Efficient Path-Coherent Retrieval via Branch Pruning in KG-RAG

Christopher Buratti[1,†], Michele Marchetti[1,†], Federica Parlapiano[1,†], Giulia Quaglieri[1,†], Davide Traini[1,2,†], Domenico Ursino[1,†] and Luca Virgili[1,*,†]

[1]DII, Polytechnic University of Marche, Ancona, Italy
[2]CHIMOMO, University of Modena and Reggio Emilia, Modena, Italy

Abstract

Knowledge Graph-based Retrieval-Augmented Generation (KG-RAG) systems typically retrieve facts independently, losing the relational structure that multi-hop reasoning requires. Restoring this structure through exhaustive graph exploration is computationally prohibitive on large graphs, creating a fundamental trade-off between retrieval quality and efficiency. To address this path coherence gap, we propose Path-Coherent Retrieval-Augmented Generation (PACO-RAG), a context-aware retrieval method that retrieves compact, path-coherent sets of triples organized into explicit reasoning chains via branch pruning. Starting from the most query-relevant nodes in each KG community, PACO-RAG explores the graph hop-by-hop, permanently discarding unpromising branches and their entire downstream subtrees to strictly bound the search space. Each candidate triple is evaluated through a composite score balancing direct query relevance, contextual non-redundancy, and relation pertinence. Experiments on two fact-checking benchmarks (MOCHEG and HoVer) show that PACO-RAG outperforms existing graph-based approaches in Accuracy and F1-score, while reducing context length by up to 96.6% and execution time by up to 98.4% compared to the most competitive baselines.

Keywords

Retrieval-Augmented Generation, Knowledge Graph, Path-Coherent Retrieval, Graph Pruning, Multi-hop Reasoning

1. Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language processing tasks, but they remain prone to factual inaccuracies and limitations in multi-hop reasoning [1]. Retrieval-Augmented Generation (RAG) has emerged as a widespread strategy to mitigate these shortcomings by anchoring the generative process to evidence retrieved from external sources [2]. Knowledge Graphs (KGs), which encode entities and semantic relationships in an explicit graph structure, represent a natural substrate for this task and have attracted growing interest as a basis for retrieval [3, 4, 5, 6].

Despite the structural richness of KGs, most existing KG-RAG frameworks do not fully exploit it during retrieval. The context provided to the LLM is typically assembled as a flat set of triples, that is, entity–relationship–entity facts selected independently via similarity or heuristic traversal, with no guarantee that they form a coherent chain of reasoning. When triples are evaluated independently, nothing prevents the retrieved set from spanning disconnected regions of the graph, forcing the LLM to bridge inferential gaps that the context simply does not support. We refer to this limitation as the

1st Workshop on Graphs Across AI: From Structural Reasoning to Foundation Models, held in conjunction with the IEEE World Congress on Computational Intelligence (WCCI 2026), June 21–26, 2026, Maastricht, The Netherlands
*Corresponding author.
†These authors contributed equally.
c.buratti@pm.univpm.it (C. Buratti); michele.marchetti@univpm.it (M. Marchetti); f.parlapiano@pm.univpm.it (F. Parlapiano); g.quaglieri@pm.univpm.it (G. Quaglieri); davide.traini@unimore.it (D. Traini); d.ursino@univpm.it (D. Ursino); luca.virgili@univpm.it (L. Virgili)
0009-0007-1675-0155 (C. Buratti); 0000-0003-3692-3600 (M. Marchetti); 0009-0009-5475-8101 (F. Parlapiano); 0009-0008-2650-5038 (G. Quaglieri); 0009-0007-3098-9349 (D. Traini); 0000-0003-1360-8499 (D. Ursino); 0000-0003-1509-783X (L. Virgili)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-08

CEUR-WS.org/Vol-4319/paper2.pdf

path coherence gap, a condition in which the LLM receives the correct entities but not the relational structure connecting them, which is exactly what multi-hop reasoning requires.

Bridging this gap requires organizing the retrieved context into path-coherent reasoning chains rather than a flat list of isolated facts. However, doing so via exhaustive graph expansion is computationally prohibitive on large KGs, as the number of candidate triples grows exponentially with the number of hops. Retrieval efficiency is therefore not a secondary concern, but a necessary condition for any practical solution.

To this end, we propose Path-Coherent Retrieval-Augmented Generation (PACO-RAG), a context-aware branch pruning approach based on hierarchical Breadth-First Search (BFS) that retrieves compact, path-coherent sets of triples organized into explicit reasoning chains, starting from the most query-relevant node in each KG community. PACO-RAG explores the graph one hop at a time, evaluating each candidate triple against three complementary signals that jointly capture direct relevance to the query, semantic coherence with the already selected triples, and relevance of the traversed relation. The key efficiency gain stems from the pruning strategy: once a branch is discarded, its entire downstream subtree is permanently excluded from further exploration without any backtracking, keeping the search space strictly bounded regardless of graph size.

The main contributions of this work are:

- A context-aware branch pruning framework that retrieves path-coherent reasoning chains via BFS, achieving efficient retrieval through branch elimination;
- A multi-signal scoring function that jointly balances query relevance, context coherence, and relation pertinence to guide pruning decisions at each hop;
- An experimental evaluation on two fact-checking benchmarks demonstrating that PACO-RAG consistently outperforms existing KG-RAG methods in accuracy and F1-score, while reducing context length by up to 96.6% and execution time by up to 98.4% compared to the most competitive baselines.

The remainder of this paper is organized as follows. Section 2 reviews related work, Section 3 presents the proposed approach, Section 4 describes the experiments, and Section 5 concludes the paper and outlines directions for future work.

2. Related Work

RAG with flat retrieval. Standard RAG systems represent documents or text segments as dense vectors and retrieve the top-k passages most similar to the query [7]. Although effective for single-hop queries, this paradigm does not model the relationships between retrieved units, often resulting in fragmented or redundant context when evidence needs to be integrated from multiple sources [1].

Graph-based RAG. To overcome the limitations of flat retrieval, recent work has incorporated graph structures into the RAG pipeline. GraphRAG [3] and CommunityKG-RAG [5] leverage graph clustering and community-level organization to support complex question answering and fact-checking, integrating knowledge at both the local entity level and the global community level. LightRAG [4] balances fine-grained entity retrieval with high-level abstraction, enabling efficient two-level retrieval. KG^{2}RAG [8] and HyperGraphRAG [9] further enrich retrieved information by leveraging structured relationships between chunks through knowledge graphs and hypergraph representations, respectively. On the reasoning front, Think-on-Graph 2.0 [10] alternates between document retrieval and KG traversal in an iterative cycle, while KG-RAG [11] combines query decomposition and multi-hop retrieval on the KG to support deep reasoning. Finally, HippoRAG [12] uses a PageRank-based traversal to simulate human associative memory, strengthening the factual connectivity among retrieved passages.

Limitations of existing approaches. Despite their effectiveness, most graph-based RAG frameworks leave room for improvement in two complementary directions. First, retrieved context is typically

assembled as a collection of individually selected triples or passages: although each unit may be locally relevant, there is no explicit mechanism to ensure structural continuity between them, which hampers multi-hop reasoning at generation time. Second, graph traversal strategies such as iterative expansion or PageRank-based diffusion do not actively reduce the search space during exploration, leading to scalability issues on large graphs as the number of candidate paths grows rapidly with the number of hops. PACO-RAG addresses both limitations jointly: pruning limits retrieval cost through early and permanent elimination of unpromising subtrees, while context-aware scoring at each hop ensures that only structurally consistent triples are retained, efficiently producing path-coherent reasoning chains.

3. Proposed Approach

3.1. Preliminaries

Given a KG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of entities and $\mathcal{E}$ the set of relational triples of the form (u, r, v) with $u, v \in \mathcal{V}$ and r a relation label, and a natural language query q, the objective is to retrieve a compact set of triples $\mathcal{R}_q \subseteq \mathcal{E}$ organized into path-coherent reasoning chains that provide structured evidence for downstream generation. PACO-RAG is agnostic to the source and construction of $\mathcal{G}$, which may be derived from document corpora via entity extraction or obtained from any pre-existing knowledge base, provided that entities and relations are represented in a shared embedding space.

PACO-RAG works in three stages: relevant community selection and root node identification (Section 3.2), multi-signal triple scoring (Section 3.3), and BFS expansion to construct the final context $\mathcal{R}_q$ delivered to the LLM (Section 3.4).

3.2. Relevant Community Extraction

Following standard KG-RAG practice [5], we assume the KG $\mathcal{G}$ is partitioned into a set of non-overlapping communities $\mathscr{C} = \{C_1, \ldots, C_M\}$, obtained via graph community detection (e.g., Louvain algorithm). Each community C_i groups structurally related entities and the relationships among them.

Given a query q, we rank the communities by the cosine similarity between the query embedding $E(q)$ and the community embedding $E(C_i)$, defined as the centroid of its nodes embedding:

$$E(C_i) = \frac{1}{|C_i|} \sum_{v \in C_i} E(v) \tag{3.1}$$

We retain the top 50% of communities by this ranking, yielding the relevant subset $\mathscr{C}_q \subseteq \mathscr{C}$.

Within each selected community $C \in \mathscr{C}_q$, we identify a root node v_C^* as the entity most semantically aligned with the query:

$$v_C^* = \arg\max_{v \in C} \Big(\mathrm{sim}\big(E(q),\, E(v)\big) \Big) \tag{3.2}$$

where $\mathrm{sim}(\cdot, \cdot)$ denotes the cosine similarity between the embeddings of two elements.

This node serves as the semantic anchor from which the hop-by-hop expansion described in Section 3.4 is initialized. The pruning procedure is applied independently to each community in $\mathscr{C}_q$.

3.3. Triple Relevance Metric

The triple selection process proceeds hop-by-hop and is guided by a composite scoring function that balances three complementary signals: query alignment, contextual non-redundancy, and relation-level pertinence.

At hop t, let $\mathcal{E}_t$ be the set of triples already accepted into the current reasoning context (we drop the community superscript for readability; the expansion is applied independently within each $C \in \mathscr{C}_q$). Given a query q, we define three scoring components for a candidate triple (u, r, v).

Query relevance measures the direct semantic alignment between the candidate target node v and the query; it is formalized as $\Psi_{\mathrm{node}}(v) = \mathrm{sim}(q,\, v)$.

Contextual redundancy penalizes candidates that are too similar to the evidence already collected. Let $S_t = \{v^*\} \cup \{w : (\cdot, \cdot, w) \in \mathcal{E}_t\}$ be the set of nodes in the current reasoning context and $\mu_t = \frac{1}{|S_t|} \sum_{w \in S_t} E(w)$ their mean embedding. Then, contextual redundancy is evaluated as $\Psi_{\text{ctx}}(v) = \text{sim}(v,\, \mu_t)$.

Finally, relation pertinence captures how well the traversed relation aligns with the query, and is defined as $\Psi_{\text{rel}}(r) = \text{sim}(q,\, r)$.

The overall score of a candidate triple is the weighted combination:

$$s(u, r, v) \;=\; \alpha \cdot \Psi_{\text{node}}(v) \;-\; \beta \cdot \Psi_{\text{ctx}}(v) \;+\; \gamma \cdot \Psi_{\text{rel}}(r) \tag{3.3}$$

where $\alpha, \gamma, \beta \in [0, 1]$ and $\alpha + \gamma = 1$. This constraint decouples the relevance–pertinence balance (controlled by α and γ) from the redundancy penalty (controlled by β), allowing the two effects to be tuned independently. The first and third terms jointly promote semantic relevance at the node and relation level, while the subtracted context term steers the expansion toward nodes that contribute complementary evidence rather than semantically overlapping information.

3.4. Branch Pruning and Context Aggregation

To prevent the exponential growth of the search space typical of multi-hop graph traversals, we apply a threshold-based pruning criterion at each hop. As in the previous subsection, we describe the procedure for a single community $C \in \mathscr{C}_q$; it is applied independently to every selected community.

The expansion follows a Breadth-First Search (BFS) rooted at v^*, with initial frontier $F_0 = \{v^*\}$ and empty triple set $\mathcal{E}_0 = \emptyset$. At each hop t, we examine every outgoing triple (u, r, v) whose source u belongs to the current frontier and whose target v lies within the community. A candidate triple is retained only if its score exceeds a predefined threshold τ:

$$\text{KEEP}(u, r, v) = \begin{cases} 1 & \text{if } s(u, r, v) \geq \tau, \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

Let Δ_t denote the set of triples accepted at hop t. These triples are accumulated into the reasoning context:

$$\mathcal{E}_t = \mathcal{E}_{t-1} \cup \Delta_t \tag{3.5}$$

and their target nodes form the new exploration frontier:

$$F_t = \{v : (u, r, v) \in \Delta_t\} \tag{3.6}$$

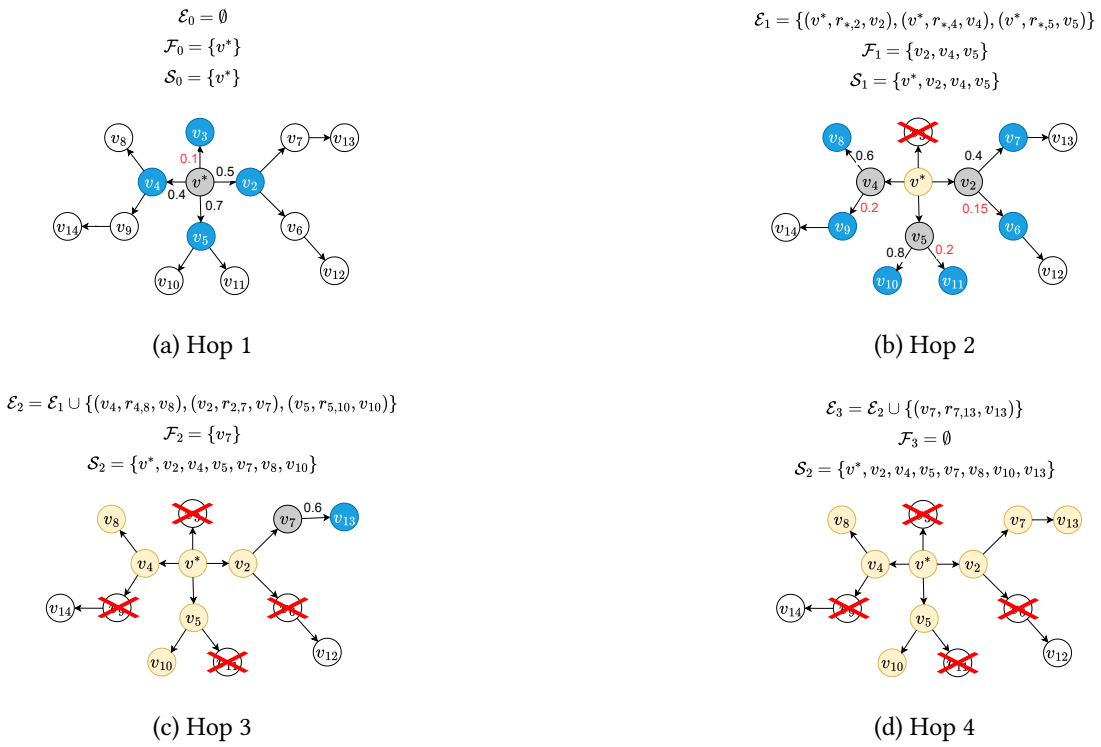
During the pruning process the discarded triples permanently exclude their target nodes from further expansion, strictly bounding the search space without any loss of path-coherent evidence, since discarded branches have already been deemed irrelevant by the scoring function.

The process terminates at the first hop T for which no triple survives the threshold, leaving the frontier empty. The output for community C is the accumulated triple set $\mathcal{E}_C^*$. Figure 1 illustrates this process across four hops, showing how unpromising branches are permanently discarded at each step, progressively reducing the search space while retaining only path-coherent triples.

The global context delivered to the language model is obtained by aggregating the triple sets produced independently from all relevant communities:

$$\mathcal{R}_q = \bigcup_{C \in \mathscr{C}_q} \mathcal{E}_C^*, \tag{3.7}$$

where the community superscript distinguishes the outputs of different communities. The resulting set $\mathcal{R}_q$ is structured as a sequence of prompts representing path-coherent reasoning chains rather than a fragmented collection of independent facts, providing the LLM with structured evidence that directly supports multi-hop inference.



(a) Hop 1

(b) Hop 2

(c) Hop 3

(d) Hop 4

Figure 1: Branch pruning within a community. The graph expands hop-by-hop from the root node v^*. At each hop, candidate triples are scored using Equation (3.3); a branch is expanded further only if its score exceeds the threshold τ, otherwise the entire downstream subtree is discarded. Grey nodes represent the current frontier and blue nodes are those selected at the current hop.

4. Experiments

This section presents the experimental evaluation of PACO-RAG. In Section 4.1, we describe the experimental setup. Then in Section 4.2, we report the main results in terms of both performance and efficiency, followed by a sensitivity analysis in Section 4.3.

4.1. Experimental Setup

Implementation Details. The construction of the KG follows the same procedure of [5], in which entities and relations are automatically extracted from source documents through natural language processing techniques, yielding a graph where entities are represented as nodes and their semantic relationships as edges. Community detection is performed using the Louvain algorithm [13]. All embeddings, including those for queries, entities, and relations, are computed with Nomic Embed Text v1.5 [14], mapping all elements into a shared 768-dimensional latent space used throughout the scoring function in Equation (3.3). The top 50% of communities are retained at each query. The retrieved context $\mathcal{R}_q$ is internally structured as a set of trees, one per selected community, rooted at v_C^*; at each hop, the frontier represents the current leaves. Once no further nodes can be explored, the root-to-leaf paths are serialized as ordered lists of triples and passed to the LLM to preserve reasoning chain coherence. All experiments are conducted on an NVIDIA A100 40GB GPU with 128 GB of RAM. The code used for our study is publicly available at the link https://github.com/DavideTraini/PACO-RAG.

Datasets. We evaluate PACO-RAG on a fact-checking task over two benchmark datasets, MOCHEG [15] and HoVer [16]. Both require verifying claims by integrating evidence distributed across multiple entities and relationships, making them well suited for evaluating structured retrieval

and multi-hop reasoning. Each dataset pairs every claim with a collection of source documents serving as the evidence pool; the system must retrieve the relevant information from this pool to determine the claim's veracity. We sample 100 claims per dataset. The prediction target is a veracity label, either *supported*, *refuted*, or *not enough information* (NEI) for MOCHEG, and *supported* or *refuted* for HoVer.

Metrics. Accuracy (Acc.) and F1-score (F1) are used as performance metrics, while response time in seconds (Time) and context length in tokens (Ctx) serve as efficiency metrics.

Models. All experiments are conducted using two generative LLMs as the answering backbone, i.e., `gemma3:27b`[1] and `gemma3:12b`[2].

Baselines. We compare PACO-RAG against three representative KG-RAG approaches and a no-retrieval baseline. All reported values are averaged over 5 independent runs. The No-RAG baseline prompts the generative model with the claim alone, without any retrieved context, isolating the model's parametric knowledge. LightRAG [4] is a dual-level retrieval framework that combines fine-grained entity-level retrieval with high-level key-phrase abstraction, balancing specificity and coverage in the retrieved context. CommunityKG-RAG [5] leverages graph clustering to organize evidence at both the local entity level and the global community level, integrating structured knowledge for fact-checking. GraphRAG [3] builds a hierarchical community structure over the KG and generates community-level summaries to support query-focused reasoning. Together, these baselines cover flat retrieval, dual-level retrieval, and community-based graph retrieval, providing a broad comparison spectrum against which the path-coherent retrieval strategy of PACO-RAG can be evaluated.

4.2. Comparison Results

Performance. Table 1 shows that PACO-RAG outperforms all baselines in Accuracy across both datasets and both model sizes. With `gemma3:27b` (Table 1a), PACO-RAG ranks first on every metric. On MOCHEG, it surpasses the second-best approach (No-RAG) by $+8.3\%$ in Accuracy. Notably, all graph-based baselines fall below the No-RAG baseline on this dataset, indicating that their retrieved context introduces more noise than useful evidence. On HoVer, the gains are particularly strong, with PACO-RAG surpassing CommunityKG-RAG by $+10.0\%$ in both Accuracy and F1, confirming the advantage of path-coherent retrieval on a dataset that requires multi-hop reasoning across multiple documents.

A similar trend holds with the smaller `gemma3:12b` model (Table 1b), where PACO-RAG again leads in Accuracy on both datasets. On MOCHEG, the margin is narrower ($+2.1\%$ over No-RAG and GraphRAG), while on HoVer PACO-RAG improves over CommunityKG-RAG by $+5.1\%$. Notably, PACO-RAG paired with the smaller model achieves better results than most baselines running on the larger one, suggesting that providing a structurally coherent multi-hop context effectively offloads the reasoning burden and allows a lighter model to perform accurate fact-checking.

Efficiency. Table 2 reports the efficiency results, where time and context length are averaged over all claims and refer to a single query. While the No-RAG setting is naturally the fastest since it skips retrieval entirely, PACO-RAG operates in less than a second on both datasets with both models, reducing execution time by up to 98.4% over CommunityKG-RAG and by 73.5% over LightRAG, the fastest graph-based baseline on `gemma3:27b`. A similar trend is observed with `gemma3:12b`. This speedup is directly driven by the pruning mechanism and the resulting context length. Existing graph-based methods retrieve large amounts of text, whereas PACO-RAG's branch pruning yields extremely compact contexts, reducing context length by up to 96.6% over CommunityKG-RAG on MOCHEG and by 92.3% over GraphRAG, the baseline with the smallest context on the same dataset. PACO-RAG retrieves only

[1] https://ollama.com/library/gemma3:27b-it-qat
[2] https://ollama.com/library/gemma3:12b

Table 1
Comparison results using different generative models: (a) `gemma3:27b` and (b) `gemma3:12b`. Best and second-best results are highlighted in bold and underlined, respectively.

Approach	MOCHEG Acc.	MOCHEG F1	HoVer Acc.	HoVer F1
No-RAG	0.48	0.46	0.57	0.54
LightRAG [4]	0.45	0.38	0.56	0.52
CommunityKG-RAG [5]	0.41	0.39	0.60	0.60
GraphRAG [3]	0.44	0.30	0.52	0.43
PACO-RAG	**0.52**	**0.47**	**0.66**	**0.66**

(a) `gemma3:27b`

Approach	MOCHEG Acc.	MOCHEG F1	HoVer Acc.	HoVer F1
No-RAG	0.48	**0.46**	0.55	0.54
LightRAG [4]	0.44	0.32	0.52	0.49
CommunityKG-RAG [5]	0.45	0.39	0.59	0.59
GraphRAG [3]	0.48	0.39	0.49	0.41
PACO-RAG	**0.49**	0.42	**0.62**	**0.62**

(b) `gemma3:12b`

the most relevant and path-coherent triples, drastically reducing the input sequence length, minimizing generation time and limiting exposure to noise-induced hallucinations without sacrificing accuracy.

Table 2
Efficiency results using different generative models: (a) `gemma3:27b` and (b) `gemma3:12b`. Time is the average retrieval and answer time in seconds; Ctx is the average context length in tokens. Best and second-best results are highlighted in bold and underlined, respectively.

Approach	MOCHEG Time (s)	MOCHEG Ctx	HoVer Time (s)	HoVer Ctx
No-RAG	**0.40**	–	**0.35**	–
LightRAG [4]	3.13	10,879	3.24	7,451
CommunityKG-RAG [5]	52.82	16,273	13.93	1,083
GraphRAG [3]	23.58	7,313	23.69	1,786
PACO-RAG	0.83	**560**	0.69	**473**

(a) `gemma3:27b`

Approach	MOCHEG Time (s)	MOCHEG Ctx	HoVer Time (s)	HoVer Ctx
No-RAG	**0.22**	–	**0.23**	–
LightRAG [4]	3.11	10,879	3.17	7,451
CommunityKG-RAG [5]	44.47	16,273	9.40	1,083
GraphRAG [3]	19.71	7,153	19.88	2,313
PACO-RAG	0.48	**560**	0.37	**473**

(b) `gemma3:12b`

4.3. Hyperparameter Analysis

In this section we perform a sensitivity analysis of PACO-RAG using the `gemma3:27b` model. The parameters under analysis are the query relevance weight α, the contextual redundancy penalty β, and the pruning threshold τ. Due to space constraints, we do not report the full grid search over all hyperparameter combinations. Instead, for each dataset, we vary one parameter while fixing the others at their optimal values ($\alpha = 0.25$, $\beta = 0.30$, $\tau = 0.05$ for MOCHEG, and $\alpha = 0.75$, $\beta = 0.10$, $\tau = 0.45$ for HoVer). Note that the relation pertinence weight γ is not varied independently, as it is fully determined by α through the constraint $\alpha + \gamma = 1$. Results are shown in Tables 3 and 4.

The query relevance weight α balances direct semantic alignment against diversity. On MOCHEG, lower values maximize performance, peaking at $\alpha = 0.25$. On HoVer, stronger query alignment is required, with best results at $\alpha = 0.75$. On both datasets, setting $\alpha = 1.00$ degrades accuracy, confirming that relation pertinence and redundancy control are necessary complements to query alignment.

The contextual redundancy penalty β prevents the accumulation of semantically overlapping nodes. Setting $\beta = 0.00$ forces the retriever to select overly homogeneous triples, harming multi-hop reasoning. MOCHEG peaks at $\beta = 0.30$, while HoVer favors a milder constraint at $\beta = 0.10$. Increasing β too aggressively (e.g., $\beta = 0.40$) excessively penalizes potentially useful triples, and although this compresses context length and reduces execution time, it degrades fact-checking accuracy on both datasets.

The pruning threshold τ controls how aggressively unpromising branches are discarded. Higher thresholds yield smaller contexts and faster execution times (e.g., 531 tokens and 0.79s on MOCHEG at $\tau = 0.45$). However, the optimal value depends on dataset complexity: MOCHEG requires a permissive threshold ($\tau = 0.05$) to preserve the reasoning paths necessary for its three-class label space, whereas

HoVer benefits from strict pruning ($\tau = 0.45$), which filters noise and produces compact, high-precision contexts. This dataset-dependent behavior is consistent with the structural differences between the two tasks.

Table 3
Sensitivity analysis of hyperparameters α, β, and τ on the MOCHEG dataset. The best overall combination is $\alpha = 0.25$, $\beta = 0.30$, $\tau = 0.05$.

α	Acc.	F1	Time (s)	Ctx
0.00	**0.52**	0.46	**0.82**	**558**
0.25	**0.52**	**0.47**	0.83	560
0.50	0.49	0.42	0.85	561
0.75	0.47	0.40	0.85	566
1.00	0.48	0.38	0.87	575

(a) Varying α

β	Acc.	F1	Time (s)	Ctx
0.00	0.49	0.42	0.86	564
0.10	0.48	0.42	0.86	564
0.20	0.49	0.42	0.84	562
0.30	**0.52**	**0.47**	**0.83**	**561**
0.40	0.51	0.44	**0.83**	**560**

(b) Varying β

τ	Acc.	F1	Time (s)	Ctx
0.05	**0.52**	**0.47**	0.83	560
0.15	0.48	0.42	0.83	560
0.25	0.50	0.46	0.83	560
0.35	0.49	0.42	0.80	542
0.45	0.49	0.42	**0.79**	**531**

(c) Varying τ

Table 4
Sensitivity analysis of hyperparameters α, β, and τ on the HoVer dataset. The best overall combination is $\alpha = 0.75$, $\beta = 0.10$, $\tau = 0.45$.

α	Acc.	F1	Time (s)	Ctx
0.00	0.63	0.63	**0.68**	**469**
0.25	0.61	0.61	**0.68**	**469**
0.50	0.62	0.62	0.69	470
0.75	**0.66**	**0.66**	0.69	473
1.00	0.61	0.61	0.70	478

(a) Varying α

β	Acc.	F1	Time (s)	Ctx
0.00	0.63	0.63	0.71	486
0.10	**0.66**	**0.66**	0.69	473
0.20	0.62	0.62	0.66	452
0.30	0.61	0.61	**0.65**	450
0.40	0.61	0.61	**0.65**	**449**

(b) Varying β

τ	Acc.	F1	Time (s)	Ctx
0.05	0.63	0.63	0.74	509
0.15	0.64	0.64	0.74	509
0.25	0.64	0.64	0.74	509
0.35	0.63	0.63	0.74	509
0.45	**0.66**	**0.66**	**0.69**	**473**

(c) Varying τ

5. Conclusion

In this paper, we addressed the path coherence gap in existing KG-RAG frameworks by proposing PACO-RAG, a context-aware branch pruning approach that retrieves compact, path-coherent sets of triples organized into explicit reasoning chains via BFS guided by a multi-signal scoring function, closing the gap between retrieved evidence and the relational structure that multi-hop reasoning requires.

Experimental evaluations on MOCHEG and HoVer demonstrate that PACO-RAG consistently outperforms LightRAG, GraphRAG, and CommunityKG-RAG in both accuracy and F1-score. The pruning strategy reduces the retrieval search space to contexts of approximately 500 tokens, allowing the system to operate in under a second while limiting exposure to noise-induced hallucinations without sacrificing factual accuracy.

A notable finding is that providing a structurally coherent multi-hop context allows a smaller model such as `gemma3:12b` to outperform larger 27B parameter models paired with traditional RAG methods, suggesting that retrieval quality can effectively substitute for model scale in structured reasoning tasks. Hyperparameter analysis further reveals that the optimal configuration depends on dataset complexity: HoVer benefits from strict pruning, while MOCHEG requires a more permissive threshold to preserve the reasoning paths necessary for its three-class verification task.

Future work could explore dynamic, learning-based hyperparameter tuning to automatically adapt pruning thresholds and scoring weights to the characteristics of individual queries. Extending PACO-RAG to other multi-hop reasoning tasks and larger-scale knowledge graphs will further assess its scalability and generalization. More broadly, path-coherent retrieval represents a promising direction for building reliable and efficient knowledge-augmented generation systems.

Declaration on Generative AI

The authors used a Generative AI tool (ChatGPT by OpenAI) solely for language polishing (grammar and readability improvements). No scientific content was generated by the tool. All content has been reviewed and approved by the authors.

Acknowledgments

We acknowledge the support of the project "MEraviglIA - Metodologie didattiche inclusive ed Intelligenza Artificiale" (J11I24000700009) under the PR Marche FSE+ 2021/2027 funded by Regione Marche.

References

[1] Y. Wang, M. Wang, M. A. Manzoor, F. Liu, G. N. Georgiev, R. J. Das, P. Nakov, Factuality of Large Language Models: A Survey, in: Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP'24), Miami, FL, USA, 2024, pp. 19519–19529. Association for Computational Linguistics.

[2] B. Peng, Y. Zhu, Y. Liu, X. Bo, H. Shi, C. Hong, Y. Zhang, S. Tang, Graph retrieval-augmented generation: A survey, ACM Transactions on Information Systems 44 (2025) 1–52. ACM New York, NY.

[3] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, J. Larson, From local to global: A graph rag approach to query-focused summarization, arXiv preprint arXiv:2404.16130 (2024).

[4] Z. Guo, L. Xia, Y. Yu, T. Ao, C. Huang, LightRAG: Simple and Fast Retrieval-Augmented Generation, in: Findings of the Association for Computational Linguistics: Conference on Empirical Methods in Natural Language Processing (EMNLP'25), Suzhou, China, 2025, pp. 10746—-10761. Association for Computational Linguistics.

[5] R. C. Chang, J. Zhang, Communitykg-rag: Leveraging community structures in knowledge graphs for advanced retrieval-augmented generation in fact-checking, arXiv preprint arXiv:2408.08535 (2024).

[6] B. Chen, Z. Guo, Z. Yang, Y. Chen, J. Chen, Z. Liu, C. Shi, C. Yang, Pathrag: Pruning graph-based retrieval augmented generation with relational paths, arXiv preprint arXiv:2502.14902 (2025).

[7] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, Retrieval-augmented generation for large language models: A survey, arXiv preprint arXiv:2312.10997 2 (2023).

[8] Y. Zhu, D. Liu, Z. Lin, W. Tong, S. Zhong, J. Shao, The LLM Already Knows: Estimating LLM-Perceived Question Difficulty via Hidden Representations, in: Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP'25), Suzhou, China, 2025, pp. 1160–1176. ACL.

[9] H. Luo, G. Chen, Y. Zheng, X. Wu, Y. Guo, Q. Lin, Y. Feng, Z. Kuang, M. Song, Y. Zhu, A. T. Luu, HyperGraphRAG: Retrieval-Augmented Generation via Hypergraph-Structured Knowledge Representation, in: Proc. of the Conference on Neural Information Processing Systems (NeurIPS'25), San Diego, California, USA, 2025.

[10] S. Ma, C. Xu, X. Jiang, M. Li, H. Qu, C. Yang, J. Mao, J. Guo, Think-on-Graph 2.0: Deep and Faithful Large Language Model Reasoning with Knowledge-guided Retrieval Augmented Generation, in: Proc. of the International Conference on Learning Representations, (ICLR'25), Singapore, 2025. OpenReview.net.

[11] J. Linders, J. M. Tomczak, Knowledge graph-extended retrieval augmented generation for question answering, Applied Intelligence 55 (2025) 1102.

[12] B. J. Gutiérrez, Y. Shu, W. Qi, S. Zhou, Y. Su, From RAG to Memory: Non-Parametric Continual Learning for Large Language Models, in: Proc. of the Forty-second International Conference on Machine Learning (ICML'25), Vancouver, BC, Canada, 2025. OpenReview.net.

[13] V. Blondel, J. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment 2008 (2008) P10008. IOP Publishing.
[14] Z. Nussbaum, J. X. Morris, A. Mulyar, B. Duderstadt, Nomic Embed: Training a Reproducible Long Context Text Embedder, Transactions on Machine Learning Research 1 (2025).
[15] B. M. Yao, A. Shah, L. Sun, J. H. Cho, L. Huang, End-to-End Multimodal Fact-Checking and Explanation Generation: A Challenging Dataset and Models, in: Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR'23), Taipei, Taiwan, 2023, pp. 2733–2743. ACM.
[16] Y. Jiang, S. Bordia, Z. Zhong, C. Dognin, M. K. Singh, M. Bansal, HoVer: A Dataset for Many-Hop Fact Extraction And Claim Verification, in: Findings of the Association for Computational Linguistics (EMNLP'20), Virtual Event, 2020, pp. 3441–3460. Association for Computational Linguistics.