical passage level. For BM25, Contriever, GTR, and HippoRAG2, the top-k list consists of the retrieved canonical passages. For ECG-RAG, relation paths are projected back to their grounded source passages; duplicate passages are removed by passage id while preserving the path-ranking order, and the first k passages are used for evaluation. Thus, the ECG-RAG top-k diagnostic is computed over the same passage-level evidence units as the baseline retrievers.

A gold evidence unit is counted as covered if one of the top-k passages has the same Wikipedia title and contains the corresponding gold sentence after text normalization. In the final-context diagnostic, coverage is computed over the normalized final generation context visible to the answer generator; a gold evidence unit is counted only when the corresponding gold sentence appears in that context. Evidence Recall is the mean fraction of covered gold evidence units, and Complete Chain Coverage is the fraction of questions for which all gold evidence units are covered. For grouped diagnostics, the evidence-step count is the number of annotated gold supporting evidence units for each question.

ECG-RAG diagnostics. Table 9 reports ECG-RAG's evidence-chain coverage on 1,000 samples from 2WikiMultiHopQA. The metrics are computed over the final generation context supplied to the answer generator. We report the two main groups in the table; Overall is computed over all 1,000 questions, including the remaining ten questions with other evidence-step counts. ECG-RAG maintains high evidence recall on the two main evidence-step groups, while complete-chain coverage decreases for questions requiring more evidence steps.

Table 9: Evidence-chain diagnostics of ECG-RAG on 2WikiMultiHopQA, grouped by the number of evidence steps. Metrics are computed over the final generation context.

Evidence Steps	Samples	Evidence Recall (%)	Complete Chain Coverage (%)
2	749	94.5	89.2
4	241	91.0	72.2
Overall	1,000	93.6	85.0

Top-k projected-passage coverage. Table 10 reports top-k evidence-chain coverage on 2WikiMultiHopQA. For BM25, Contriever, GTR, and HippoRAG2, top-k denotes the retrieved canonical passages. For ECG-RAG, top-k is measured over projected atomic passages before final context packing. This follows the same passage-level diagnostic used in the preliminary analysis.

At small retrieval budgets, HippoRAG2 achieves stronger overall coverage. As the budget increases, however, ECG-RAG progressively closes the gap and achieves the highest overall CCC at top-15. Notably, the two methods have nearly identical overall ER@15, while ECG-RAG achieves significantly higher CCC@15. The advantage is much larger on questions requiring four evidence steps,

Table 10: Top-k evidence-chain coverage on 2WikiMultiHopQA. ER and CCC are computed over annotated supporting evidence units, and results are grouped by the number of evidence steps. For ECG-RAG, top-k is measured over projected atomic passages.

Method	Top-k	Overall		2 Steps		4 Steps	
		ER (%)	CCC (%)	ER (%)	CCC (%)	ER (%)	CCC (%)
BM25	5	65.62	32.70	69.83	41.52	52.80	5.81
BM25	10	70.44	39.80	75.23	50.87	56.02	6.22
BM25	15	72.14	42.70	77.10	54.47	57.05	6.64
Contriever	5	61.11	35.00	71.03	44.73	30.81	4.98
Contriever	10	66.88	41.90	76.10	53.81	38.59	5.39
Contriever	15	70.46	46.20	79.31	59.55	43.46	5.39
GTR	5	67.72	35.70	72.76	45.79	52.18	4.56
GTR	10	70.84	40.80	76.17	52.34	54.56	5.39
GTR	15	72.11	43.20	77.57	55.14	55.39	6.64
HippoRAG2	5	85.46	65.30	90.19	80.51	71.58	19.50
HippoRAG2	10	89.13	71.40	93.26	86.65	76.87	25.31
HippoRAG2	15	90.21	73.10	94.19	88.38	78.32	26.56
ECG-RAG	5	79.49	54.40	83.38	67.16	67.95	16.18
ECG-RAG	10	87.61	70.80	89.59	79.31	81.74	45.23
ECG-RAG	15	90.48	77.20	91.99	83.98	85.68	56.02

indicating that ECG-RAG is particularly effective at recovering complete support chains when evidence requirements become more complex.

H Statistical Significance Analysis

We use paired bootstrap resampling with 20,000 resamples to assess whether the observed differences are stable across aligned evaluation instances. For each resample, question indices are sampled with replacement and the same sampled indices are applied to both compared methods. We report the observed performance difference, the percentile-based 95% confidence interval, and the corresponding p-value. For QA performance, we compare ECG-RAG with IGMiRAG on the three multi-hop benchmarks. The macro-average result is computed by independently resampling each dataset and averaging the three dataset-level differences within each bootstrap replicate. For retrieval diagnostics, we compare ECG-RAG with HippoRAG2 on 2WikiMultiHopQA at top-15 under the same supporting-evidence-unit coverage protocol.

Table 11: Paired bootstrap significance for multi-hop QA performance against IGMiRAG. Differences are reported as ECG-RAG minus IGMiRAG.

Dataset	ΔEM	95% CI	p	ΔF1	95% CI	p
MuSiQue	+0.70	$[-2.30, +3.60]$	0.6760	-2.99	$[-5.85, -0.21]$	0.0350
2WikiMultiHopQA	+3.40	$[+0.50, +6.30]$	0.0247	+0.67	$[-2.08, +3.45]$	0.6437
HotpotQA	+3.60	$[+0.80, +6.40]$	0.0161	+0.67	$[-1.74, +3.03]$	0.5900
Macro Avg.	+2.57	$[+0.87, +4.27]$	0.0030	-0.55	$[-2.10, +0.99]$	0.4871

I Additional Sensitivity and Robustness Results

Table 13 reports the complete results for varying the path budget K. In ECG-RAG, K is applied inside path search rather than only as a post-retrieval cutoff: it controls the number of terminal candidate paths retained for context construction. Target-reaching paths are ranked ahead of incomplete

Table 12: Paired bootstrap significance for top-15 evidence-chain coverage on 2WikiMultiHopQA against HippoRAG2. Differences are reported as ECG-RAG minus HippoRAG2.

Subset	Metric	Δ	95% CI	p
Overall	ER@15	+0.27	$[-1.05, +1.58]$	0.7021
Overall	CCC@15	+4.10	$[+1.00, +7.20]$	0.0081
2 Steps	ER@15	-2.20	$[-3.67, -0.73]$	0.0039
2 Steps	CCC@15	-4.41	$[-7.34, -1.47]$	0.0039
4 Steps	ER@15	+7.37	$[+4.67, +9.96]$	< 0.0001
4 Steps	CCC@15	+29.46	$[+21.58, +36.93]$	< 0.0001

Table 13: Path budget sensitivity on 2WikiMultiHopQA.

K	Best Goal Progress	Complete-Path Rate (%)	Complete Paths / Query
1	0.897	80.1	0.801
2	0.901	81.2	1.506
4	0.906	81.9	2.605
6	0.907	82.2	3.202
8	0.908	82.3	3.243

candidates; if fewer than K such paths are available, remaining slots are filled by partial paths with higher goal progress and lower path cost. The maximum search depth and expansion budget are fixed across all K values.

Best Goal Progress measures the fraction of factual evidence-chain steps covered by the best retrieved path, Complete-Path Rate measures the percentage of questions with at least one path that covers all factual steps, and Complete Paths / Query reports the average number of such paths per question. ECG-RAG already retrieves complete paths for most questions with $K = 1$, while larger K mainly provides more complete path candidates for context construction.

Graph sparsity robustness. Table 14 reports the complete results for randomly removing edges from the local relation graph. For each non-zero edge drop ratio, we repeat the experiment with three random seeds and report mean $\pm$ standard deviation. ECG-RAG degrades smoothly as the local relation graph becomes sparser, indicating that goal-guided path retrieval can exploit alternative relation routes under structural perturbation.

Table 14: Graph sparsity robustness on 2WikiMultiHopQA. Edge drop ratios above zero are averaged over three random seeds.

Edge Drop Ratio	Best Goal Progress	Complete-Path Rate (%)	Complete Paths / Query
0%	0.907	82.2	3.202
10%	0.902 ± 0.001	81.2 ± 0.1	3.080 ± 0.007
20%	0.899 ± 0.002	80.7 ± 0.1	2.986 ± 0.014
30%	0.897 ± 0.003	80.3 ± 0.1	2.912 ± 0.010
40%	0.895 ± 0.002	79.8 ± 0.1	2.834 ± 0.005
50%	0.892 ± 0.003	79.0 ± 0.2	2.770 ± 0.016

J ECG-RAG Pipeline Example

We provide a qualitative example from 2WikiMultiHopQA to illustrate the complete ECG-RAG workflow. For readability, we show shortened evidence excerpts and only the relation nodes directly involved in the evidence chain. The question requires two parallel evidence branches: identifying the director of each film, retrieving each director's birth date, and comparing the two dates.

Indexing: extracting grounded relation evidence from source passages

Source chunk 1
Title: *Golmaal Again*
Evidence excerpt: *Golmaal Again* is a 2017 Indian Hindi-language supernatural action comedy film co-produced and directed by Rohit Shetty.
Entities:

- *Golmaal Again*
- Rohit Shetty

Pairwise relation:

- ⟨*Golmaal Again*, directed by, Rohit Shetty⟩

Grounding: This relation node is grounded to the source passage *Golmaal Again*.

Source chunk 2
Title: Rohit Shetty
Evidence excerpt: Rohit Shetty was born on 14 March 1973 and is an Indian film director and producer.
Entities:

- Rohit Shetty
- 14 March 1973

Pairwise relation:

- ⟨Rohit Shetty, date of birth, 14 March 1973⟩

Grounding: This relation node is grounded to the source passage Rohit Shetty.

Source chunk 3
Title: *Ma femme s'appelle Maurice*
Evidence excerpt: *Ma femme s'appelle Maurice* is a 2002 French comedy film directed by Jean-Marie Poiré.
Entities:

- *Ma femme s'appelle Maurice*
- Jean-Marie Poiré

Pairwise relation:

- ⟨*Ma femme s'appelle Maurice*, directed by, Jean-Marie Poiré⟩

Grounding: This relation node is grounded to the source passage *Ma femme s'appelle Maurice*.

Source chunk 4
Title: Jean-Marie Poiré
Evidence excerpt: Jean-Marie Poiré was born on 10 July 1945 and is a French film director and screenwriter.
Entities:

- Jean-Marie Poiré
- 10 July 1945

Pairwise relation:

- ⟨Jean-Marie Poiré, date of birth, 10 July 1945⟩

Grounding: This relation node is grounded to the source passage Jean-Marie Poiré.

Query analysis: deriving the evidence-chain goal

Question: Which film whose director is younger, *Golmaal Again* or *Ma Femme S'Appelle Maurice*?

Activated query anchors:

- *Golmaal Again*
- *Ma Femme S'Appelle Maurice*

Answer target: the film whose director is younger.

Factual evidence-chain goal:

1. Ground the director of *Golmaal Again*.
2. Ground the birth date of that director.
3. Ground the director of *Ma femme s'appelle Maurice*.
4. Ground the birth date of that director.

Answer derivation: Compare the two grounded birth dates and return the film whose director is younger.
Key point: The retrieval target is not any single relevant passage, but a complete support chain containing both film-director links and both director birth-date facts.

Retrieval: relation exploration and goal-guided path search

Local relation exploration.
Starting from the activated anchors, ECG-RAG explores the pairwise-relation layer and constructs a compact local relation graph for the current query. In this example, the local graph contains relation nodes grounded to the candidate films, their directors, and the directors' biographical passages.

Recovered evidence paths:

- **Path 1:** *Golmaal Again* → directed by → Rohit Shetty → date of birth → 14 March 1973.
- **Path 2:** *Ma femme s'appelle Maurice* → directed by → Jean-Marie Poiré → date of birth → 10 July 1945.

Goal coverage:

- Step 1 is covered by ⟨*Golmaal Again*, directed by, Rohit Shetty⟩.
- Step 2 is covered by ⟨Rohit Shetty, date of birth, 14 March 1973⟩.
- Step 3 is covered by ⟨*Ma femme s'appelle Maurice*, directed by, Jean-Marie Poiré⟩.
- Step 4 is covered by ⟨Jean-Marie Poiré, date of birth, 10 July 1945⟩.

The retrieved paths therefore provide the complete evidence needed for the final comparison.

Generation: path-structured context and final answer

Packed path-structured context:

Path 1 evidence.

- *Golmaal Again* was directed by Rohit Shetty.
- Rohit Shetty was born on 14 March 1973.

Path 2 evidence.

- *Ma femme s'appelle Maurice* was directed by Jean-Marie Poiré.
- Jean-Marie Poiré was born on 10 July 1945.

Answer reasoning.
Rohit Shetty was born on 14 March 1973, while Jean-Marie Poiré was born on 10 July 1945. Since Rohit Shetty was born later, he is the younger director.

Prediction: *Golmaal Again*.

K PROMPT TEMPLATES

We provide normalized prompt templates used in ECG-RAG. The templates preserve the task definitions, input fields, output formats, and main constraints used in our implementation. For readability, we omit only dataset-specific examples and low-level formatting guards that do not change the task semantics.

K.1 PROMPTS FOR EVIDENCE-CHAIN GOAL ANALYSIS

Evidence-Chain Goal Analysis

Prompt:
-Goal-
You are a query analysis assistant for evidence-chain-guided retrieval. Given a question, decompose it into a minimal evidence-chain goal that specifies what evidence must be grounded before the answer can be produced. Do not answer the question. Do not guess missing bridge entities, dates, locations, or other facts. Only describe the evidence operations that should be grounded from the corpus.

-Input-
Question: {`question`}

-Output-
Return a valid JSON object with the following fields:

- `anchors`: explicit entities, works, places, organizations, events, or noun phrases copied from the question. These anchors are used as retrieval starting points.
- `steps`: ordered evidence operations. Each step contains `id`, `kind`, `source`, `relation`, `target`, and `depends_on`.
- `answer`: the final answer specification, including `source_step`, answer `type`, a short `description`, and explicit `candidates` when the question asks to select among alternatives.

-Requirements-

1. Preserve every relation needed to connect the explicit anchors in the question to the final answer.
2. Do not skip intermediate relations stated or implied by the question.
3. Represent unknown bridge entities or values as unresolved slots rather than guessing their values.
4. Use `fact` for steps that require retrieving new evidence from the corpus, and use `derive` only for operations that combine, compare, or select from previously grounded facts.
5. Each non-anchor source must be produced by an earlier step or refer to an unresolved slot introduced by an earlier dependency.
6. For comparison questions, retrieve the relevant attribute for each candidate before adding the comparison step.
7. For yes/no questions, retrieve the evidence needed to verify the relation before deriving the final answer.
8. Keep the plan minimal and use no more than six evidence steps.
9. The final answer type should describe the object to be returned, not an intermediate attribute used only for selection.

K.2 Prompts for Answer Generation

Short-Answer Generation

Prompt:
-Goal-
You are a question-answering assistant. Answer the original question using only the provided evidence. The evidence may include retrieved passages, relation descriptions, and path-structured evidence chains. Analyze the evidence and return the shortest answer span supported by the context.

-Input-
Evidence: {`retrieved_context`}
Question: {`question`}

-Requirements-

1. Use only facts supported by the provided evidence.
2. Retrieval metadata, query-analysis results, answer-target constraints, and bridge hints may guide reasoning, but should not be treated as factual evidence unless supported by retrieved text or relation evidence.
3. For multi-hop questions, identify the bridge entity or intermediate fact before deriving the final answer.
4. If the question asks to choose among candidates, return the selected candidate rather than the intermediate date, number, or attribute used for comparison.
5. Preserve the wording and granularity of the answer when the evidence provides an exact answer span.
6. If the provided evidence is insufficient, output `Unknown`.
7. Do not add explanations after the final answer.

-Output-
Start with `Thought:` and briefly identify the evidence chain used to answer the question. Conclude with `Answer:` followed only by the final answer.

Explanatory Answer Generation

Prompt:
-Goal-
Use the provided evidence to answer the question comprehensively and accurately. The answer should be coherent, focused on the question, and supported by the retrieved context.

-Input-
Evidence: {`retrieved_context`}
Question: {`question`}

-Requirements-

1. Use all relevant evidence from the context.
2. Write a coherent explanatory answer rather than a list of disconnected facts.
3. Organize the response around the question.
4. When supported by the evidence, cover the core concept, underlying mechanism, important distinctions, practical implications, and domain-specific details.
5. Do not introduce unsupported facts or external knowledge.
6. Keep the answer focused and avoid unnecessary background information.

-Output-
Provide a complete long-form answer supported by the retrieved evidence.

K.3 PROMPTS FOR EVALUATION

LLM-as-a-Judge Evaluation

Prompt:
-Goal-
Given a question, a reference answer, and a model response, evaluate whether the model response is semantically correct, sufficiently complete, and faithful to the reference answer.

-Input-
Question: {`question`}
Reference Answer: {`reference_answer`}
Model Response: {`model_response`}

-Scoring Dimensions-

1. `Exact Match`: whether the model response is semantically equivalent to the reference answer and contains the required answer information.
2. `Recall`: how much information from the reference answer is covered by the model response.
3. `Precision`: how much information in the model response is factually consistent with the reference answer.

-Requirements-
Assign each dimension a score from 0 to 100. Penalize missing key information, unsupported claims, overly vague answers, and content inconsistent with the reference. Provide a short explanation for the judgment.

-Output-
Return a JSON object with fields `exact_match`, `recall`, `precision`, and `explanation`.