SEMANTIC GAIN GRAPHS FOR COMPLEMENTARITY-GUIDED RETRIEVAL-AUGMENTED GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-hop Retrieval-Augmented Generation (RAG) requires discovering evidence whose value emerges through combination across documents. Dense retrievers rank chunks by isolated query relevance, while graph-based retrievers traverse connections that do not indicate whether neighboring documents contribute useful evidence. We propose SG^{2}RAG, which reframes graph retrieval around semantic gain, defined as the useful and nonredundant evidence that one document chunk contributes in the context of another. SG^{2}RAG aggregates these assessments into document-pair strengths in a Semantic Gain Graph, turning cross-document complementarity into an explicit retrieval relation. Starting from query-relevant seeds, controlled Personalized PageRank composes local gain relations into a global multi-hop ranking, following complementary evidence chains while maintaining query focus. This design moves evidence discovery beyond similarity ranking and topology-only traversal. Experiments on three multi-hop benchmarks show that SG^{2}RAG achieves the best overall performance. Under identical candidate support and propagation, semantic-gain weighting consistently outperforms similarity weighting. Gold-support analysis further validates the gain signal, while improvements across iterative retrieval frameworks demonstrate its modularity.

1 INTRODUCTION

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) has become an important paradigm for mitigating hallucinations in Large Language Models (LLMs) and supporting knowledge-intensive applications (Song & Lee, 2026; Chu et al., 2026). By incorporating external evidence, RAG improves the groundedness and reliability of LLM outputs, making knowledge organization and retrieval effectiveness critical to its reasoning capability (Li et al., 2025). Early studies improve retrieval by preserving the semantic coherence of text snippets through heuristic chunking (Hostnik & Robnik-Šikonja, 2025; Zhao et al., 2025) and summarization (Ji et al., 2024; Hosking et al., 2024). However, these methods remain overly reliant on semantic similarity. In multi-hop reasoning (Wei et al., 2025), key evidence is distributed across documents, and a document with weak query similarity may become critical once another document is known. Because similarity-based retrievers score each document primarily against the query, they do not directly represent how documents complete one another. Iterative methods expand the context through multi-round retrieval-generation interactions (Shao et al., 2023; Yue et al., 2025), but introduce substantial inference overhead while leaving cross-document complementarity implicit.

To better organize complex evidence structures, recent studies have introduced graph-based retrieval into RAG (Wang et al., 2026), moving from *similarity-based* retrieval toward *topology-based* evidence expansion. LightRAG (Guo et al., 2024), for example, constructs LLM-assisted entity-relation graphs and retrieves contexts through entity linking and graph traversal. This topology extends retrieval beyond direct query matches, but connectivity does not reveal which neighboring documents contribute useful evidence. Methods such as HippoRAG2 (Jimenez Gutierrez et al., 2024; Gutiérrez et al., 2025) incorporate query-document relevance to constrain the traversal scope, yet still rely on entity-level relations organized by LLMs rather than explicitly assessing cross-document complementarity. As Figure 1 illustrates, traversal from the query-relevant "Mugain" document can follow entity co-occurrence links to "Deep Blue Sea" and "Doria Ragland", which do not support the target family relation, while evidence needed to complete the reasoning chain may not be favored by topology or document-level relevance. Existing retrievers therefore locate evidence through query

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.

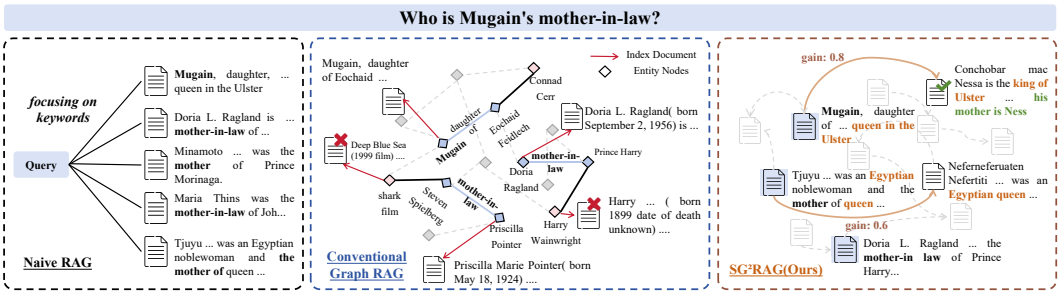


Figure 1: Naive RAG retrieves query-similar texts but misses implicit complementary evidence. GraphRAG extends retrieval through topology, but connected documents such as "Deep Blue Sea" do not necessarily support the target relation. Starting from query-relevant seeds, SG^{2}RAG follows gain-weighted connections to recover complementary evidence showing that the "King of Ulster's mother" is "Mugain's mother-in-law."

relevance or expand it through graph connectivity, but neither makes cross-document complementarity a first-class retrieval relation.

To address this gap, we propose **SG^{2}RAG**, a **S**emantic **G**ain **G**raph based **R**etrieval-**A**ugmented **G**eneration framework. SG^{2}RAG makes cross-document complementarity a first-class retrieval relation, moving evidence discovery beyond independent query-document matching and topology-only expansion. It represents document chunks as nodes and estimates the relevant, novel, and significant evidence that one chunk contributes when another is available. These ordered assessments are aggregated into pair-level strengths that form a reusable Semantic Gain Graph. Similarity provides efficient candidate support and query entry points, while semantic gain defines the graph relation and routing preference. During retrieval, controlled Personalized PageRank (PPR) (Page et al., 1999) composes local gain-weighted connections into a multi-hop evidence ranking anchored to the query. Our contributions are threefold. (1) We formulate multi-hop evidence discovery around document-level conditional evidence contribution as an explicit retrieval objective rather than independent query relevance or topology alone. (2) We introduce the Semantic Gain Graph as a reusable complementarity index whose pair strengths are derived from ordered evidence-contribution assessments, and develop query-anchored controlled PPR to compose these relations into globally coordinated multi-hop evidence rankings. (3) Extensive experiments across two LLM backbones show that SG^{2}RAG achieves the best average performance, isolate semantic-gain weighting under fixed graph support and propagation, validate the score with gold supporting-document pairs, and demonstrate modularity in iterative retrieval frameworks.

2 Related Work

2.1 Retrieval-Augmented Generation

RAG grounds LLM generation in external knowledge, improving factuality and reducing hallucinations in knowledge-intensive tasks (Lewis et al., 2020; Vidyarthi et al., 2026; Jiang et al., 2025). Retrieval advances refine knowledge units through fine-grained segmentation, as in MetaChunking (Zhao et al., 2024) and SAKIRAG (Tao et al., 2025), and semantic chunking (Moreno-Cediel et al., 2025), while dense embeddings efficiently match these units to queries (Lee et al., 2024; Zhang et al., 2025). Although effective for locally self-contained evidence, this flat, similarity-driven pipeline struggles with multi-hop reasoning, where supporting evidence is distributed across implicitly related chunks and may have low query similarity (Wei et al., 2025). Iterative methods, including iterDRAG (Yue et al., 2025), RT-RAG (Shi et al., 2026), and iterRetGen (Shao et al., 2023), address this limitation through query rewriting, question decomposition, and repeated retrieval-generation interactions. However, they incur online inference costs while leaving complementary relations among knowledge units implicit in the flat corpus index.

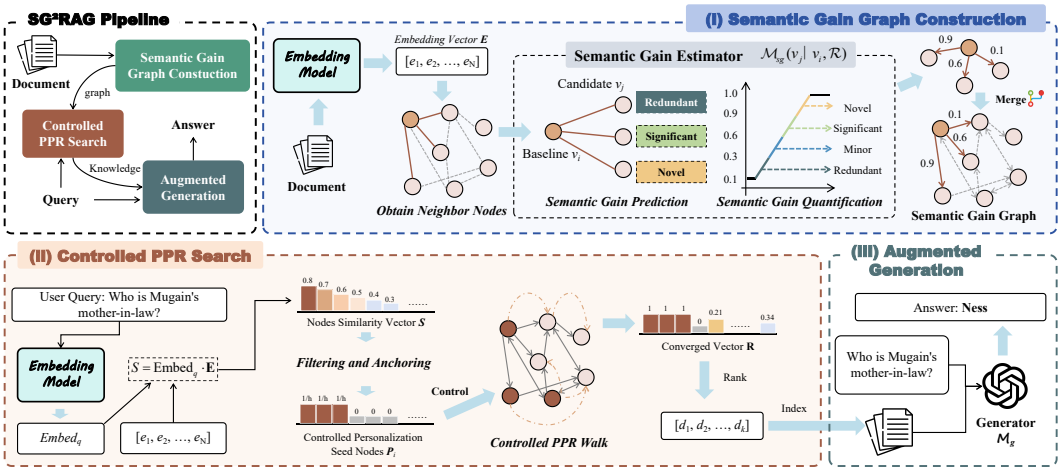


Figure 2: Framework of SG^{2}RAG. The offline stage estimates ordered evidence contribution and aggregates the assessments into pair-level strengths in a reusable Semantic Gain Graph. The online stage uses query-relevant seeds to anchor controlled PPR over these complementarity relations.

2.2 Graph-based RAG

Graph-based RAG addresses the limitations of flat retrieval by organizing knowledge into structured indexes. RAPTOR (Sarthi et al., 2024) builds a hierarchical tree through recursive clustering and summarization. GraphRAG (Edge et al., 2024) and LightRAG (Guo et al., 2024) construct entity-relation graphs for entity linking and graph expansion. Building on such structures, KG^{2}RAG (Zhu et al., 2025) expands semantically retrieved seeds under knowledge graph guidance and organizes the evidence into coherent paragraphs. Together, these methods capture associations beyond surface similarity. However, most graph-based retrievers remain governed by topology. Unconstrained expansion can introduce noisy nodes and retrieval drift. PathRAG (Chen et al., 2026) uses flow-based pruning to remove redundant information from retrieved relational paths. LinearRAG (Zhuang et al., 2025) and HippoRAG2 (Jimenez Gutierrez et al., 2024; Gutiérrez et al., 2025) restrict traversal to improve topic relevance, but still assume that topological neighbors are useful without explicitly estimating document complementarity. GraphReader (Li et al., 2024) and GeAR (Shen et al., 2025) improve control through LLM agents that direct graph traversal, at the cost of additional online inference and tighter coupling to agentic reasoning. SG^{2}RAG instead makes estimated evidence contribution part of a reusable document graph, rather than treating complementarity as an implicit consequence of topology or a query-time ranking signal.

3 Preliminaries

Given a corpus $\mathcal{D} = \{d_i\}_{i=1}^{N}$, conventional Graph-RAG methods transform unstructured documents into a relation graph $\mathcal{G}_{rel} = (\mathcal{V}_{rel}, \mathcal{E}_{rel})$ through information extraction (Edge et al., 2024; Guo et al., 2024; Gutiérrez et al., 2025). The nodes represent knowledge units such as entities and phrases, while the edges encode extracted relations, shared mentions, or other associations. Given a query q, semantic matching identifies query-relevant seed nodes $\mathcal{V}_q \subseteq \mathcal{V}_{rel}$, and graph propagation ranks additional nodes. Retrieval is therefore governed largely by topology induced from recurring mentions and extracted relations. In contrast, SG^{2}RAG represents each document chunk d_i as a node v_i and assigns an ordered assessment w_{ij} to the useful and nonredundant evidence contributed by d_j when d_i is available. The ordered records are aggregated into a symmetric pair strength a_{ij} for graph traversal. Thus, assessment direction specifies how evidence contribution is measured, while pair strength determines how the graph is traversed. Query matching locates entry nodes, and controlled PPR (Page et al., 1999) ranks graph-reachable evidence over this reusable complementarity index.

4 METHODOLOGY

This section presents SG^{2}RAG as a complementarity-indexed retrieval framework with offline Semantic Gain Graph construction and online query-anchored controlled PPR, as shown in Figure 2.

4.1 SEMANTIC GAIN GRAPH CONSTRUCTION

Given a corpus, SG^{2}RAG first segments documents into chunks and represents each chunk as a graph node $v \in \mathcal{V}$. Because user queries are unavailable during offline construction, SG^{2}RAG precomputes reusable query-independent complementarity relations, separating corpus-level evidence completion from query-specific seed selection. To avoid exhaustive comparison among all chunk pairs, we adopt a coarse-to-fine graph construction strategy. Let $\mathbf{E} \in \mathbb{R}^{N\times d}$ denote the embedding matrix of all N chunks. By evaluating the top-n candidates of each node jointly in a single LLM request, graph construction requires only N LLM requests in total. For each node v_i, we first retrieve its top-n nearest neighbors according to cosine similarity:

$$\mathcal{N}(v_i) = \text{Top-}n\left(\cos(\mathbf{e}_i, \mathbf{E})\right), \tag{1}$$

where $\mathcal{N}(v_i)$ denotes the local semantic neighborhood of v_i. Similarity sparsifies the pairs to be assessed, while semantic gain assigns their retrieval utility. Similarity therefore defines candidate support rather than final routing, allowing graph search to move beyond direct query similarity within that support. Appendix B.2 analyzes this coverage boundary and its interaction with gain-guided traversal.

Inspired by conditional gain in set functions (Iyer et al., 2021) and novelty-aware selection (Carbonell & Goldstein, 1998), we define semantic gain as an operational measure of conditional evidence contribution. For an ordered pair, it scores the relevant, novel, and significant information that v_j adds when v_i is available. Unlike query-conditioned reranking, this assessment defines a reusable document-to-document relation without reference to a query. An LLM-based scorer $\mathcal{M}_{sg}$ interprets each baseline-candidate pair under a fixed rubric $\mathcal{R}$ and returns the bounded score record

$$w_{ij} = \mathcal{M}_{sg}(v_j \mid v_i, \mathcal{R}) \in [0, 1]. \tag{2}$$

Because the score is conditioned on one ordered baseline-candidate pair, it is local rather than marginal gain conditioned on the complete retrieved set. Thus, w_{ij} and w_{ji} are evaluated separately and need not be equal. Assessment direction and traversal direction serve different roles. Each observed score is retained as an ordered record, and the default method defines the symmetric pair strength $a_{ij} = a_{ji} = \sum_{e\in\mathcal{E}_{ij}} w_e$, where $\mathcal{E}_{ij}$ contains all available records for the pair. A missing reverse record can result from nonreciprocal Top-n proposal rather than zero contribution in the reverse order. Interpreting records as directed arcs would therefore turn proposal sparsity into a one-way reachability constraint. Pair Sum avoids this constraint while retaining all observed assessments. After degree normalization, a pair with greater aggregated gain receives a larger relative transition probability. The rubric $\mathcal{R}$ defines four intervals, namely *Redundant*, *Minor*, *Significant*, and *Novel*. Evaluating all local neighborhoods yields the undirected semantic gain multigraph $\mathcal{G}_{sg} = (\mathcal{V}, \mathcal{E}_{sg})$. Section 5.6 and Appendix C.1 provide further analysis.

4.2 CONTROLLED PPR SEARCH

Given the Semantic Gain Graph $\mathcal{G}_{sg}$, the online stage uses query relevance to locate entry points rather than to determine the final ranking directly. SG^{2}RAG matches the query with chunk nodes and uses the most relevant chunks to construct a personalization vector. Controlled PPR then remains anchored to this query-relevant region while expanding through semantic-gain connections.

For a query q, we compute its relevance scores to all chunk nodes as $S = \text{Embed}_q \cdot \mathbf{E}$. Standard PPR uses the all-node vector $(P_q^{\text{all}})_i = S_i^+ / \sum_l S_l^+$, where $S_i^+ = \max(S_i, 0)$, with a uniform fallback if the denominator is zero. Controlled PPR retains the Top-h query-similar chunks as the seed set and uses $(P_q^{\text{ctrl}})_i = \mathbb{I}(v_i \in \mathcal{V}_q)/|\mathcal{V}_q|$. An optional LLM-based $Filter$ can refine these candidates before propagation, giving $\mathcal{V}_q = \text{Filter}_{\mathcal{M}}(\text{Top-}h(S))$. Without the filter, $\mathcal{V}_q = \text{Top-}h(S)$. If the filter returns an empty set, we fall back to the unfiltered Top-h set. Each retained seed receives mass $1/|\mathcal{V}_q|$, while all other nodes receive zero. We use P_q^{ctrl} as both the initial distribution and the restart vector

$$r^{(0)} = P_q^{\text{ctrl}}, \qquad r^{(t+1)} = \alpha\mathcal{T} \cdot r^{(t)} + (1-\alpha) \cdot P_q^{\text{ctrl}}, \tag{3}$$

where $\mathcal{T}$ is the column-stochastic transition matrix derived from the symmetric pair strengths described above. Let $\mathcal{N}_{\mathcal{G}}(v_i)$ denote the neighbors of v_i in the Semantic Gain Graph and let $d_i = \sum_{v_l \in \mathcal{N}_{\mathcal{G}}(v_i)} a_{il}$ be its weighted degree. For $d_i > 0$, $\mathcal{T}_{ji} = a_{ij}/d_i$, so column i represents propagation from v_i to its graph neighbors. If $d_i = 0$, we set $\mathcal{T}_{ii} = 1$ and all other entries in column i to zero, treating v_i as a self-loop. The ordered assessments determine pair strength, while the symmetric adjacency determines reachability. Weighted-degree normalization then converts each a_{ij} into a relative routing preference, assigning less transition probability to weakly complementary connections when stronger alternatives are available. Query anchoring and propagation depth are controlled separately by P_q^{ctrl} and the restart probability $1 - \alpha$. Following the value recommended in prior work (Gutiérrez et al., 2025), we set $\alpha = 0.5$ by default. Here, $r^{(t)}$ is the ranking vector at iteration t. The iteration stops when $\|r^{(t+1)} - r^{(t)}\|_1 < \epsilon$. After convergence, the stationary ranking has the path expansion

$$\mathbf{R} = (1-\alpha)(I - \alpha\mathcal{T})^{-1} P_q^{\text{ctrl}} = (1-\alpha) \sum_{\ell=0}^{\infty} \alpha^{\ell} \mathcal{T}^{\ell} P_q^{\text{ctrl}}, \tag{4}$$

where I is the identity matrix. The transition matrix $\mathcal{T}$ encodes query-independent relative pairwise complementarity, while P_q^{ctrl} supplies query relevance through uniform mass on the selected seeds. Each term $\mathcal{T}^{\ell} P_q^{\text{ctrl}}$ aggregates length-ℓ paths through normalized semantic-gain connections, while α^{ℓ} discounts longer paths. Thus, $\mathbf{R}$ composes local complementarity relations into a query-anchored multi-hop ranking. It is not a reranking of the Top-h seed set because it assigns scores to graph-reachable chunks beyond the initial query-based retrieval, subject to the fixed graph support. This path aggregation does not compute set-conditioned marginal gain over the complete retrieved context. We then select the top-k nodes as retrieved evidence, denoted by $\mathcal{D} = \text{Index}\,(\text{Top-}k(\mathbf{R}))$. The retrieved chunks are concatenated with the query and fed into the generator LLM $\mathcal{M}_g$ to produce the final answer $\mathcal{A} = \mathcal{M}_g(\mathcal{D}, q)$.

5 Experiments

5.1 Datasets

We evaluate our method on the curated MuSiQue (Trivedi et al., 2022), HotpotQA (Yang et al., 2018), and 2WikiMultiHopQA (2Wiki) (Ho et al., 2020) datasets provided by HippoRAG2 (Gutiérrez et al., 2025). Each dataset contains 1000 queries together with supporting documents, distractors, and ground-truth answers for retrieval and QA evaluation. The document collections contain 11656 documents for MuSiQue, 9811 for HotpotQA, and 6119 for 2Wiki. MuSiQue is considered more challenging, while HotpotQA is relatively easier.

5.2 Baselines

We compare SG^{2}RAG with four groups of representative baselines. First, we evaluate pure LLMs without retrieval, including **Qwen2.5-72B-Instruct** (abbreviated as **Qwen-72B**) (Yang et al., 2024) and **Llama-3.3-70B-Instruct** (abbreviated as **Llama-70B**) (Grattafiori et al., 2024). Second, we consider standard dense-retrieval RAG methods using **BGE-M3** (Chen et al., 2024), **text-embedding-v3**[1], and **Qwen3-Embedding-8B** (abbreviated as **Qwen3-Emb**) (Zhang et al., 2025). We additionally construct a strong **Qwen3-Emb w/ Reranker** baseline to test whether query-conditioned LLM reranking can account for the gains of SG^{2}RAG. Third, we compare with structure-augmented retrieval methods, including **LightRAG** (Guo et al., 2024), **PathRAG** (Chen et al., 2026), **KG^{2}RAG** (Zhu et al., 2025), **HippoRAG2** (Gutiérrez et al., 2025), and **RAPTOR** (Sarthi et al., 2024). HippoRAG2 uses $Filter$ by default, and KG^{2}RAG also enables its filtering module, implemented as a reranker. Finally, to evaluate the modularity of SG^{2}RAG, we compare it with representative iterative retrieval methods, including **GeAR** (Shen et al., 2025), **iterDRAG** (Yue et al., 2025), **iterRetGen** (Shao et al., 2023), and **RT-RAG** (Shi et al., 2026).

[1] A closed-source embedding model sourced from https://bailian.console.aliyun.com/

Table 1: Performance of retrieval methods under Qwen-72B and Llama-70B backbone settings across three multi-hop datasets. The **bold** values indicate the best performance among the main methods, and underlined values indicate the second-best performance. Arrows on SG^{2}RAG indicate whether applying $Filter$ improves performance over its unfiltered variant.

Methods	Musique			2Wiki			HotpotQA			Average		
	EM	F1	Recall	EM	F1	Recall	EM	F1	Recall	EM	F1	Recall
Qwen-72B as LLM Backbone												
No Retrieval	5.10	13.57	-	26.50	30.81	-	25.60	34.64	-	19.07	26.34	-
Dense retriever												
BGE-M3	23.80	34.11	68.92	44.20	50.77	75.23	52.90	66.19	93.85	40.30	50.36	79.33
text-embedding-v3	25.50	36.30	64.06	42.90	49.67	85.48	50.90	64.15	92.15	39.77	50.04	80.56
Qwen3-Emb	27.20	38.91	69.76	45.10	51.63	84.95	53.90	67.08	93.85	42.07	52.54	82.85
Qwen3-Emb w/ $Reranker$	28.20	39.95	70.79	48.00	54.31	77.68	57.00	69.87	95.25	44.40	54.71	81.24
Structure-Augmented RAG												
LightRAG (EMNLP 2025)	15.72	25.03	33.16	38.69	48.55	75.03	42.30	57.41	76.39	32.24	43.66	61.53
PathRAG (AAAI 2026)	27.10	35.98	-	46.00	52.11	-	51.10	64.02	-	41.40	50.70	-
KG^{2}RAG (NAACL 2025)	27.20	38.24	62.63	53.70	61.91	**91.88**	56.70	70.33	95.64	45.87	56.83	83.38
RAPTOR (ICLR 2024)	26.60	38.64	68.68	46.70	53.66	74.42	54.40	67.40	91.85	42.57	53.23	78.32
HippoRAG2 (ICML 2025)	28.70	40.72	**74.02**	51.10	58.49	87.20	55.40	68.75	**96.35**	45.07	55.99	85.86
SG^{2}RAG w/o $Filter$ (Ours)	**30.00**	40.79	70.57	53.10	61.47	90.62	56.30	69.33	95.20	46.47	57.20	85.46
SG^{2}RAG (Ours)	29.40↓	**41.38**↑	71.63↑	**55.00**↑	**62.92**↑	90.55↓	**57.30**↑	**70.56**↑	96.10↑	**47.23**↑	**58.29**↑	**86.09**↑
Llama-70B as LLM Backbone												
No Retrieval	6.90	16.67	-	29.20	35.34	-	29.70	40.63	-	21.93	30.88	-
Dense retriever												
BGE-M3	26.20	38.56	68.92	45.00	52.15	75.23	53.00	66.21	93.85	41.40	52.31	79.33
text-embedding-v3	26.50	39.58	64.06	55.20	63.12	85.48	53.30	67.52	92.15	45.00	56.74	80.56
Qwen3-Emb	27.03	40.40	69.76	56.00	64.53	84.95	54.90	69.24	93.85	45.98	58.06	82.85
Qwen3-Emb w/ $Reranker$	29.60	43.03	71.74	52.60	60.37	77.70	57.00	70.75	95.20	46.40	58.05	81.55
Structure-Augmented RAG												
LightRAG (EMNLP 2025)	14.46	23.18	28.48	34.10	40.37	55.65	33.03	42.34	37.50	27.20	35.30	40.54
PathRAG (AAAI 2026)	25.10	33.56	-	44.80	50.48	-	55.30	67.48	-	41.73	50.51	-
RAPTOR (ICLR 2024)	27.70	41.35	69.18	46.70	53.66	74.45	54.80	69.21	91.75	43.07	54.74	78.46
KG^{2}RAG (NAACL 2025)	27.40	41.18	63.78	60.40	68.37	**92.26**	56.40	70.84	94.85	48.07	60.13	83.63
HippoRAG2 (ICML 2025)	29.60	43.24	**73.69**	57.10	65.18	84.07	56.40	70.76	**96.40**	47.70	59.73	84.72
SG^{2}RAG w/o $Filter$ (Ours)	29.40	43.28	70.69	60.80	68.91	90.95	56.40	71.44	95.45	48.87	61.21	85.70
SG^{2}RAG (Ours)	**32.10**↑	**45.22**↑	72.18↑	**61.00**↑	**69.08**↑	90.73↓	**57.70**↑	**71.74**↑	96.10↑	**50.26**↑	**62.01**↑	**86.34**↑

5.3 Implementation Details

We evaluate two unified LLM backbone settings using Qwen-72B and Llama-70B, with Qwen3-Emb as the shared embedding model. Within each setting, the same LLM is used for all applicable LLM-dependent stages, including graph construction, semantic-gain scoring, reranking or filtering, and answer generation. For the main comparison in Table 1, SG^{2}RAG and HippoRAG2 use their LLM-based $Filter$ to ensure a fair comparison under the same filtered-retrieval setting. KG^{2}RAG retains its built-in reranker, which is required by its original pipeline. We also report the unfiltered SG^{2}RAG variant to isolate the performance of semantic-gain-guided propagation. All subsequent ablation experiments disable $Filter$ so that changes can be attributed to graph weighting and propagation rather than additional LLM filtering. All methods report Exact Match (EM), F1, and Recall@10. EM and F1 measure final QA performance, while Recall@10 measures the coverage of annotated supporting documents. Detailed model configurations and hyperparameters are provided in Appendix A.

5.4 Main Results

Single Retrieval and Generation. Table 1 reports results under the Qwen-72B and Llama-70B backbone settings. Among the dense retrievers, Qwen3-Emb provides the strongest overall performance. Adding the matched LLM reranker improves its average EM and F1 by 2.33 and 2.17 points with Qwen-72B, while reducing Recall by 1.61 points. With Llama-70B, it yields only a 0.42-point EM gain, leaves F1 essentially unchanged, and reduces Recall by 1.30 points. This pattern exposes a structural limitation of reranking. It can only reorder documents already returned by dense retrieval and cannot reach missing support, which is particularly restrictive when multi-hop evidence has weak direct similarity to the query. SG^{2}RAG instead builds a reusable complementarity index offline. Query similarity locates entry seeds, and controlled PPR follows semantic-gain connections to rank graph-reachable evidence beyond the initial dense list. Even without $Filter$, SG^{2}RAG outperforms Qwen3-Emb with the matched reranker by 2.07, 2.49, and 4.22 points in average EM, F1, and Recall with Qwen-72B, with corresponding gains of 2.47, 3.16, and 4.15 points under Llama-70B. Because no additional query-time filtering is used, **this comparison dis-**

Table 2: Iterative retrieval results with Qwen-72B. "w/ SG^{2}RAG" replaces the per-iteration retriever while preserving the original framework.

Methods	Musique		2Wiki		HotpotQA		Average	
	EM	F1	EM	F1	EM	F1	EM	F1
GeAR (ACL 2025)	35.70	**51.49**	49.40	67.35	49.90	70.66	45.00	63.17
iterDRAG (ICLR 2025)	35.60	47.85	56.60	71.40	57.50	72.03	49.90	63.76
w/ SG^{2}RAG	**37.20** (+1.60 ↑)	51.33 (+3.48 ↑)	**64.70** (+8.10 ↑)	**73.11** (+1.71 ↑)	**61.60** (+4.10 ↑)	**74.66** (+2.63 ↑)	**54.50** (+4.60 ↑)	**66.37** (+2.61 ↑)
iterRetgen (EMNLP 2023)	30.80	43.36	48.70	55.77	55.40	68.70	44.97	55.94
w/ SG^{2}RAG	31.20 (+0.40 ↑)	43.14 (−0.22 ↓)	54.60 (+5.90 ↑)	62.86 (+7.09 ↑)	56.50 (+1.10 ↑)	69.39 (+0.69 ↑)	47.43 (+2.46 ↑)	58.46 (+2.52 ↑)
RT-RAG (WWW 2026)	27.20	40.77	55.20	69.43	52.10	59.09	44.83	56.43
w/ SG^{2}RAG	31.00 (+3.80 ↑)	44.70 (+3.93 ↑)	57.30 (+2.10 ↑)	70.96 (+1.53 ↑)	56.20 (+4.10 ↑)	64.23 (+5.14 ↑)	48.17 (+3.34 ↑)	59.96 (+3.53 ↑)

Table 3: Ablation study results. GS denotes the use of a greedy strategy on the semantic gain graph.

Methods	Musique			2Wiki			HotpotQA			Average		
	EM	F1	Recall	EM	F1	Recall	EM	F1	Recall	EM	F1	Recall
Naive	27.20	38.91	69.76	45.10	51.63	84.95	53.90	67.08	93.85	42.07	52.54	82.85
MMR	18.50	29.34	52.64	36.30	41.85	61.12	44.90	56.67	69.10	33.23	42.62	60.95
Sim w/ Controlled PPR	26.30	38.34	68.60	47.40	54.62	82.33	53.70	66.40	92.55	42.47	53.12	81.15
SG^2 w/ GS	27.20	38.64	68.50	52.10	59.98	85.68	55.30	68.52	93.85	44.87	55.71	82.68
SG^2 w/ PPR	28.10	38.99	70.22	50.70	58.68	89.95	53.70	66.77	94.85	44.17	54.81	85.00
SG^{2}RAG	**30.00**	**40.79**	**70.57**	**53.10**	**61.47**	**90.62**	**56.30**	**69.33**	**95.20**	**46.47**	**57.20**	**85.46**

tinguishes complementarity-indexed graph retrieval from LLM reranking. KG^{2}RAG and HippoRAG2 also outperform reranked dense retrieval on the average metrics, showing the value of graph propagation over a flat candidate list. SG^{2}RAG without $Filter$ further surpasses both graph baselines in average EM and F1 under both backbone settings while maintaining comparable or higher Recall. Together, **these results show that SG^{2}RAG builds a more effective retrieval graph by encoding document complementarity rather than connectivity alone, enabling controlled PPR to prioritize evidence chains not favored by query similarity or topology alone**. With $Filter$, SG^{2}RAG achieves the best average EM, F1, and Recall under both backbone settings, outperforming HippoRAG2 by 2.16, 2.30, and 0.23 points with Qwen-72B and by 2.56, 2.28, and 1.62 points with Llama-70B. It also obtains the highest F1 on all three datasets under both backbone settings, indicating that the improvement is consistent across the evaluated benchmarks.

Iterative Retrieval and Generation. As shown in Table 2, SG^{2}RAG can replace the single-retrieval component of an iterative framework while preserving its overall reasoning procedure. Because iterative frameworks compute Recall differently, we use EM and F1 under the same generation setting to indicate whether retrieval reaches answer-bearing documents. This replacement improves average EM by 4.60, 2.46, and 3.34 points for iterDRAG, iterRetGen, and RT-RAG, respectively, with corresponding F1 gains of 2.61, 2.52, and 3.53 points. The SG^{2}RAG-enhanced iterDRAG also outperforms GeAR by 9.50 points in average EM and 3.20 points in average F1. These consistent gains show that the quality of the underlying retriever remains crucial beyond query decomposition and multi-round interaction. Controlled propagation over the semantic gain graph retrieves evidence with higher complementary value, making **SG^{2}RAG a flexible plug-and-play retriever for iterative multi-hop reasoning**.

5.5 Ablation Study

We conduct ablation studies to determine whether the gains arise from the complementarity-indexed graph, the propagation strategy, or query anchoring. All ablations use Qwen-72B without $Filter$, so SG^{2}RAG in this subsection refers to the unfiltered configuration. Naive RAG is the same Qwen3-Emb dense baseline reported under Qwen-72B in Table 1, and its results are reproduced here. Table 3 compares it with MMR (Carbonell & Goldstein, 1998), Sim w/ Controlled PPR, SG^2 w/ GS, and SG^2 w/ PPR. Sim w/ Controlled PPR and SG^{2}RAG use identical Top-n candidate pairs and the same controlled PPR procedure. Their only difference is whether cosine similarity or semantic gain defines the edge weights. SG^2 w/ GS replaces controlled PPR with a two-hop greedy strategy. SG^2 w/ PPR uses P_q^{all}, which distributes mass over all graph nodes according to normalized query similarity rather than restricting it uniformly to the Top-h seeds. The full unfiltered configuration achieves the best performance across all datasets and metrics. It outperforms MMR by 13.24 EM, 14.58 F1, and 24.51 Recall points on average, showing that marginal-relevance selection does not effectively capture cross-document complementarity. Compared with Sim w/ Controlled PPR, SG^{2}RAG improves

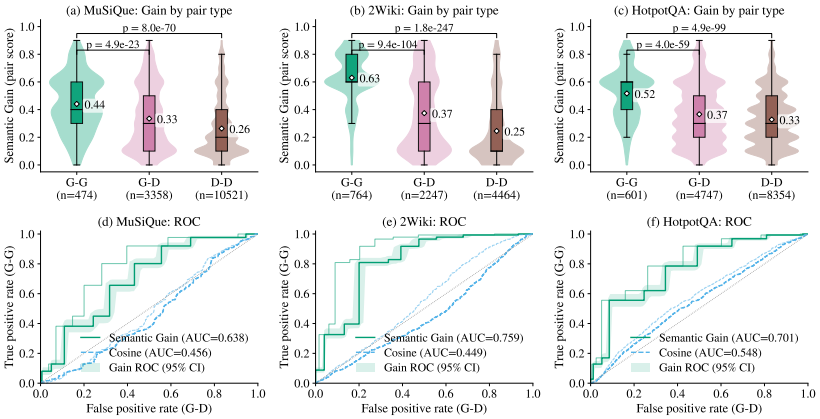


Figure 4: Cross-dataset validity of semantic gain. Panels (a)–(c) compare score distributions for Gold–Gold (G–G), Gold–Distractor (G–D), and Distractor–Distractor (D–D) pairs. Panels (d)–(f) compare semantic gain with cosine similarity in distinguishing G–G from G–D pairs.

average EM, F1, and Recall by 4.03, 4.08, and 4.31 points. Because candidate support and propagation are held fixed, **these gains isolate the value of encoding complementarity in the graph and cannot be explained by applying diffusion to a similarity graph alone**. Compared with SG^2 w/ GS, the full unfiltered configuration improves average EM, F1, and Recall by 1.63, 1.49, and 2.78 points, verifying that **controlled PPR composes local gain relations more effectively than greedy expansion**. Compared with SG^2 w/ PPR, SG^{2}RAG further improves average EM, F1, and Recall by 2.33, 2.39, and 0.46 points. This comparison shows that uniform restart mass on query-relevant Top-h seeds provides stronger topic anchoring than similarity-weighted mass over every node.

5.6 Robustness to Pair-Weight Aggregation

Figure 3 compares alternative treatments of the same candidate pairs using the Qwen-72B results of SG^{2}RAG. The default Pair Sum aggregates the available ordered assessments into symmetric pair strengths. Pair Mean, labeled Symmetric in the figure, averages those assessments instead and remains within 0.60 points of Pair Sum across all metrics. This small difference shows that retrieval is robust to how multiple assessments on a pair are aggregated. Directed instead interprets each ordered assessment as a directed transition arc. Reversed flips those arcs while preserving their weights, and Random Swap randomizes their assignment. These directed treatments produce larger reductions because Top-n proposal membership is not necessarily reciprocal. A missing reverse record represents an unobserved assessment, but directed PPR turns it into a hard one-way reachability constraint. The pair-level variants retain the observed conditional assessments in the connection strength while allowing traversal from either endpoint. The results therefore support the distinction between ordered evidence-contribution assessment and symmetric graph traversal.

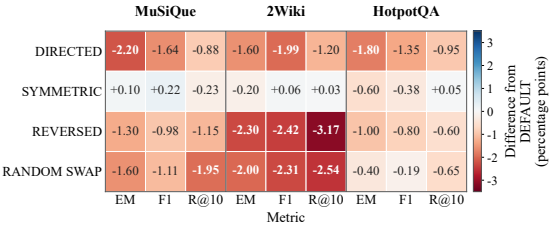


Figure 3: Effects of alternative edge-weight treatments on SG^{2}RAG. Values are percentage-point differences from the default pair-sum PPR reference.

5.7 Validity of Semantic Gain

We evaluate semantic gain across all three benchmarks using gold supporting annotations only for post-hoc analysis. A G–G record links two documents annotated as supporting evidence for the same query, while G–D and D–D records are defined by the endpoint annotations. This analysis uses the original ordered assessment w_{ij} before pair aggregation, so Pair Sum, Pair Mean, and directional transition treatments do not change the score analyzed here. Supporting-document membership

provides an empirical proxy for useful cross-document complementarity and is never used during graph construction. The upper panels of Figure 4 show the same G–G > G–D > D–D ordering across datasets. The respective mean scores are 0.44, 0.33, and 0.26 on MuSiQue, 0.63, 0.37, and 0.25 on 2Wiki, and 0.52, 0.37, and 0.33 on HotpotQA. All G–G differences from the comparison groups are statistically significant. This consistent hierarchy shows that the scorer assigns larger values to records whose endpoints contain more answer-supporting information. The lower panels treat G–G as positive and G–D as negative. Semantic gain obtains AUC values of 0.638, 0.759, and 0.701 on MuSiQue, 2Wiki, and HotpotQA, compared with 0.456, 0.449, and 0.548 for cosine similarity. Because both signals are evaluated on the same candidate records, **the improvements of 0.182, 0.310, and 0.153 show that semantic gain encodes an evidence-contribution relation not captured by similarity alone**. This validates the ordered assessments before pair aggregation and controlled PPR without treating semantic gain as a theoretically verified latent utility.

5.8 Effect of Other Settings

LLMs of different sizes. Figure 5(a) compares semantic-gain scoring with Qwen2.5 models at 7B, 14B, 32B, and 72B scales. SG^{2}RAG improves average F1 and Recall over Naive RAG at every scale. The F1 gain increases overall from 2.33 points at 7B to 4.66 points at 72B, while the Recall gain remains between 4.85 and 5.52 points. The consistent improvements from 7B onward demonstrate robustness across evaluator capacities, while the stronger gains at 72B show that additional model capacity can further improve performance.

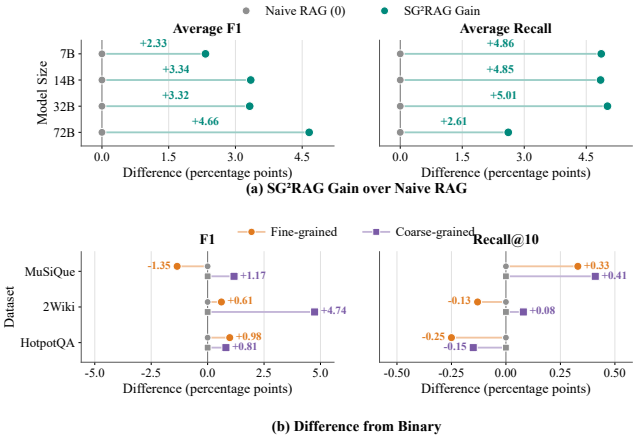


Figure 5: Effects of Other Settings. Panel (a) shows the average F1 and Recall gains of SG^{2}RAG over Naive RAG across Qwen2.5. Panel (b) shows the F1 and Recall@10 differences of Fine-grained and Coarse-grained semantic gain from Binary scoring.

Semantic Gain Granularity. Using Llama-70B, we compare Binary scoring with Fine-grained scores at two decimal places and the default Coarse-grained scores at one decimal place. Figure 5(b) reports their differences from Binary. Coarse-grained scoring improves F1 by 1.17, 4.74, and 0.81 points on MuSiQue, 2Wiki, and HotpotQA, respectively, while all Recall@10 changes remain within 0.41 points. Fine-grained scoring is less consistent, decreasing F1 by 1.35 points on MuSiQue despite modest gains on 2Wiki and HotpotQA.

Overall, these results show that SG^{2}RAG remains effective across evaluator scales and benefits from coarse-grained scoring that preserves useful distinctions without requiring excessive numerical precision. Additional sensitivity analyses of graph construction, seed selection, and retrieval size are provided in Appendix B.1.

6 Conclusion

This paper presents SG^{2}RAG, a complementarity-indexed retrieval framework that derives a reusable Semantic Gain Graph from ordered evidence-contribution assessments. Their aggregated pair strengths support robust traversal, while query-anchored controlled PPR composes local complementarity relations into multi-hop evidence rankings beyond the initial dense retrieval. Experiments across three multi-hop QA benchmarks and two LLM backbones show consistent improvements over strong dense reranking and graph-based RAG baselines. Controlled ablations and gold-support analyses validate the graph representation and routing design. Because queries are unknown during offline construction, the graph is fixed by similarity-based neighborhoods and cannot adapt to query-specific evidence needs. Future work will explore dynamic, query-aware graph construction.

AI USE STATEMENT

In this work, we used generative AI tools solely for document-oriented assistance, including language polishing, improving clarity and presentation. We did not use generative AI tools to develop the proposed method or generate experimental results. The authors reviewed all AI-assisted text. We take full responsibility for the final content of this work, including all text produced with the aid of generative AI.

ETHICS STATEMENT

This work does not raise any concerns related to the ICLR Code of Ethics or other applicable ethical guidelines.

REPRODUCIBILITY STATEMENT

We provide the source code in the supplementary materials to support the reproducibility of the proposed method.

REFERENCES

Jaime G Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, volume 98, pp. 290941–291025, 1998.

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.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024. URL `https://arxiv.org/abs/2402.03216`.

Changwook Chu, Yongtae Jeong, Hansam Cho, Jaehoon Kim, Jungmin Lee, Byungwoo Bang, Junyeon Lee, Uiseok Song, and Seoung Bum Kim. Cats-rag: Contextual augmented triplet synthesis for rag in technical qa. *Expert Systems with Applications*, pp. 131491, 2026.

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.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*, 2025.

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, 2020.

Tom Hosking, Hao Tang, and Mirella Lapata. Hierarchical indexing for retrieval-augmented opinion summarization. *Transactions of the Association for Computational Linguistics*, 12:1533–1555, 2024.

Marko Hostnik and Marko Robnik-Šikonja. Retrieval-augmented code completion for local projects using large language models. *Expert Systems with Applications*, 292:128596, 2025.

Rishabh Iyer, Ninad Khargoankar, Jeff Bilmes, and Himanshu Asanani. Submodular combinatorial information measures with applications in machine learning. In *Algorithmic learning theory*, pp. 722–754. PMLR, 2021.

Yuelyu Ji, Zhuochun Li, Rui Meng, Sonish Sivarajkumar, Yanshan Wang, Zeshui Yu, Hui Ji, Yushui Han, et al. Rag-rlrc-laysum at biolaysumm: Integrating retrieval-augmented generation and readability control for layman summarization of biomedical texts. In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pp. 810–817, 2024.

Zhuoxuan Jiang, Tianyang Zhang, Shengguang Bai, Lin Lin, Haotian Zhang, Yinong Xun, Jiawei Ren, Wen Si, and Shaohua Zhang. Towards enterprise-specific question-answering for it operations and maintenance based on retrieval-augmented generation mechanism. *Expert Systems with Applications*, pp. 130961, 2025.

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.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*, 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.

Shilong Li, Yancheng He, Hangyu Guo, Xingyuan Bu, Ge Bai, Jie Liu, Jiaheng Liu, Xingwei Qu, Yangguang Li, Wanli Ouyang, et al. Graphreader: Building graph-based agent to enhance long-context abilities of large language models. *arXiv preprint arXiv:2406.14550*, 2024.

Xiaopeng Li, Pengyue Jia, Derong Xu, Yi Wen, Yingyi Zhang, Wenlin Zhang, Wanyu Wang, Yichao Wang, Zhaocheng Du, Xiangyang Li, et al. A survey of personalization: From rag to agent. *ACM Transactions on Information Systems*, 2025.

Antonio Moreno-Cediel, Eva Garcia-Lopez, Antonio Garcia-Cabot, and David De-Fitero-Dominguez. Optimising retrieval performance in rag systems: A new growing window semantic chunking strategy to address weak semantic boundaries. *Knowledge-Based Systems*, pp. 114896, 2025.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford infolab, 1999.

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.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9248–9274, 2023.

Zhili Shen, Chenxin Diao, Pavlos Vougiouklis, Pascual Merita, Shriram Piramanayagam, Enting Chen, Damien Graux, Andre Melo, Ruofei Lai, Zeren Jiang, et al. Gear: Graph-enhanced agent for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 12049–12072, 2025.

Yuling Shi, Maolin Sun, Zijun Liu, Mo Yang, Yixiong Fang, Tianran Sun, and Xiaodong Gu. Reasoning in trees: Improving retrieval-augmented generation for multi-hop question answering. *arXiv preprint arXiv:2601.11255*, 2026.

Eunji Song and Hwansoo Lee. Defending financial rag systems against jailbreak attacks. *Expert Systems With Applications*, 2026.

Wenyu Tao, Xiaofen Xing, Zeliang Li, and Xiangmin Xu. Saki-rag: Mitigating context fragmentation in long-document rag via sentence-level attention knowledge integration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1195–1213, 2025.

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.

Aditi Vidyarthi, Motilal Khoirom Singh, and Dennis Singh Moirangthem. Sagerag: Query rewriting for retrieval enhancement and retrieval-augmented generation for grounded responses in ai research assistance. *Expert Systems with Applications*, pp. 131160, 2026.

Jing Wang, Zhiwei Xu, Siyuan Liu, Ye Lu, and Tao Li. Towards efficient graph-rag via structure-aware intermediate representation: Incremental collaborative exploration on knowledge graph. *Knowledge-Based Systems*, pp. 115884, 2026.

Qikai Wei, Huansheng Ning, Chunlong Han, and Jianguo Ding. A query-aware multi-path knowledge graph fusion approach for enhancing retrieval-augmented generation in large language models. *Knowledge-Based Systems*, 2025.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

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.

Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. Inference scaling for long-context retrieval augmented generation. *arXiv preprint arXiv:2410.04343*, 2025.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. 2025. URL https://arxiv.org/abs/2506.05176.

Jihao Zhao, Zhiyuan Ji, Yuchen Feng, Pengnian Qi, Simin Niu, Bo Tang, Feiyu Xiong, and Zhiyu Li. Meta-chunking: Learning text segmentation and semantic completion via logical perception. *arXiv preprint arXiv:2410.12788*, 2024.

Jihao Zhao, Zhiyuan Ji, Zhaoxin Fan, Hanyu Wang, Simin Niu, Bo Tang, Feiyu Xiong, and Zhiyu Li. Moc: Mixtures of text chunking learners for retrieval-augmented generation system. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5172–5189, 2025.

Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. Knowledge graph-guided retrieval augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8912–8924, 2025.

Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. Linearrag: Linear graph retrieval augmented generation on large-scale corpora. *arXiv preprint arXiv:2510.10114*, 2025.

A Appendix: Detailed Implementation

A.1 Baseline Details

For pure LLM baselines, no external knowledge base is used during inference. For standard RAG baselines, each dense retriever retrieves relevant chunks from the same corpus, and the retrieved contexts are then fed into the generator LLM. For structure-augmented baselines, LightRAG constructs an entity-relation graph for lightweight graph-based retrieval, HippoRAG2 introduces text nodes into the entity graph to constrain graph traversal, and RAPTOR builds a hierarchical tree through clustering and recursive summarization.

To ensure a fair comparison, we uniformly adopt zero-shot prompting for final answer generation across all methods. Although HippoRAG2 originally uses a few-shot prompting setup, we remove the few-shot examples in our experiments to avoid introducing additional prompt-based advantages. In addition, both HippoRAG2 and SG^{2}RAG support an LLM-based $Filter$. We enable the filter for their main comparison to maintain a consistent filtered-retrieval setting and report the unfiltered SG^{2}RAG variant separately. All subsequent ablation experiments disable the filter so that the effects of graph construction and propagation can be evaluated without additional LLM filtering.

For the iterative retrieval comparison, SG^{2}RAG is integrated into an iterative question-answering framework to evaluate its compatibility with multi-round reasoning. GeAR performs LLM-guided multi-round graph traversal, iterDRAG decomposes complex questions into subquestions for iterative retrieval, iterRetGen incorporates explicit reasoning into subquestion generation and refinement, and RT-RAG builds a multi-level iterative reasoning process. Since GeAR's multi-round interaction is tightly coupled with its own graph structure, we do not include it in the combination strategy for iterative expansion.

A.2 LLM and Embedding Models settings

In this paper, all LLMs use a temperature coefficient of 0, with the maximum model output length set to 8192 tokens and the model's context length opened up to 32648 tokens. For the Embedding models, the context length is only opened up to 8192, the encoding format adopts the float type, and the embedding dimension of Qwen3-Emb is 4096, while that of the other models is 1024.

A.3 Hyperparameters of Structure-Augmented RAG

SG^{2}RAG. We keep all hyperparameters consistent across datasets and LLM backbones, except in parameter sensitivity analyses. Specifically, the number of semantic gain edges (*top-n*) and query-relevant seed nodes (*top-h*) are both fixed at 5. The graph propagation coefficient in controlled Personalized PageRank is fixed at $\alpha = 0.5$, following the commonly used configuration. During offline graph construction, semantic gain scores lie in $[0, 1]$ and are interpreted using four rubric-defined intervals, namely Redundant (0.0–0.2), Minor (0.3–0.5), Significant (0.6–0.8), and Novel (0.9–1.0). Each scored proposal is retained as an ordered assessment record attached to an undirected node pair. When both proposal orders are evaluated, they remain as parallel records. The default Pair Sum treatment aggregates their weights into the symmetric pair strength used by PPR. The main comparison enables the optional LLM-based $Filter$, while all subsequent ablation experiments disable it.

RAPTOR, constructs a hierarchical semantic tree by recursively clustering and abstracting textual units: starting from fine-grained leaf nodes (e.g., sentences or paragraphs), semantically similar nodes are iteratively grouped based on embedding similarity, and a LLM generates abstractive summaries to form parent nodes, resulting in a multi-level tree that enables retrieval at varying granularities. In our implementation, the tree depth is set to 5, and a pruning threshold of 0.5 is applied to the cosine similarity between node embeddings, only pairs exceeding this threshold are merged, thereby balancing abstraction fidelity and retrieval precision.

LightRAG, is a graph-based RAG method that constructs a knowledge graph by extracting entities from documents and linking them based on co-occurrence and contextual relationships to capture structured semantic dependencies. For LightRAG, we adopt the "local" query model setting, which

restricts retrieval to the immediate neighborhood of relevant entities in the graph, thereby focusing on contextually coherent and relationally grounded evidence.

PathRAG retrieves relational paths and applies path-level pruning to reduce irrelevant and redundant evidence. We initialize its path selection with 20 candidate paths, after which its internal filtering procedure retains 10 paths for final question answering. Because PathRAG returns paths rather than a ranked list of individual documents, document-level Recall@10 is not applicable and is therefore not reported.

KG^{2}RAG expands semantically retrieved seed nodes under knowledge graph guidance and organizes the resulting evidence for answer generation. We configure it with 20 initial seed nodes. After its internal ranking and reranking procedures, the method returns 10 documents as the final context for question answering.

HippoRAG2, is a retrieval framework that structures knowledge into a graph of phrase nodes, enabling associative search across concepts. For HippoRAG2, the number of documents initially retrieved is set to 100, with each node forming up to 10 links. The PPR restart probability α is fixed at 0.5.

MMR, first retrieves the top 50 query-similar candidates and greedily selects $k = 10$ documents. At each step, each remaining document d is scored by $\text{score}(d) = \lambda \,\text{sim}(q, d) - (1 - \lambda) \max_{d_j \in \mathcal{S}} \text{sim}(d, d_j)$, where $\mathcal{S}$ denotes the set of already selected documents and λ balances query relevance against redundancy, defualt value of 0.5.

For experimental settings not explicitly specified in the baseline methods, we adhere to their default implementations. Furthermore, apart from the final answer prompt, which is kept identical across all methods, the prompts used for constructing knowledge graphs in each structured-augmented RAG variant also follow their respective defaults.

A.4 Hyperparameters of Iterative RAG

In all iterative comparative experiments, we uniformly set the maximum number of interaction rounds to 3, with all other hyperparameters strictly following the default configurations of their respective official open source repositories. For the three architectures iterDRAG, iterRetGen, and RT-RAG, this study directly replaced their original underlying retrieval modules with the proposed SG^{2}RAG framework. Furthermore, to ensure fairness and rigor in the performance evaluation, we strictly aligned the number of text nodes recalled in a single retrieval after the module replacement, thereby completely eliminating any additional performance interference caused by differences in retrieval capacity.

A.5 Prompts

During the reproduction and evaluation of the baseline methods, we strictly maintained the consistency of the prompts used in the underlying construction phase across all comparative frameworks. Considering the significant structural differences among the default prompt templates utilized by various baseline models in the final answer generation stage, we uniformly standardized the prompt engineering for the final question answering phase across all benchmarks. This ensures absolute fairness in the performance evaluation and completely eliminates any additional interference caused by different generation paradigms. The specific template details are fully illustrated in Figure A.1. Furthermore, for the unique heuristic semantic gain evaluation mechanism of the SG^{2}RAG framework, its specially customized evaluation prompt instructions are displayed in Figure A.2.

A.6 Source Code

The source code accompanying this submission is included in the supplementary materials and will be released publicly when permitted by the venue policy. The submitted implementation provides complete prompt templates and additional details omitted from the paper due to space constraints, including the prompt used by the filtering module and the examples omitted from the semantic gain evaluation prompt.

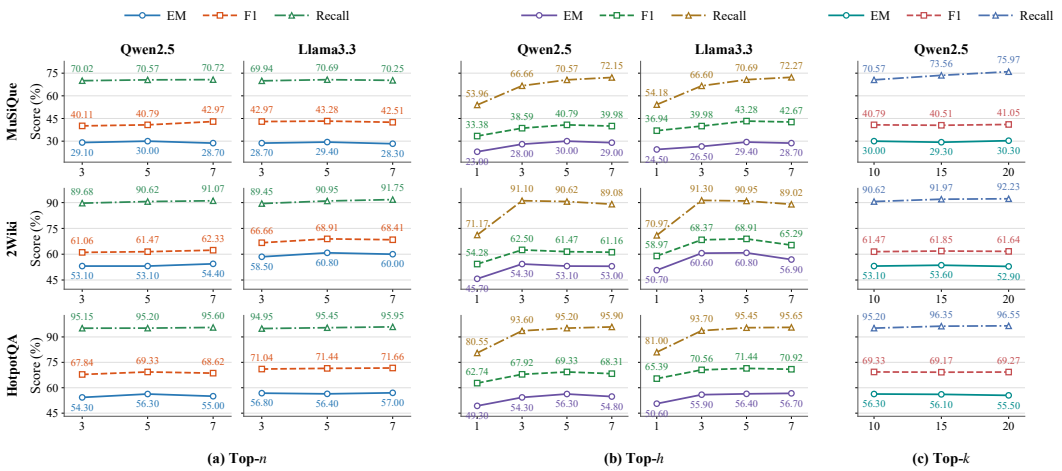


Figure 6: Hyperparameter analysis. Panels (A), (B), and (C) vary the graph construction neighborhood size top-n, the controlled PPR seed nodes top-h, and the final retrieval size top-k, respectively.

B Appendix: Additional Experimental

B.1 Hyperparameter Sensitivity Analysis

We conduct a post-hoc sensitivity analysis of three key hyperparameters rather than tuning them for individual datasets or LLM backbones. All main results use top-$n = 5$ for the graph construction neighborhood, top-$h = 5$ query-relevant seed nodes for controlled PPR, and top-$k = 10$ retrieved chunks. These parameters determine the candidate scope for semantic-gain scoring, the initial support for graph propagation, and the final context capacity, respectively.

Number of Semantic Gain Edges (top-n). top-n controls graph density and the candidate scope available to each node. As shown in Figure 6, performance remains stable around top-$n = 5$ under both Qwen-72B and Llama-70B. A smaller neighborhood can produce an overly sparse graph and omit useful transitions, while a larger neighborhood yields limited performance changes but increases offline construction cost. This trend suggests that a moderate local neighborhood provides an effective balance between candidate coverage and scoring cost. Appendix B.2 further analyzes gold-pair coverage and multi-hop connectivity under different neighborhood sizes.

Number of Seed Nodes (top-h). Using only one seed substantially lowers all metrics because it provides overly narrow initial query coverage. Increasing top-h to 3 or 5 yields large gains by supplying multiple query-relevant entry points. Expanding it to 7 produces mixed returns. Recall increases slightly on MuSiQue and HotpotQA but decreases on 2Wiki, while EM and F1 generally plateau or decline. These results support top-$h = 5$ as a stable balance between initial coverage, query anchoring, and complementary exploration across datasets and backbones.

Number of Retrieved Chunks (top-k). Increasing top-k generally broadens evidence coverage, although its effect varies across datasets. A smaller retrieval size is often sufficient on HotpotQA, where reasoning paths are relatively concentrated, while MuSiQue benefits more clearly from a larger context. These results indicate that the appropriate retrieval capacity depends on the structure and dispersion of supporting evidence.

B.2 Candidate Coverage and Routing Stability

Figure 7 examines how the candidate neighborhood size affects gold-pair reachability. Increasing top-n consistently improves both one-hop coverage and two-hop connectivity across all three datasets. On MuSiQue, for example, one-hop coverage increases from 16.3% at top-$n = 3$ to 22.5% at top-$n = 7$. The effective-edge markers further show that most covered Gold–Gold candidates receive positive gain. Yet Figure 6(A) shows only modest performance changes over the same range.

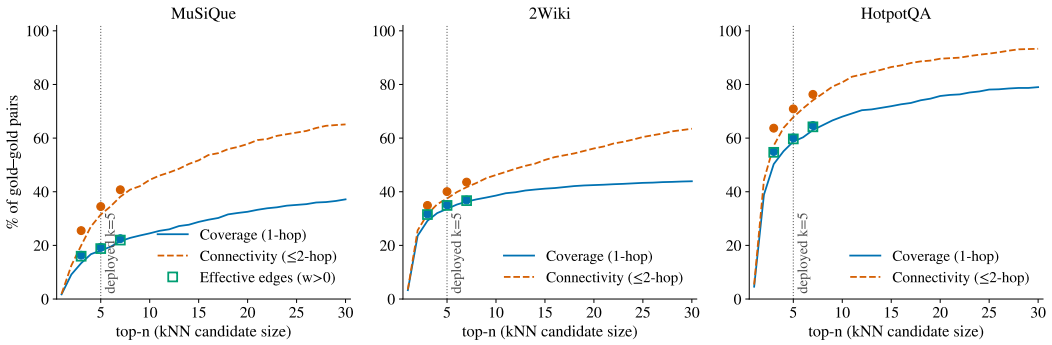


Figure 7: Gold-pair reachability as the *top-n* candidate neighborhood grows. Solid lines show one-hop coverage of Gold–Gold pairs, dashed lines show connectivity within two hops, and square markers show positive-gain Gold–Gold edges at the evaluated settings. Dotted lines mark the deployed setting *top-n* $= 5$.

Table 4: Cost and performance comparison.

Method	PT	CT	TT	Costs (USD)[a]	LRC	Index (s)	GPU hours	SS (MB)	Infer (s)	EM	F1
HippoRAG2	10211608	4261367	14472975	7.16	23312	10560.63	11.73	4188.16	1.17	26.40	<u>38.41</u>
KG^{2}RAG	6533479	1561937	8095416	3.18	12699	2728.84	3.03	1101.01	3.28	<u>27.20</u>	38.24
SG^{2}RAG	34865515	1945024	36810539	9.31	11656	4845.60	5.38	198	0.66	**30.00**	**40.79**

[a] API costs are estimated using GPT-5.6-Luna pricing.

Similarity and semantic gain play distinct roles. Query similarity selects the Top-h seeds, while document similarity defines each node's Top-n candidate transitions. Semantic gain then allocates transition probability within this support, and controlled PPR propagates through sequences of gain-weighted local pairs. Thus, increasing *top-n* expands where the walk can go without proportionally changing where most probability mass goes. These results suggest that the main high-gain routes are already available at small *top-n*, while additional candidates provide limited marginal routing utility.

Query-conditioned edge reweighting could better exploit larger or adaptive candidate neighborhoods by adjusting routing preferences to each query. However, it would replace query-independent offline weights with substantial online computation and graph maintenance overhead. Developing efficient dynamic routing is therefore an important direction for future work.

B.3 Costs Analysis

For this cost comparison, neither HippoRAG2 nor SG^{2}RAG uses *Filter* because LLM-based filtering introduces additional online requests and latency. KG^{2}RAG retains its built-in reranker because it is a required component of its original retrieval pipeline. Following the performance results in Section 5.4, we compare the computational overhead of these three methods. We evaluate all methods on MuSiQue, the largest and most challenging benchmark in our experiments, using Qwen-72B with an LLM batch size of 32 and an embedding batch size of 256. The reported metrics include prompt tokens (PT), completion tokens (CT), total tokens (TT), LLM request count (LRC), indexing time (Index), storage space (SS), inference time (Infer), EM, F1, estimated API cost in USD, and local deployment GPU hours [3]. API costs are calculated using the adopted GPT-5.6-Luna rates of \$0.20 per 1M prompt tokens and \$1.20 per 1M completion tokens.

As shown in Table 4, KG^{2}RAG has the lowest one-time offline construction cost. It consumes 8.10M tokens, has an estimated API cost of \$3.18, and requires 2728.84 seconds or 3.03 GPU hours for indexing. HippoRAG2 costs \$7.16, while SG^{2}RAG consumes 36.81M tokens, costs \$9.31, and requires 4845.60 seconds or 5.38 GPU hours. The higher offline cost of SG^{2}RAG mainly comes from repeatedly providing long system instructions and identical document contexts for document-level semantic-gain scoring. This expense is incurred once during graph construction, after which the

[3] https://huggingface.co/Qwen/Qwen2.5-72B-Instruct

complementarity index is reused without online semantic-gain scoring. Moreover, since SG^{2}RAG repeatedly queries the same instructions and document contents, the input tokens can benefit from prompt caching mechanisms. With cache hits, where the cached input tokens are charged at a much lower rate ($0.02 / 1M tokens), the actual input cost can be further reduced, making the offline construction overhead more economical in practical deployments.

The lower offline cost of KG^{2}RAG does not translate into a lower deployment cost. Its index occupies 1101.01 MB, compared with only 198 MB for SG^{2}RAG. More importantly, its complex tree-structured search and built-in reranker increase online inference time to 3.28 seconds, nearly five times the 0.66 seconds required by SG^{2}RAG. Unlike one-time graph construction, online inference is repeated for every query, so this cost accumulates continuously as the system is used. Consequently, the lifecycle deployment cost of KG^{2}RAG can become substantially higher than that of SG^{2}RAG as query volume grows.

SG^{2}RAG is also more efficient than HippoRAG2 for private local deployment. It requires 6.35 fewer GPU hours, reduces LLM requests by approximately 50%, uses less than 5% of the storage space, and lowers online inference time from 1.17 to 0.66 seconds. Although KG^{2}RAG and HippoRAG2 construct graph indexes, their graph relations primarily encode entity-level connectivity rather than document-level evidence contribution. The additional tree search and reranking in KG^{2}RAG increase computation but do not resolve this representation limitation. In contrast, SG^{2}RAG makes evidence contribution explicit in a reusable complementarity index and uses controlled PPR to prioritize answer-bearing evidence. It therefore achieves an EM of 30.00 and an F1 of 40.79, outperforming KG^{2}RAG at 27.20 and 38.24 and HippoRAG2 at 26.40 and 38.41.

In summary, KG^{2}RAG offers cheaper one-time offline construction, but its storage and recurring inference overhead make deployment more expensive. By amortizing complementarity indexing offline, SG^{2}RAG provides a better balance of answer quality, storage efficiency, local construction cost, and online latency.

B.4 Statistical Significance

Table 5: Paired permutation tests against HippoRAG2 and KG^{2}RAG using Llama-70B.

			vs. HippoRAG2				**vs. KG^{2}RAG**			
Dataset	**Metric**	**SG^{2}RAG**	**Score**	**Diff**	**p-value**	**Sig.**	**Score**	**Diff**	**p-value**	**Sig.**
2Wiki	F1	69.08	65.18	+3.90	0.0004	Yes	68.37	+0.71	0.5180	No
	EM	61.00	57.10	+3.90	0.0007	Yes	60.40	+0.60	0.6549	No
HotpotQA	F1	71.74	70.76	+0.98	0.1561	No	70.84	+0.90	0.2098	No
	EM	57.70	56.40	+1.30	0.1503	No	56.40	+1.30	0.1776	No
MuSiQue	F1	45.22	43.24	+1.98	0.0302	Yes	41.18	+4.04	0.0002	Yes
	EM	32.10	29.60	+2.50	0.0189	Yes	27.40	+4.70	0.0001	Yes

Using the Llama-70B results in Table 1, we perform paired permutation tests on the query-level EM and F1 outcomes rather than only comparing their aggregate means. The comparison with HippoRAG2 uses the $Filter$-enabled configurations of both methods, while KG^{2}RAG follows its main-table pipeline. For each dataset and metric, every query yields a paired outcome from SG^{2}RAG and the corresponding baseline. Under the null hypothesis of no systematic difference, we randomly swap the two method labels within each pair and recompute the mean difference across all 1,000 queries. The resulting null distribution is used to estimate the p-value. We use $\alpha = 0.05$ as the significance threshold.

Compared with HippoRAG2, SG^{2}RAG improves both F1 and EM by 3.90 points on 2Wiki, with $p = 0.0004$ and $p = 0.0007$. On MuSiQue, the improvements of 1.98 F1 and 2.50 EM are also significant, with $p = 0.0302$ and $p = 0.0189$. The HotpotQA differences remain positive at 0.98 F1 and 1.30 EM, but their p-values of 0.1561 and 0.1503 do not reach the significance threshold. These results show that the improvements over HippoRAG2 are supported at the query level on two of the three benchmarks.

Compared with KG^{2}RAG, SG^{2}RAG obtains smaller positive differences on 2Wiki and HotpotQA. The gains are 0.71 F1 and 0.60 EM on 2Wiki, with $p = 0.5180$ and $p = 0.6549$, and 0.90 F1 and 1.30 EM on HotpotQA, with $p = 0.2098$ and $p = 0.1776$. None of these differences are statistically

significant. In contrast, SG^{2}RAG improves F1 by 4.04 points and EM by 4.70 points on MuSiQue, with $p = 0.0002$ and $p = 0.0001$. Both improvements are significant and substantially larger than those observed on the other two datasets.

The strongest statistical evidence therefore appears on MuSiQue, the more challenging benchmark in our evaluation, where SG^{2}RAG significantly outperforms both HippoRAG2 and KG^{2}RAG in F1 and EM. This pattern suggests that propagation over locally complementary pairs provides a clearer advantage for more demanding multi-hop retrieval. The non-significant results on 2Wiki and HotpotQA against KG^{2}RAG do not imply the absence of an improvement, since all observed differences remain positive. Rather, the smaller effect sizes are not sufficient to rule out query-level variation at the selected threshold.

C Appendix: Operational Interpretation of Semantic Gain

Semantic gain is inspired by conditional gain in set functions (Iyer et al., 2021), but SG^{2}RAG does not define or learn a latent utility function over graph nodes. Instead, $\mathcal{M}_{sg}$ directly interprets each baseline-candidate pair and assigns a bounded score $w_{ij} \in [0, 1]$ under the fixed rubric $\mathcal{R}$. The rubric asks whether v_j contributes relevant, novel, and significant information beyond the baseline v_i. This definition is local to an ordered chunk pair and does not represent marginal gain conditioned on the complete retrieved set. Reciprocal scores may differ because the baseline and candidate roles are evaluated separately. Crucially, w_{ij} is an ordered assessment record rather than a mandatory directed PPR transition. SG^{2}RAG retains every observed assessment and aggregates the records of each node pair into a symmetric traversal strength. Query-specific seed membership is supplied separately by P_q^{ctrl}.

C.1 From Ordered Assessments to Pairwise Traversal Strength

For every connected unordered pair (v_i, v_j), let $\mathcal{E}_{ij}$ denote its available ordered assessment records. The default Pair Sum design follows the additive treatment of parallel records in the PPR implementation and defines the effective pair strength

$$a_{ij} = a_{ji} = \sum_{e \in \mathcal{E}_{ij}} w_e. \tag{5}$$

The graph contains one record when the pair is proposed by only one Top-n neighborhood and two parallel records when both ordered proposals are evaluated. In the latter case, $a_{ij} = w_{ij} + w_{ji}$, so both order-specific assessments contribute to the connection strength. On 2Wiki, 37.9% of connected pairs contain two records. Importantly, a missing reverse record means that the reverse order was not assessed. It does not imply zero contribution in that order because Top-n neighborhood membership need not be reciprocal. Using the records directly as directed arcs would therefore couple graph reachability to sparse proposal coverage. It could also prevent traversal of a useful connection when a future query selects the opposite endpoint as an entry node. Pair Sum avoids this hard reachability artifact without adding new document pairs.

To convert these pair weights into transition probabilities, let $d_i = \sum_{v_l \in \mathcal{N}_{\mathcal{G}}(v_i)} a_{il}$ denote the weighted degree of v_i. For $d_i > 0$, PPR uses

$$\mathcal{T}_{ji} = \frac{a_{ij}}{d_i}. \tag{6}$$

An isolated node is assigned a self-loop. This normalization makes each pair strength relative to the other connections of the current node. Consequently, a pair with greater aggregated semantic gain receives a larger transition probability from either endpoint. The ordered assessments therefore determine connection strength without imposing their observation direction as a reachability constraint.

Pair Sum assigns greater total weight to a pair when both proposal orders are present. To determine whether this multiplicity materially affects retrieval, we additionally evaluate Pair Mean in the robustness analysis:

$$\widetilde{a}_{ij} = \widetilde{a}_{ji} = \frac{1}{|\mathcal{E}_{ij}|} \sum_{e \in \mathcal{E}_{ij}} w_e. \tag{7}$$

Pair Mean preserves the average semantic-gain value of the available assessments while removing the additional weight associated with having two records. It therefore changes aggregation without changing candidate support or traversal symmetry. Section 5.6 shows that Pair Mean remains within 0.60 points of the default Pair Sum across all metrics. This small difference indicates that retrieval is robust to summing or averaging the available assessments. The larger reductions from directed treatments show that the main loss comes from imposing one-way reachability, supporting the separation of ordered assessment from symmetric traversal.

Incremental graph update. The undirected multigraph also enables lightweight graph maintenance. When a new chunk v_{new} arrives, it is matched to a small semantic neighborhood and each resulting ordered assessment is attached to the corresponding node pair. If a reciprocal assessment is later available, it is added as a parallel record and contributes to the same pair sum. Existing records among old nodes remain unchanged, and only the weighted degrees of nodes incident to the new records require updating. Compared with entity-centric graph updates that may require entity extraction, entity resolution, and relation reconstruction, this chunk-level insertion avoids reconsidering connections among existing nodes.

D Appendix: Case Study

MuSiQue Case When asking *"Who is the child of the Victim of Romance performer?"*, the query is first matched with documents based on semantic similarity, yielding query-relevant seed nodes. Among these, the document *"Victim of Romance\nVictim of Romance ..."* is most relevant to the query, as it directly mentions the target work. However, relying solely on similarity-based seed retrieval is insufficient, since the answer requires complementary evidence from another document. SG^{2}RAG addresses this issue by performing controlled PPR over the semantic gain graph, which discovers the document *"Wilson Phillips\nWilson Phillips is an ..."* as complementary evidence. As a result, the final input to the generative model contains both the performer information and the family-relation evidence needed for answering the question. Specifically, the document *"Victim of Romance\nVictim of Romance ..."* states that *"Victim of Romance"* was performed by *"Michelle Phillips"*, while the document *"Wilson Phillips\nWilson Phillips is an ..."* further indicates that *"Chynna Phillips"* is the daughter of *"John and Michelle Phillips"*. Details are shown in Fig. A.3.

HotpotQA Case When asked *"Who is the film dedicated to that Paramount Classics and MTV Films co-purchased the rights to?"*, the initial seed nodes primarily match prominent query entities such as *"Paramount Classics"* and *"MTV Films"*. However, these seed documents alone do not provide the complete reasoning chain required to identify the final answer. Through controlled PPR expansion over the semantic gain graph, SG^{2}RAG retrieves the passage *"Hustle & Flow ..."*, which serves as complementary evidence to the passage *"Beneath (2007 film) Beneath is a straight-to-DVD ..."*. Together, these documents provide the key bridge from the rights co-purchase information to the dedicated film, ultimately supporting the answer *"Sam Phillips"*. Details are shown in Fig. A.4.

2Wiki Case When requesting *"Who is Mugain's mother-in-law?"* as shown in Figure 3, the query first retrieves relevant seed documents through the key entity *"Mugain"*. From the seed document, we can identify that *"Mugain"* is the wife of *"Conchobar mac Nessa"*, who is described as the *"king of Ulster"*. However, answering the question requires an additional relation concerning Conchobar's family. SG^{2}RAG therefore performs controlled PPR over the semantic gain graph to retrieve complementary documents about *"Conchobar mac Nessa"*. These documents further reveal that his mother is *"Ness"*. Therefore, *"Mugain's mother-in-law"* is *"Ness"*. Details are shown in Fig. A.5.

Prompt Template:
Based on the following retrieved content, please answer the user's question.

User Question: {query}
Retrieved Content:
{pasg_content}

Instructions:

1. Analyze the content to find relevant information.
2. Provide only the final answer without any explanation or analysis process.
3. If multiple pieces contain relevant information, synthesize them appropriately.
4. Only the final answer, no other content. The answer may be entities such as names of people, organizations, places, relationships, times, dates, etc.

Answer (Only the final answer, no other content):

Figure A.1: Prompt template for question answering.

You are an expert analyst specializing in textual information assessment. Your task is to meticulously evaluate the "semantic gain" of a "Paragraphs to Evaluate" relative to a "Baseline Paragraph".

Primary Goal: The purpose of this evaluation is to identify paragraphs that add new, significant, and complementary information to the baseline, helping to build a more comprehensive understanding of the core topic.

Definition of semantic gain:
semantic gain is a score from 0.0 to 1.0 that quantifies how much new, relevant, and significant information the "Paragraph to Evaluate" provides beyond what is already present in the "Baseline Paragraph". Relevance is determined by how closely the new information relates to the main subject and key entities of the baseline.

Evaluation Rubric (Score from 0.0 to 1.0):

- **0.0 - 0.2 (Redundant):** The paragraph largely repeats information already in the baseline, or adds only trivial, irrelevant details (e.g., rephrasing, common knowledge filler).
- **0.3 - 0.5 (Minor):** The paragraph adds some new, relevant information, but it is of low significance. It might be a minor detail, a specific example of a point already made, or a slight expansion of an existing concept.
- **0.6 - 0.8 (Significant):** The paragraph introduces substantial new facts, perspectives, or details that are highly relevant and complementary to the baseline. This new information significantly enriches the understanding of the core topic.
- **0.9 - 1.0 (Novel):** The paragraph provides entirely new information that is critical and highly relevant to the baseline's topic. It might introduce a new dimension, a surprising counter-argument, or key data that fundamentally changes or completes the picture presented in the baseline.

Expected JSON output for example:
{"data": [{"chunk_id": "doc_3_78ee7a4c1f82", "info_gain": 0.2}, {"chunk_id": "doc_6_6953a7af7c5b", "info_gain": 0.9}]}

(Note: Some few-shot examples are omitted for brevity.)

Now your task:
You will receive a new baseline paragraph and new paragraphs to evaluate in the exact same format.

Instructions:

1. For each paragraph, compare with the baseline paragraph. In your internal reasoning, follow the same explanation style as the example above, but DO NOT include explanations in the final output. Output only the JSON with `chunk_id` and `info_gain`. Keep <chunk_id> exactly as provided. Do not change order.
2. Values must be numbers with one decimal place.
3. The JSON should contain the `chunk_id` and `info_gain` of each evaluated paragraph, with no omissions allowed.

Input:
Baseline paragraph: {baseline}
Paragraphs to evaluate: {chunks_text}

Output (Only JSON):

Figure A.2: System prompt for evaluating semantic gain of retrieved paragraphs. Due to space limitations, the omitted few-shot examples are available in the provided source code.

SG²RAG

Query: Who is the child of the Victim of Romance performer? Answer: Chynna Phillips

Seed Nodes

Victim of Romance\n***Victim of Romance is singer and songwriter Michelle Phillips***' first and only solo album, and was released in February 1977 (see 1977 in music). The record was unsuccessful and Phillips (previously with The Mamas & the Papas) then favored her acting career. The front cover photography was by Terry O'Neill.

The Last Song\nAt seventeen, Veronica ``Ronnie "Miller (Miley Cyrus) remains as rebellious as she was the day her parents divorced and her father moved to North Carolina three years prior. Once a classical piano child prodigy under the tutelage of her father, Steve Miller (Greg Kinnear), Ronnie now ignores the instrument and has not spoken with her father since he left. While Juilliard School has been interested in her since she was young, Ronnie refuses to attend.

Louis Chedid\nLouis Chedid is **the son of** the writer Andrée Chedid and the father of Matthieu Chedid (better known as -M-).

American Idol\nPhillips became the winner, beating Sanchez. Prior to the announcement of the winner, season five finalist Ace Young proposed marriage to season three runner-up Diana DeGarmo on stage – which she accepted.

Lina Medina\nMedina has never revealed the father of the child nor the circumstances of her impregnation. Escomel suggested that she might not actually know herself, as she \"couldn't give precise responses\". Lina's father was arrested on suspicion of child sexual abuse, but he was released due to lack of evidence and the biological father was never identified. **Her son grew up healthy**. He died in 1979 at the age of 40. In young adulthood, Medina worked as a secretary in the Lima clinic of Lozada, which gave her an education and helped put her son through high school. She married Raúl Jurado, who fathered her second son in 1972. As of 2002, they lived in a poor district of Lima known as \"Chicago Chico\". She refused an interview with Reuters that year, just as she had turned away many reporters in years past.

PPR to Expand

The Official Story\nThe film deals with the story of an upper middle class couple who lives in Buenos Aires with an illegally adopted child. The mother comes to realize that her daughter may be the child of a desaparecido, a victim of the forced disappearances that occurred during Argentina's last military dictatorship (1976 - 1983), which was marred by widespread human rights violations and a genocide.

The Last Song\nKelly Preston as Kim Miller, Ronnie and Jonah's mother who raised her children in New York City after her divorce.

Wilson Phillips\nWilson Phillips is an American vocal group consisting of Carnie Wilson, Wendy Wilson, and ***Chynna Phillips, the daughters***, respectively, of Brian Wilson of The Beach Boys and ***of John and Michelle Phillips*** of The Mamas & the Papas.

Matthieu Chedid\nMatthieu Chedid was born in Boulogne-Billancourt, Hauts-de-Seine, France, the son of French singer Louis Chedid, and the grandson of the Egyptian-born French writer and poet of Lebanese descent Andrée Chedid who has written lyrics for him. His sister is the music video and concert director Émilie Chedid.

Je dis aime\nJe dis aime (1999) is the second studio album by French singer-songwriter Matthieu Chedid, in his persona as -M-, described by reviewers as a \"conceptual icon to rival Bowie's Ziggy Stardust and Aladdin Sane\". The album manages to take a remarkable variety of musical directions and pull them together into a consistent whole. Another reviewer describes the album as sounding like a 'French Lenny Kravitz' and notes the 'vintage 70s sound and textures'.

Figure A.3: Details of Musique case study. It details the selection of seed documents (Seed Nodes) for SG²RAG in Musique and the documents obtained through the random walk expansion of controlled PPR (PPR to Expand). Among them, documents with the same background color are indicated as complementary documents, even if they are not retrieved during seed selection, they can be obtained through PPR expansion to supplement multi-hop questions. In the document nodes, **black bold text** indicates fields with a relatively large contribution to similarity, while ***red italic bold text*** indicates fields that complement each other in terms of facts.



Figure A.4: Details of HotpotQA case study.

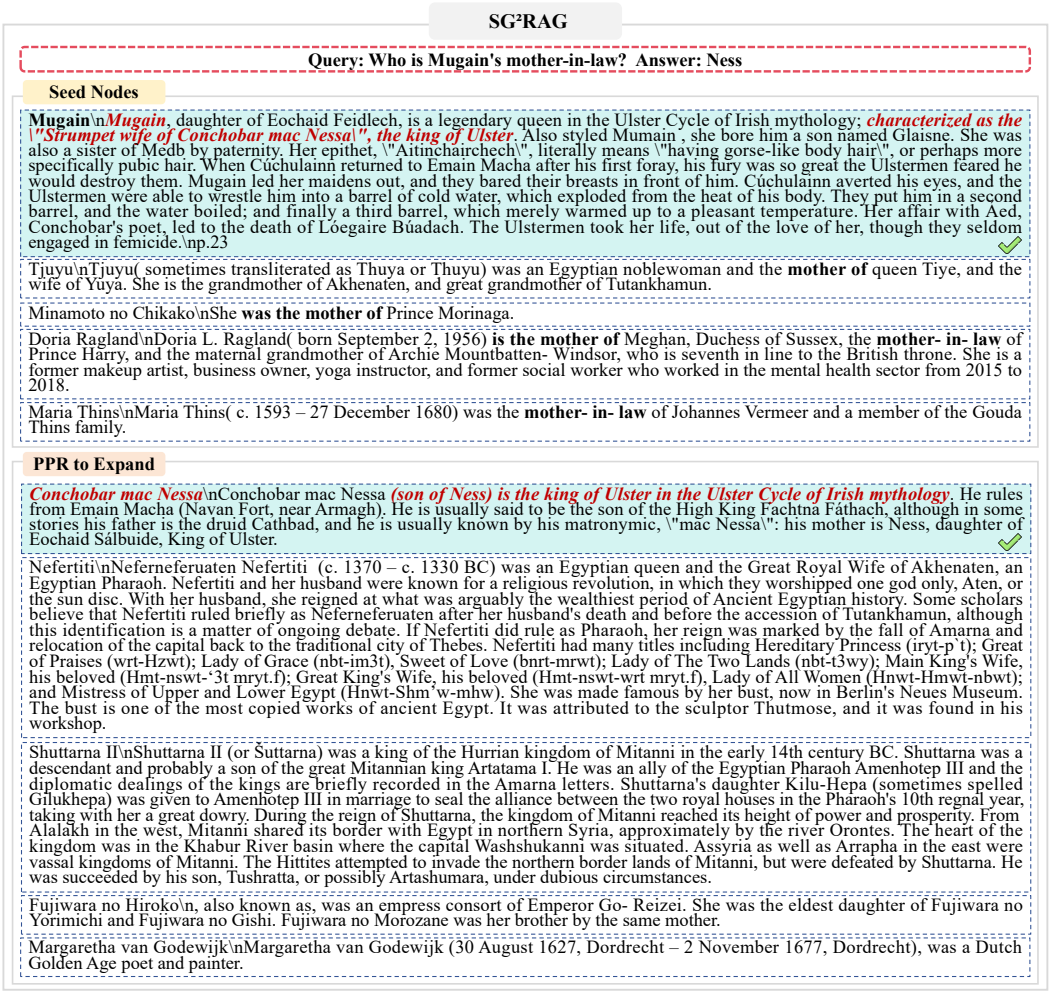


Figure A.5: Details of 2Wiki case study.