SAG: SQL-Retrieval Augmented Generation with Query-Time Dynamic Hyperedges

Anonymous authors
Paper under double-blind review

Abstract

While retrieval-augmented generation (RAG) has proven effective at giving LLMs access to external knowledge, mainstream dense-retrieval implementations remain inherently limited in handling structured constraints and multi-hop reasoning. Graph-based methods address this by constructing knowledge graphs offline, but they often fragment semantics, incur high maintenance, and complicate incremental updates. We propose SAG (SQL-Retrieval Augmented Generation), a structured retrieval architecture that organizes documents into an event-entity index without building a global knowledge graph. SAG represents each chunk as a semantically complete event paired with its entities, forming a latent hyperedge that preserves n-ary relations without decomposing them into triples. At query time, SAG treats shared entities as join keys to connect related chunks. This dynamically yields a query-scoped neighborhood of events, and yet every piece of evidence remains the original chunk throughout. Experiments on HotpotQA, 2WikiMultiHopQA, and MuSiQue show that SAG achieves the best retrieval and end-to-end QA performance on all three benchmarks. On MuSiQue, where multi-hop evidence chaining is most demanding, SAG reaches 80.36% Recall@5, outperforming the strongest baseline by 11.52 points. This work paves the way for knowledge infrastructure that enables LLM agents to retrieve and reason over continually growing organizational knowledge.[1]

1 Introduction

What an agent can do is bounded by what it can retrieve. Organizational knowledge is distributed across documents, messages, customer systems, and code; much of it is unstructured, relational, and continuously accumulating. Making this knowledge available to agents is therefore an infrastructure problem as well as a modeling problem. Multi-hop question answering makes this gap concrete. Answers often require evidence from multiple documents, whereas conventional RAG pipelines (Lewis et al., 2020; Karpukhin et al., 2020) rank passages independently by their similarity to the query. This approach works well when a single passage contains the answer, but it does not explicitly represent associations across passages. In multi-hop retrieval, failing to recover one intermediate passage can break the entire evidence chain (Yang et al., 2018; Trivedi et al., 2022; Mavi et al., 2024). The challenge is therefore not only to improve single-shot recall, but also to organize evidence so that cross-document associations can be recovered reliably.

Structure-augmented methods address this problem by constructing knowledge graphs or hierarchical representations offline (Edge et al., 2024; Gutiérrez et al., 2025). When n-ary events are decomposed into independently indexed triples, however, their shared event boundary may be lost. Hypergraph-based methods (Luo et al., 2025) preserve higher-order relations but still encode them in a persistent corpus-level structure. These structures support graph traversal and ranking, but the connections available during retrieval are determined before the query arrives, while ingestion and maintenance remain separate from query execution. This raises a different question: can we activate only the structure needed by a query during retrieval, without constructing a global knowledge graph?

Our central claim is that multi-hop retrieval need not rely exclusively on either dense similarity or a globally constructed graph (Figure 1). The required associative structure is latent in the events

[1] Benchmark and evaluation code: `https://anonymous.4open.science/r/sag-benchmark-0CC4`

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.

described by passages and the entities shared across them. SAG exposes this structure through an event–entity index. Each chunk is independently represented as one semantically complete event linked to a set of indexing entities. Because an event binds all its entities within one retrieval unit, it defines a latent hyperedge that retains the event's n-ary relation rather than decomposing it into independent triples. At query time, SQL joins over shared entities activate a local neighborhood of related events. This structural path is combined with dense retrieval, after which an LLM selects evidence from the compressed candidate set. SAG constructs no global corpus graph, and selected events are mapped back to their original chunks before answer generation. Under unified evaluation settings, SAG achieves the best retrieval performance on HotpotQA, 2WikiMultiHopQA, and MuSiQue, and the best end-to-end QA performance on all four benchmarks including NarrativeQA. Its largest retrieval gain occurs on MuSiQue, where SAG reaches 80.36% Recall@5 and exceeds the strongest baseline by 11.52 points.

To summarize, our contributions are fourfold. (1) We introduce SAG, a retrieval architecture that activates query-specific latent hyperedges over an event–entity index through SQL joins, without constructing a global knowledge graph. (2) We formalize the event–entity representation, characterize the neighborhoods recovered through incidence joins, and compare its corpus-level connectivity with four structure-augmented baselines. (3) We evaluate SAG on three multi-hop benchmarks and NarrativeQA under a unified configuration, ablate representation, expansion, and final selection across all three multi-hop benchmarks, and examine candidate budgets on MuSiQue. (4) We evaluate robustness to embedding replacement and continual corpus growth, and show how bounded candidate activation limits the downstream SQL and LLM workloads. Together, these properties make SAG a step toward knowledge infrastructure for agents operating over continually growing corpora.

2 Related Work

Retrieval-Augmented Generation. RAG combines a parametric generator with a dense passage index to incorporate external knowledge (Lewis et al., 2020). Later work adapts when and how retrieval is performed. Self-RAG uses reflection tokens to decide when to retrieve and to evaluate

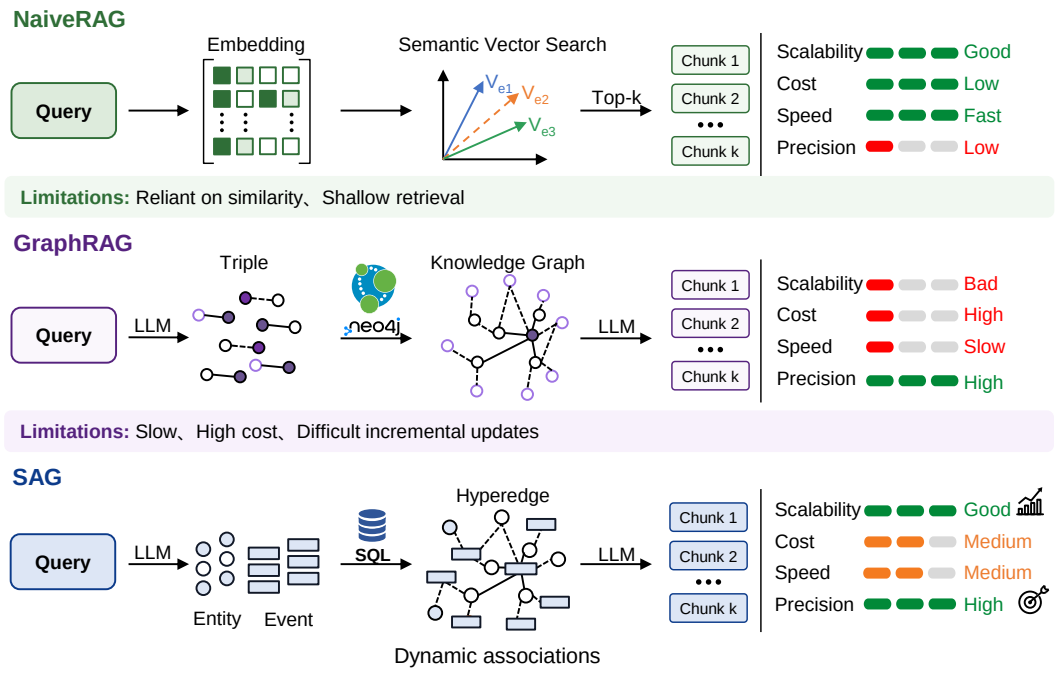


Figure 1: Three RAG paradigms. NaiveRAG retrieves top-k chunks by dense similarity, while GraphRAG organizes evidence in an offline knowledge graph. SAG stores event–entity relations in an append-only index and activates query-specific latent hyperedges through SQL joins.

retrieved evidence (Asai et al., 2024), while FLARE retrieves from predicted upcoming sentences when generation contains low-confidence tokens (Jiang et al., 2023b). Adaptive-RAG routes queries among no-retrieval, single-step, and iterative strategies according to predicted complexity (Jeong et al., 2024). IRCoT instead alternates retrieval with chain-of-thought generation, using each reasoning step to guide the next retrieval (Trivedi et al., 2023). These methods modify retrieval control or query formulation, whereas SAG changes how corpus knowledge is represented and connected.

Structure-Augmented Retrieval. GraphRAG constructs an entity graph and community summaries for answering global corpus-level questions (Edge et al., 2024). HippoRAG builds a knowledge graph from extracted relations and propagates relevance with personalized PageRank, while HippoRAG 2 integrates passages more directly and strengthens the online use of an LLM (Gutiérrez et al., 2024; 2025). LightRAG combines incremental graph updates with low- and high-level retrieval over entities and relations (Guo et al., 2025). SiReRAG constructs similarity and relatedness trees and flattens their nodes into a unified retrieval pool (Zhang et al., 2025), whereas StructRAG selects and constructs a task-appropriate structure at inference time (Li et al., 2025). These methods either maintain corpus-level structures or reconstruct structured views for individual queries. SAG instead stores event–entity incidence records and activates query-specific associations through SQL joins without materializing a global graph. ChatDB and StructGPT query existing databases or structured sources, but do not organize unstructured corpora for retrieval (Hu et al., 2023; Jiang et al., 2023a).

Hyperedge and Higher-Order Representations. Hypergraphs and n-ary knowledge-base models represent relations among multiple entities without reducing each relation to independent binary triples (Zhou et al., 2006; Feng et al., 2019; Fatemi et al., 2020; Galkin et al., 2020; Liu et al., 2020; 2021). HyperGraphRAG extracts n-ary facts and stores a corpus-level hypergraph with vector indexes for entities and hyperedges (Luo et al., 2025). Hyper-RAG retrieves both entities and correlations from a hypergraph repository and expands them through structural diffusion (Feng et al., 2026). HGRAG treats entities as nodes and passages as hyperedges, combining entity- and passage-level similarity through hypergraph diffusion (Wang et al., 2026). Graph-R1 constructs a lightweight knowledge hypergraph and learns multi-turn retrieval through reinforcement learning (Luo et al., 2026). These methods materialize a corpus-level hypergraph. SAG instead stores each event and its entities as relational incidence rows. SQL joins over shared entities activate only the latent hyperedges required by the query, after which the original chunks are returned to the reader.

3 SAG

SAG has an offline indexing phase and an online phase with seed retrieval, query-time expansion, and final selection (Figure 2). SQL performs exact entity joins, vector retrieval finds semantically relevant seed chunks even when the query and the passage share no exact terms, and the LLM is reserved for event-entity extraction, query-entity identification, and final selection over the compressed candidate set.

3.1 Event-Entity Index

SAG builds an event-entity index rather than a global knowledge graph. Each chunk is processed independently into one event e and a set of entities $V(e)$, which together define a latent hyperedge. New documents can therefore be appended without recomputing existing records.

Event. An event is a concise statement of the chunk's core content, with one event extracted per chunk. Decomposing a chunk into independent triples can fragment its meaning, especially when several entities jointly constrain the underlying fact. SAG instead keeps the event intact: retrieval operates on the concise event text, while the corresponding original chunk is returned as evidence.

Entities. Entities span 11 semantic types.[2] They are not treated as self-contained evidence; instead, they serve primarily as indexing and expansion keys that connect events sharing the same participant or concept. Events and entities are extracted in parallel by a single LLM call per chunk. Their many-to-many associations are stored in SQL, while their textual representations are indexed for vector retrieval.

[2]The complete inventory is time, location, person, organization, group, subject, work, product, action, metric, and tags; see Appendix I.

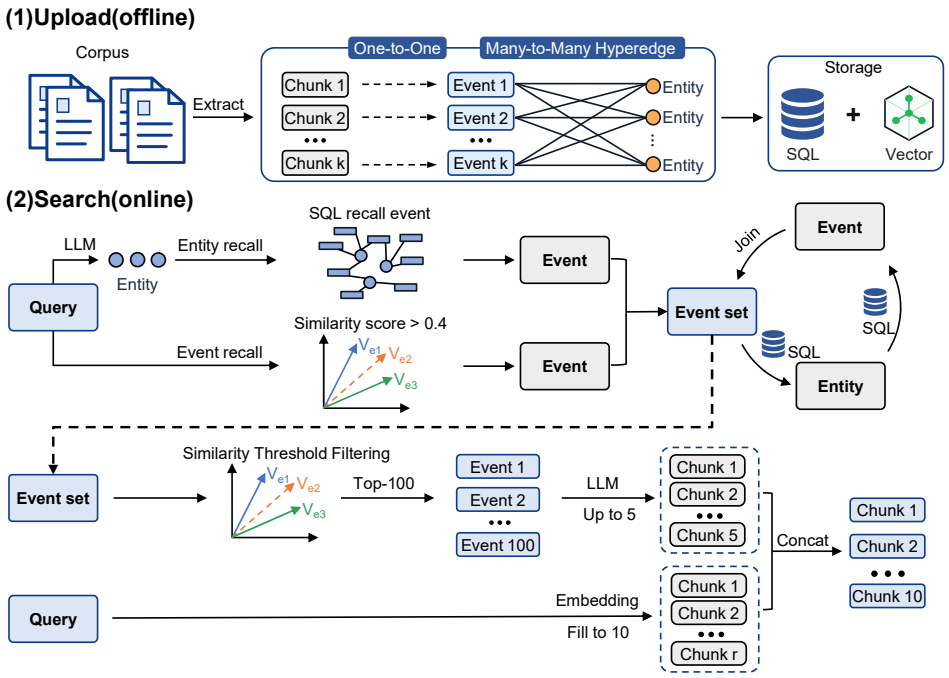


Figure 2: Architecture overview of SAG. Offline, chunks are indexed as events and entities across SQL and vector stores; online, initial recall is expanded at query time, and final selection operates over the compressed candidate set. The output panel illustrates the default fixed allocation: the structural path contributes exactly five event-linked chunks, and the semantic path contributes five non-overlapping chunks after overlap removal, yielding ten chunks in total.

The index deliberately avoids full entity disambiguation. It relies only on string normalization and SQL deduplication, preserving independent chunk processing and append-only ingestion. This tolerance for imperfect normalization is a deployment choice: production corpora arrive continuously and are rarely clean, and requiring global disambiguation or re-chunking before indexing would forfeit the append-only property that keeps ingestion practical as the corpus grows.

3.2 Latent Hyperedges

Let V denote the entity vertices and H the indexed events (hyperedges). Each event $h \in H$ defines a hyperedge over its entity set $V(h) \subseteq V$, keeping all entities mentioned in the event together as one retrieval unit. The event text preserves the relational meaning that a bare set of entity names cannot express, while the linked original chunk remains the evidence returned to the downstream reader.

The event-entity table stores each hyperedge as a set of rows (h, v) for $v \in V(h)$. Grouping these rows by event identifier recovers the complete entity set of the event, making the representation lossless with respect to incidence. Appendix A provides the complete formalization of the incidence representation and its query-time activation.

One expansion round (Section 3.4) walks from seed events to their entities and then back to co-incident events. The activated structure exists only for the current query, and ingesting a new chunk requires only appending new rows rather than rebuilding a global graph. Corpus statistics support this structural interpretation, with SAG showing less fragmentation than GraphRAG, LightRAG, and HyperGraphRAG on MuSiQue and more than 93% giant-component coverage on every benchmark (Appendix B). Section 5.2 further shows that hyperedges outperform triple decomposition in Recall@5.

3.3 Seed Retrieval

Given a query q, SAG first defines a query-specific working corpus D_q. In the full-pool configuration used for the main results, $D_q = D$. Under bounded activation, embedding retrieval first selects the top K_{pool} chunks from the global corpus, and all subsequent event and entity operations are restricted to records associated with those chunks. This corpus-level activation budget is distinct from K_{cand}, which controls the number of events passed to the LLM selector.

Within D_q, SAG constructs an initial event set R through two parallel retrieval paths, each addressing a failure mode of the other.

Path A: entity-guided structured recall. A lightweight LLM call identifies key entities in the query and produces a seed set $Q = \{v_i\}$. Similarity search over the entity vector index expands these seeds into an augmented set $\widehat{Q} \supseteq Q$ using $\text{Sim}_{\text{ent}} : V \times V \to [0, 1]$. Because entities are short strings, SAG uses a high similarity threshold $\tau_{\text{ent}} = 0.9$ by default; looser matching is more likely to conflate distinct entities than to recover genuine aliases. A SQL join then retrieves all events sharing at least one entity with $\widehat{Q}$:

$$R_{\text{ent}} = \left\{ h \in H \;\middle|\; V(h) \cap \widehat{Q} \neq \varnothing \right\}. \tag{1}$$

Path B: direct event recall via query vector. In parallel, SAG retrieves events directly by embedding similarity to the query using $\text{Sim}_{\text{evt}} : H \times \{q\} \to [0, 1]$. Because events are longer semantic statements, this path uses a deliberately low threshold $\tau_{\text{evt}} = 0.4$ by default, to favor recall and leave precision control to later ranking. The retrieved events form R_{evt}.

The two paths are merged to form the seed event set:

$$R = R_{\text{ent}} \cup R_{\text{evt}}. \tag{2}$$

3.4 Expansion and Selection

Expansion. Seed retrieval finds events that either share entities with the query or have high direct semantic similarity. Multi-hop questions, however, may require evidence connected through intermediate entities that do not appear in the query.

Starting from R, SAG uses a reverse SQL join to collect the entities associated with the seed events. Previously unexplored entities form an expansion frontier, which is pruned to a fixed budget before the next join to bound both computation and candidate growth. SAG then retrieves previously unseen events associated with the retained frontier entities.

Each event-to-entity-to-event composition constitutes one expansion round. Expansion runs for at most L rounds, with $L = 1$ by default, and each round introduces only entities and events not visited earlier. Let $X \subseteq H$ denote the events added through expansion. The complete candidate pool is

$$C = R \cup X, \tag{3}$$

Coarse ranking. SAG ranks the events in C by embedding similarity to the query and retains the top K_{cand} events:

$$\widehat{C} = \text{TopK}_{K_{\text{cand}}}(C, q). \tag{4}$$

Dual-path output. The structural path preserves evidence recovered through cross-entity expansion, which may have low direct query similarity despite being necessary for multi-step reasoning. The semantic path retains chunks that are directly relevant to the query but may not participate in the expanded structure.

The system returns $K_{\text{out}} = 10$ chunks. In the structural path, the LLM reads $\widehat{C}$ in context and selects exactly $K_{\text{event}} = 5$ events that jointly entail the answer, producing E_{sel} with $|E_{\text{sel}}| = K_{\text{event}}$. These events are mapped to chunks as $D_{\text{evt}} = \{\phi(h) : h \in E_{\text{sel}}\}$ through the event-to-chunk mapping $\phi : H \to D$. The LLM selector evaluates the candidate events jointly and selects a subset based on their relevance to the query and their relations to one another. The semantic path produces a ranked chunk list R_{chunk} by direct retrieval over the chunk index. After removing chunks already in D_{evt}, it fills exactly $K_{\text{direct}} = K_{\text{out}} - K_{\text{event}}$ slots. Under the default configuration, the structural and semantic paths therefore contribute five non-overlapping chunks each:

$$D_{\text{out}} = D_{\text{evt}} \cup \text{TopK}_{K_{\text{direct}}}\left(R_{\text{chunk}} \setminus D_{\text{evt}}\right). \tag{5}$$

Appendix C presents successful and unsuccessful activation traces.

4 Experiments

4.1 Datasets and Metrics

We evaluate on four QA benchmarks. MuSiQue (Trivedi et al., 2022) (answerable subset) includes questions requiring up to four reasoning hops. 2WikiMultiHopQA (Ho et al., 2020) tests inference across multiple documents, while HotpotQA (Yang et al., 2018) requires two-hop reasoning over encyclopedic passages. NarrativeQA (Kočiský et al., 2018) tests reasoning across characters and events in long narratives drawn from books and movie scripts.

We directly reuse HippoRAG 2's evaluation subsets and retrieval corpora (Gutiérrez et al., 2025). MuSiQue, 2WikiMultiHopQA, and HotpotQA each contain 1,000 development questions, with 11,656, 6,119, and 9,811 passages, respectively. NarrativeQA contains 293 queries over 10 documents, split into 4,111 passages. For the corpus-growth analysis, we use MuSiQue and the predominantly single-hop NQ subset derived from Wang et al. (2024) (1,000 queries; 9,633 passages). We hold the first 250 queries from each dataset fixed as its retrieval corpus grows.

Our primary retrieval metric is Recall@K, the fraction of gold supporting passages recovered among the top K retrieved passages, averaged across queries. We report Recall@5 in the main results and Recall@1/2/5/10 in Appendix D. For QA, we report EM and F1, together with four LLM-judge metrics following Xiang et al. (2025): answer correctness, coverage, context relevancy, and evidence recall (Appendix H). Retrieval metrics and evidence recall cover the three multi-hop benchmarks; the remaining metrics also cover NarrativeQA, which lacks passage-level gold evidence in this setup.

4.2 Baselines and Implementation

The baseline suite spans three families. Simple retrievers include BM25 and Contriever. Large embedding models include BGE-Large-EN-v1.5 (Xiao et al., 2024), GTE-Qwen2-7B (Li et al., 2023), GritLM-7B (Muennighoff et al., 2025), and NV-Embed-v2 (Lee et al., 2025). Structure-augmented methods include GraphRAG (Edge et al., 2024), LightRAG (Guo et al., 2025), HippoRAG 2 (Gutiérrez et al., 2025), HyperGraphRAG (Luo et al., 2025), and HyperRAG (Feng et al., 2026). We evaluate the complete suite on all four datasets. All structure-augmented methods use BGE-Large-EN-v1.5 for retrieval and Qwen3.6-Flash (Qwen Team, 2025) for QA, keeping the models fixed across comparisons. SAG and HippoRAG 2 provide raw chunks; the other structure-augmented methods retain their native structured contexts. SAG's retrieval-stage LLM calls use temperature 0.7 and top-p 1.0; QA and LLM judging use temperature 0.0, with top-p 1.0 for judging.

SAG stores its index in MySQL and Elasticsearch, with dense-vector search handled by Elasticsearch rather than a separate vector database. Unless otherwise noted, we set expansion depth $L = 1$, seed budget $K_{\text{seed}} = 50$, frontier budget 50, and candidate budget $K_{\text{cand}} = 100$. SAG returns $K_{\text{out}} = 10$ chunks per query: $K_{\text{event}} = 5$ from structural selection and $K_{\text{direct}} = 5$ non-overlapping chunks from direct retrieval.

5 Results and Analysis

5.1 Main Results

Retrieval gains are largest on MuSiQue. At Recall@5, SAG exceeds the strongest baseline by 1.25 points on HotpotQA, 2.99 points on 2WikiMultiHopQA, and 11.52 points on MuSiQue. SAG also ranks first or ties for first at all four reported cutoffs (Recall@1/2/5/10; Appendix D). The largest gain occurs on MuSiQue, where counterfactual filtering reduces single-hop shortcuts and increases the need to connect evidence across passages.

QA gains extend across multi-hop and narrative reasoning. With the reader fixed across all methods, SAG achieves the highest F1 on all four benchmarks (Table 1 and Appendix E). On the three multi-hop datasets, SAG outperforms the strongest F1 baseline by 6.37 points on MuSiQue, 1.17 points on 2WikiMultiHopQA, and 1.66 points on HotpotQA. NarrativeQA provides a qualitatively

Table 1: Main results under a unified evaluation setup. Except for HippoRAG 2, the structure-augmented RAG baselines do not directly produce passage retrieval results. The R@5 Avg. covers the three multi-hop datasets (1,000 queries each); the F1 Avg. is query-weighted across these datasets and NarrativeQA (293 queries). Under bootstrap resampling, † indicates $p < 0.05$ against the strongest baseline for the corresponding metric. **Best** and second-best results are highlighted.

Method	Multi-Hop QA						Discourse Understanding		
	MuSiQue		2Wiki		HotpotQA		NarrativeQA	Avg.	
	R@5	F1	R@5	F1	R@5	F1	F1	R@5	F1
Simple Baselines									
BM25 (Robertson & Walker, 1994)	42.11	35.19	63.78	56.26	73.05	64.64	16.42	59.65	48.86
Contriever (Izacard et al., 2022)	45.32	34.39	59.25	50.62	74.85	64.06	18.93	59.81	46.95
Large Embedding Models									
BGE-Large-EN-v1.5 (Xiao et al., 2024)	56.32	42.80	69.83	61.49	89.20	74.81	20.66	71.78	56.24
GTE-Qwen2-7B (Li et al., 2023)	61.92	44.60	74.18	64.91	88.45	73.04	19.71	74.85	57.20
GritLM-7B (Muennighoff et al., 2025)	66.22	49.30	75.62	66.12	93.40	73.22	26.23	78.41	59.62
NV-Embed-v2 (Lee et al., 2025)	68.84	52.88	77.00	68.45	95.25	78.00	25.59	80.36	62.81
Structure-Augmented RAG									
GraphRAG (Edge et al., 2024)	–	54.14	–	72.08	–	76.29	27.24	–	63.92
LightRAG (Guo et al., 2025)	–	46.66	–	65.75	–	73.09	24.23	–	58.49
HyperGraphRAG (Luo et al., 2025)	–	54.78	–	73.06	–	76.74	25.71	–	64.41
HyperRAG (Feng et al., 2026)	–	51.80	–	76.89	–	77.19	25.95	–	64.84
HippoRAG 2 (Gutiérrez et al., 2025)	65.13	47.49	90.35	74.34	94.35	77.24	23.49	83.28	62.54
SAG (Ours)	**80.36**†	**61.15**†	**93.34**†	**78.06**†	**96.50**†	**79.66**†	**31.90**†	**90.07**	**69.30**

different evaluation setting: it requires discourse-level reasoning over long narratives rather than following the explicit evidence chains characteristic of multi-hop QA benchmarks. SAG achieves 31.90 F1, outperforming the strongest baseline, GraphRAG, by 4.66 points. Together, these results suggest that SAG's QA gains are not confined to benchmark-specific multi-hop retrieval patterns, but extend to reasoning over events and temporal dependencies in long-form narratives.

LLM-based evaluation supports the QA findings. Using Qwen3.7-Plus as the judge, SAG ranks first in answer correctness, coverage, and evidence recall on each of the three multi-hop benchmarks (Table 9), and first in answer correctness (35.22), coverage (38.02), and context relevancy (71.93) on NarrativeQA. Context relevancy is the one metric where a baseline leads on the multi-hop benchmarks, and judge inputs differ there: several baselines supply entity and relation summaries alongside the retrieved passages, whereas SAG and HippoRAG 2 supply only raw chunks. Because context relevancy rates whether the supplied context suffices to answer the question, these summaries can affect the score independently of passage retrieval quality. SAG scores within 1.63 to 5.97 points of the best context-relevancy result. Together with the retrieval results, SAG's leading evidence recall, answer correctness, and coverage show that it recovers more supporting evidence and produces more accurate and complete answers.

5.2 Analysis

Component ablations. All three component ablations reduce recall on every benchmark (Table 2). Replacing hyperedge indexing with triples lowers Recall@5 by 2.75–13.80 percentage points. Removing expansion causes losses of 10.95 points on MuSiQue and 14.07 on 2WikiMultiHopQA, compared with 1.20 on HotpotQA, indicating that the benefit of expansion varies across datasets. Replacing the LLM selector with Qwen3-Reranker-8B lowers Recall@5 by 3.20–15.29 points. Together, these results support the contributions of event-level indexing, structural expansion, and LLM selection across all three benchmarks.

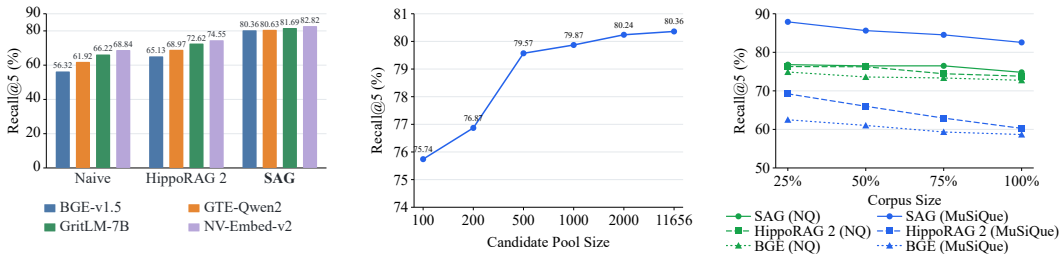
Parameter sensitivity. The parameter sweeps indicate that shallow expansion captured most of the observed retrieval benefit (Appendix F). Increasing expansion depth beyond one hop improved Recall@5 by at most 0.35 points. Increasing the candidate budget from 50 to 100 added 1.01 points, whereas increasing it from 100 to 500 added 1.88 points. We therefore use one-hop expansion and $K_{\text{cand}} = 100$ as the default setting for the main experiments; the cost analysis below evaluates $K_{\text{cand}} = 50$ as a lower-cost alternative. Structural retrieval alone achieved 79.80% Recall@5, compared with 56.23% for semantic retrieval alone. The equal allocation achieved the highest Recall@5 of 80.36%

Table 2: Component ablations on three multi-hop benchmarks. The full model uses hyperedge indexing, one expansion round ($L = 1$), and the Qwen3.6-Flash selector. Each variant changes only the indicated component. R@K denotes Recall@K (%); the best result in each column is bold.

		MuSiQue			2Wiki			HotpotQA		
Component	Configuration	R@1	R@2	R@5	R@1	R@2	R@5	R@1	R@2	R@5
Full model	SAG (Ours)	**36.82**	**63.62**	**80.36**	**43.81**	**83.93**	**93.34**	**47.80**	**91.55**	**96.50**
Indexing	w/ Triple indexing	35.66	61.83	77.61	37.60	71.80	84.10	38.15	72.75	82.70
Expansion	w/o Expansion ($L = 0$)	35.70	57.75	69.41	43.25	75.68	79.27	47.65	90.40	95.30
Final selection	w/ Qwen3-Reranker-8B	32.12	48.97	67.11	40.57	66.70	78.05	45.55	80.00	93.30

among the tested configurations, indicating that structural retrieval provided most of the performance and direct retrieval offered a modest complement.

SAG remains robust across embedding models. We next examine whether SAG's retrieval advantage depends on the choice of embedding model (Figure 3a). Replacing NV-Embed-v2 with BGE-v1.5 reduces HippoRAG 2 Recall@5 from 74.55% to 65.13%, a loss of 9.42 points, whereas SAG decreases from 82.82% to 80.36%, a loss of only 2.46 points. SAG with BGE-v1.5 still outperforms HippoRAG 2 with NV-Embed-v2. These results are consistent with the different propagation mechanisms of the two systems. Within SAG's activated neighborhood, SQL joins introduce connected events without repeatedly attenuating their relevance scores. HippoRAG 2 instead propagates embedding-derived relevance through personalized PageRank, whose damping can reduce the influence of evidence farther from the seed. The substantially smaller performance drop shows that SAG remains robust across embedding models, supporting the conclusion that its retrieval advantage arises primarily from the structured expansion mechanism rather than encoder strength.



(a) Embedding Models (b) Candidate Pool (c) Corpus Expansion

Figure 3: Robustness and scaling analyses. (a) Sensitivity to four embedding models on MuSiQue. (b) Recall@5 across candidate-pool sizes on MuSiQue. (c) Recall@5 under corpus growth on NQ and MuSiQue.

Cost–quality trade-off. Table 3 reports indexing and query costs on MuSiQue. SAG requires less indexing time and fewer tokens than GraphRAG, LightRAG, and HyperGraphRAG, while HippoRAG 2 remains cheaper to index. Online, SAG achieves higher F1 than the compared baselines but is not uniformly faster. Reducing K_{cand} from 100 to 50 lowers end-to-end latency from 25.03 to 22.68 seconds and token consumption from 11,536 to 8,236, while F1 decreases from 61.15 to 60.41. The smaller budget thus offers a lower-cost alternative to the default configuration. Appendix G provides stage-level cost comparisons and further analysis.

Scaling with fixed query-time budgets. SAG separates corpus-level vector search from query-local processing. HNSW retrieves the top K_{pool} chunks from the corpus; event–entity retrieval and expansion operate on their associated records, and the LLM selector receives at most K_{cand} events. On MuSiQue, $K_{\text{pool}} = 500$ achieves 79.57% Recall@5, compared with 80.36% for the full corpus of 11,656 chunks (Figure 3b). This retains approximately 99% of the full-pool score while activating less than 5% of the corpus. If K_{pool} and K_{cand} remain fixed as the corpus grows, HNSW searches the larger index while the SQL and LLM stages continue to operate on a bounded, query-specific working set.

Table 3: Runtime, tokens, and F1 on MuSiQue (11,656 chunks; 1,000 queries). Offline values are corpus totals; online values are per-query averages. End-to-end time sums retrieval and QA time; online tokens include inputs and outputs from both stages. F1 values for the baselines are from Table 7.

			Offline indexing		Online querying (per query)			
Method		F1	Time (h)	Tokens (M)	Retrieval (s)	QA (s)	End-to-end (s)	Tokens
Dense RAG (BGE)		42.80	0.06	0	0.10	1.75	1.85	1,727
GraphRAG		54.14	10.15	125.12	0.23	12.56	12.79	11,438
LightRAG		46.66	25.94	103.14	11.97	18.13	30.10	9,974
HyperGraphRAG		54.78	16.21	216.11	4.45	23.54	27.99	32,536
HippoRAG 2		47.49	2.30	15.93	0.66	8.50	9.16	4,788
SAG (Ours)	$K_{\text{cand}} = 50$	60.41	4.57	28.26	15.93	6.75	22.68	8,236
	$K_{\text{cand}} = 100$	**61.15**			18.67	6.36	25.03	11,536

SAG remains robust under continual corpus growth. In a separate corpus-growth analysis, we keep the evaluation queries fixed and track Recall@5 as the retrieval corpus expands from 25% to 100% on NQ and MuSiQue (Figure 3c). On the single-hop NQ benchmark, SAG decreases from 76.80% to 74.81% and remains the strongest method at every corpus size. On the more challenging multi-hop MuSiQue benchmark, SAG decreases from 87.90% to 82.57%, a loss of 5.33 points, whereas HippoRAG 2 falls from 69.23% to 60.27%, a loss of 8.96 points. As additional passages compete for the fixed retrieval budget, SAG incorporates them into its append-only event–entity index and uses query-local SQL joins with fixed seed and frontier budgets. The smaller decline on MuSiQue is consistent with this bounded retrieval design.

5.3 Limitations

SAG has two current limitations. First, it normalizes and deduplicates entity strings but does not resolve aliases. It therefore cannot recognize that "Apple Inc." and "Apple" refer to the same entity, and connections that depend on matching these surface forms may be lost. This affects recall when bridge entities appear under different names in different documents. A lightweight alias table could link common full names, abbreviations, and aliases without requiring full entity disambiguation or compromising incremental writes, though building and maintaining such a table introduces its own tradeoff between coverage and precision.

The second limitation concerns temporal updates. SAG's index is append-only by design, which supports continual ingestion but provides no mechanism for revising or deleting stale events. This is sufficient for a static retrieval corpus, but an agent memory must also represent facts, preferences, and task states that change over time. When a fact is updated, the old version remains indexed and may be retrieved alongside the new one, leaving the downstream reader to resolve the conflict. Versioned events, informed by bitemporal knowledge graphs (Rasmussen et al., 2025; Lairgi et al., 2026), are a natural next step toward supporting knowledge that can be updated while preserving its history.

6 Conclusion

SAG represents chunks through event–entity indexes and activates query-specific latent hyperedges through SQL joins, without constructing a global knowledge graph. Under unified evaluation settings, SAG achieves the best QA performance on three multi-hop benchmarks and NarrativeQA. Query-time expansion over shared entities, LLM-based contextual selection, and hyperedge preservation of event semantics each contribute measurably to retrieval quality. SAG remains robust across embedding models and degrades more slowly as the corpus grows. Fixed activation budgets bound the downstream SQL and LLM workloads, while append-only indexing supports continual writes without global graph reconstruction. Together, these properties position SAG as infrastructure for agents operating over continually growing knowledge.

Future work will examine larger-scale deployments and extend SAG with alias resolution and versioned updates toward a memory substrate for agents.

AI use statement

Large language models were used during the preparation of this manuscript for writing assistance, including grammar checking, sentence-level rephrasing, and LaTeX formatting. All scientific content, experimental design, data analysis, and technical claims were produced entirely by the authors. The authors take full responsibility for the correctness and integrity of all reported results.

Ethics statement

The experiments use public multi-hop QA datasets and collect no new human-subject data. Because SAG uses language models to extract and link entities, errors or biases may produce misleading associations, while deployment on private corpora may expose sensitive cross-document relationships. Such deployments should enforce data provenance, access control, and human review in high-impact settings.

Reproducibility statement

All experiments and results reported in the main paper and appendices can be reproduced using the benchmark code, evaluation scripts, and configurations available at `https://anonymous.4open.science/r/sag-benchmark-0CC4`. The paper and appendices specify the datasets, models, evaluation settings, ablations, costs, and prompts. Exact values involving hosted Qwen models may vary with provider updates and nondeterministic inference.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

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. URL https://arxiv.org/abs/2404.16130.

Bahare Fatemi, Perouz Taslakian, David Vazquez, and David Poole. Knowledge hypergraphs: Prediction beyond binary relations. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, 2020. URL https://arxiv.org/abs/1906.00137.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019. URL https://arxiv.org/abs/1809.09401.

Yifan Feng, Hao Hu, Shihui Ying, Xingliang Hou, Shiquan Liu, Mingyuan Yang, Junchang Li, Shaoyi Du, Nanning Zheng, Han Hu, and Yue Gao. Hyper-RAG: Combating LLM hallucinations using hypergraph-driven retrieval-augmented generation. *Nature Communications*, 2026. doi: 10.1038/s41467-026-71411-1. URL https://www.nature.com/articles/s41467-026-71411-1.

Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, and Jens Lehmann. Message passing for hyper-relational knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. URL https://arxiv.org/abs/2009.10847.

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

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2405.14831.

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

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 (COLING)*, 2020. URL https://aclanthology.org/2020.coling-main.580/.

Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. ChatDB: Augmenting LLMs with databases as their symbolic memory. *arXiv preprint arXiv:2306.03901*, 2023. URL https://arxiv.org/abs/2306.03901.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research (TMLR)*, 2022. URL https://arxiv.org/abs/2112.09118.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 7036–7050, 2024. URL https://aclanthology.org/2024.naacl-long.389/.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. StructGPT: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023a. URL https://arxiv.org/abs/2305.09645.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023b. URL https://aclanthology.org/2023.emnlp-main.495/.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. URL https://aclanthology.org/2020.emnlp-main.550/.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics (TACL)*, 6:317–328, 2018. URL https://aclanthology.org/Q18-1023/.

Yassir Lairgi, Ludovic Moncla, Khalid Benabdeslem, Rémy Cazabet, and Pierre Cléau. ATOM: AdapTive and OptiMized dynamic temporal knowledge graph construction using LLMs. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 950–966, 2026. URL https://aclanthology.org/2026.findings-eacl.49/.

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. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2405.17428.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2005.11401.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023. URL https://arxiv.org/abs/2308.03281.

Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, and Yongbin Li. StructRAG: Boosting knowledge intensive reasoning of LLMs via inference-time hybrid information structurization. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=GhexuBLxb0.

Yu Liu, Quanming Yao, and Yong Li. Generalizing tensor decomposition for N-ary relational knowledge bases. In *Proceedings of The Web Conference (WWW)*, 2020. URL https://arxiv.org/abs/2007.03988.

Yu Liu, Quanming Yao, and Yong Li. Role-aware modeling for N-ary relational knowledge bases. In *Proceedings of The Web Conference (WWW)*, 2021. URL https://arxiv.org/abs/2104.09780.

Haoran Luo, Haihong E, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin Kuang, Meina Song, Yifan Zhu, and Luu Anh Tuan. HyperGraphRAG: Retrieval-augmented generation via hypergraph-structured knowledge representation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 152206–152234, 2025. URL https://arxiv.org/abs/2503.21322.

Haoran Luo, Haihong E, Guanting Chen, Qika Lin, Yikai Guo, Fangzhi Xu, Zemin Kuang, Meina Song, Xiaobao Wu, Yifan Zhu, and Luu Anh Tuan. Graph-R1: Towards agentic GraphRAG framework via end-to-end reinforcement learning. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, volume 306 of *Proceedings of Machine Learning Research*, 2026. URL https://arxiv.org/abs/2507.21892.

Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. Multi-hop question answering. *Foundations and Trends in Information Retrieval*, 17(5):457–586, 2024. URL https://arxiv.org/abs/2204.09140.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2402.09906.

Qwen Team. Qwen3 technical report. Technical report, Alibaba Group, 2025. URL https://arxiv.org/abs/2505.09388.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*, 2025. URL https://arxiv.org/abs/2501.13956.

Stephen E. Robertson and Steve Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 232–241, 1994.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics (TACL)*, 10:539–554, 2022. URL https://aclanthology.org/2022.tacl-1.31/.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023. URL https://aclanthology.org/2023.acl-long.557/.

Changjian Wang, Weihong Deng, Weili Guan, Quan Lu, and Ning Jiang. Cross-granularity hypergraph retrieval-augmented generation for multi-hop question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33368–33376, 2026. URL https://ojs.aaai.org/index.php/AAAI/article/view/40623.

Yuhao Wang, Ruiyang Ren, Junyi Li, Wayne Xin Zhao, Jing Liu, and Ji-Rong Wen. REAR: A relevance-aware retrieval-augmented framework for open-domain question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5613–5626. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.emnlp-main.321/.

Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. When to use graphs in RAG: A comprehensive analysis for graph retrieval-augmented generation. *arXiv preprint arXiv:2506.05690*, 2025. URL https://arxiv.org/abs/2506.05690.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed resources for general Chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2024. URL https://arxiv.org/abs/2309.07597.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. 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 (EMNLP)*, 2018. URL https://aclanthology.org/D18-1259/.

Nan Zhang, Prafulla Kumar Choubey, Alexander Fabbri, Gabriel Bernadett-Shapiro, Rui Zhang, Prasenjit Mitra, Caiming Xiong, and Chien-Sheng Wu. SiReRAG: Indexing similar and related information for multihop reasoning. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=yp95goUAT1.

Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1601–1608, 2006.

A Latent Hyperedge Formalization

This appendix states in full the formalization summarized in Section 3.2.

Definition 1 (Latent Hyperedge). Let V be the set of entity vertices extracted from a corpus and H the set of indexed events (hyperedges), each with a stable identifier. Every event $h \in H$ is represented as a labeled hyperedge incident to an entity set $V(h) = \{v_1, \ldots, v_n\} \subseteq V$, where $n \geq 1$. Because one latent hyperedge simultaneously connects n entities, it preserves the n-ary co-occurrence structure of one chunk as an atomic retrieval unit. A single event "Company A acquired Company B for $X on Date D" links acquirer, target, price, and date in one unit. A direct binary decomposition requires multiple records and retains their common acquisition context only if it introduces an auxiliary event node or an equivalent reification mechanism. In SAG, the event description labels the hyperedge, while a stable event-to-chunk mapping $\phi : H \to D$ preserves the original chunk as the final evidence unit. This definition concerns the representation of each event; it does not define SAG as a materialized hypergraph.

Proposition 1 (Lossless Incidence Encoding). Define the incidence set

$$I = \{(h, v) \in H \times V \mid v \in V(h)\}. \tag{6}$$

Assuming that every incidence row is retained and event identifiers are unique, the event-entity SQL table of Section 3.1 stores I exactly and is therefore a lossless bipartite representation of the latent-hyperedge incidence relation (Luo et al., 2025). For every event,

$$V(h) = \{v \in V \mid (h, v) \in I\}. \tag{7}$$

Thus, grouping table rows by h reconstructs each hyperedge, while enumerating the pairs in all reconstructed hyperedges recovers the table. This proposition concerns storage after extraction; it makes no claim that mapping a source chunk to an event and entities preserves all source information.

Proposition 2 (Query-Time Dynamic Activation). Fix an index and retrieval configuration, and let $S_0 = R$ be the query-dependent seed event set used in Section 3.4. For unpruned expansion, define

$$F_{t+1} = \bigcup_{h \in S_t} V(h), \qquad S_{t+1} = \{h \in H \mid V(h) \cap F_{t+1} \neq \varnothing\}. \tag{8}$$

One expansion round computes $S_t \to F_{t+1} \to S_{t+1}$ through two SQL joins, corresponding to two traversals in the bipartite incidence graph $B = (V \cup H, I)$. Consequently, after L rounds the unpruned candidate events are exactly the event vertices within bipartite distance at most $2L$ from S_0. Frontier pruning only removes entities or events from these reachable sets, so practical SAG activates a budget-constrained, query-scoped set of incident latent hyperedges rather than the complete neighborhood. The activation is dynamic because this set is assembled for the query from its seed events under the fixed retrieval configuration, not because new corpus-level hyperedges are created at query time.

B Corpus-Level Connectivity Analysis

The formalization above describes one event in isolation. We next examine how latent hyperedges organize an entire corpus. A three-layer view makes the indexing structure explicit: each source chunk has a one-to-one mapping to a semantically complete event, while the event has a one-to-many mapping to its indexing entities. Entities that recur across events create paths between otherwise separate chunks. Thus, SAG does not connect chunks by replacing their text with graph fragments; it connects them through an intermediate event layer while retaining every original chunk as the evidence carrier.

Connection capacity of an event. Let $n_h = |V(h)|$ be the number of entities attached to event h. If the hyperedge is projected onto an entity–entity graph, its members admit

$$P(h) = \binom{n_h}{2} \tag{9}$$

distinct pairwise co-memberships. We call $P(h)$ the *pairwise connection capacity* of the event. It is a representational quantity rather than a count of physically stored edges: SAG stores only the n_h

incidence rows (h, v) and recovers the co-memberships by grouping on the event identifier. Across a corpus, the corresponding capacity is

$$P_{\text{corp}} = \sum_{h \in H} \binom{n_h}{2}. \tag{10}$$

A star-shaped binary decomposition needs $n_h - 1$ records to connect the same n_h entities to a designated anchor, giving the corpus-level amplification factor

$$A_{\text{corp}} = \frac{\sum_{h \in H} \binom{n_h}{2}}{\sum_{h \in H,\, n_h \geq 2} (n_h - 1)}. \tag{11}$$

This factor should not be read as a universal storage advantage over every possible triple schema. It quantifies how many pairwise co-memberships are jointly available from one labeled event unit relative to a fixed anchor-based binary decomposition. An arbitrary triple representation can preserve the same information only by choosing and maintaining an explicit reification scheme; independently indexed triples do not, by themselves, retain the common event boundary.

Cross-framework comparison on MuSiQue. Following Xiang et al. (2025), the evaluation converts each framework's native index into an undirected graph and reports the same node and link scale statistics. Because that protocol characterizes graph quality through density measures rather than reachability, we extend it with component-level statistics—component count, giant-component coverage, and isolation rate—which are the quantities relevant to whether evidence can be reached at all. The resulting graphs are not isomorphic representations: SAG uses event–entity incidences, GraphRAG and HyperGraphRAG use entity–entity links, LightRAG uses entity–relation incidences, and HippoRAG 2 uses passage–entity links. Raw node and link counts must therefore be read as native index scale rather than directly interchangeable units. The component-level statistics remain comparable, because they describe how fragmented each native retrieval structure is regardless of what its vertices denote. Here and throughout the paper, GraphRAG refers to the official Microsoft GraphRAG implementation (release v2.0.0).

Metric definitions and interpretation. For a framework's undirected native index $G = (V_G, E_G)$:

Nodes. The total number of vertices $|V_G|$ in the native index. A vertex may represent an entity, event, relation, or passage depending on the framework.

Links. The total number of edges or incidence links $|E_G|$. Link semantics likewise depend on the native index schema.

Avg. degree. The mean number of incident links per vertex, $\bar{d} = 2|E_G|/|V_G|$. It measures connection intensity and traversal burden, not retrieval accuracy.

Components. The number of connected subgraphs that are mutually unreachable. Fewer components indicate less fragmentation, but the value should be interpreted together with graph size.

Giant comp. The number of vertices $|C_{\max}|$ in the largest connected component and its percentage of all nodes, $100 \times |C_{\max}|/|V_G|$. A higher percentage indicates that more of the index belongs to one potentially reachable region.

Isolated. The number of degree-zero vertices and their percentage of all nodes, $100 \times |\{v \in V_G \mid \deg(v) = 0\}|/|V_G|$. These vertices cannot be reached from any other vertex through graph traversal.

Nodes and links describe native index scale rather than semantic richness because their meanings differ across frameworks. For the connectivity metrics, lower component and isolation values indicate less disconnected content, while higher giant-component coverage indicates broader potential reachability.

We omit density, clustering coefficient, and whole-graph diameter from the primary comparison. Density is dominated by graph scale, triangle-based clustering is structurally zero for bipartite indices such as SAG, and whole-graph diameter is infinite whenever an index is disconnected. These

descriptors therefore do not directly answer whether chunks participate in reachable cross-document structures.

Table 4: Native index scale and connectivity on MuSiQue. "Index relation" identifies the vertex and link semantics used by each framework. Since these semantics differ, nodes and links are descriptive scale statistics; components, giant-component coverage, and isolation quantify fragmentation within each native index. Percentages in parentheses use the Nodes column as the denominator.

Method	Index relation	Nodes	Links	Avg. degree	Components	Giant comp.	Isolated
GraphRAG	Entity–entity	128,189	106,489	1.66	66,472	58,815 (45.9%)	65,506 (51.1%)
LightRAG	Entity–relation	82,321	116,555	2.83	9,277	70,196 (85.3%)	8,512 (10.3%)
HyperGraphRAG	Entity–entity	186,897	272,524	2.92	4,600	171,196 (91.6%)	485 (0.3%)
HippoRAG 2	Passage–entity	97,043	570,458	11.76	96	96,773 (99.7%)	59 (0.1%)
SAG	Event–entity	95,341	126,540	2.65	400	92,206 (96.7%)	1 (0.001%)

GraphRAG has 166 times as many connected components as SAG, LightRAG has 23.2 times as many, and HyperGraphRAG has 11.5 times as many. SAG's largest component covers 96.7% of the native index, compared with 45.9%, 85.3%, and 91.6%, respectively, while its isolated-vertex rate is effectively zero. These differences show that repeated entities connect semantically complete events into a broad cross-chunk substrate rather than leaving them as local records.

HippoRAG 2 occupies the opposite end of the connectivity spectrum: it has only 96 components and 99.7% giant-component coverage, but uses 570,458 links, 4.5 times SAG's count, and raises the average degree from 2.65 to 11.76. Its MuSiQue Recall@5 is nonetheless 65.13%, below SAG's 80.36%. Connectivity statistics therefore characterize the retrieval substrate rather than predict retrieval quality: SAG attains comparable coverage at substantially lower degree, while query-time event–entity traversal converts this structure into useful retrieved evidence.

Table 5: Connectivity of SAG's event–entity incidence index on the three evaluation corpora. Vertices include events and entities; incidence rows are event–entity links. Lower component and isolated-vertex counts indicate less fragmentation, while larger giant-component coverage indicates greater corpus-level reachability. Percentages in parentheses use the Vertices column as the denominator.

Dataset	Vertices	Incidences	Avg. degree	Components	Giant comp.	Isolated vertices
MuSiQue	95,341	126,540	2.65	400	92,206 (96.7%)	1 (0.001%)
HotpotQA	100,570	186,634	3.71	5,869	93,593 (93.1%)	5,787 (5.8%)
2WikiMultiHopQA	57,812	75,399	2.61	272	56,270 (97.3%)	0 (0.0%)

SAG connectivity across datasets. The largest connected component covers more than 93% of the index on every dataset and more than 96% on MuSiQue and 2WikiMultiHopQA. This does not mean that every pair of chunks is relevant to the same query, nor that traversal should be global. It shows that shared entities provide a broad substrate of possible cross-chunk routes from which SAG activates a small query-conditioned neighborhood. The moderate average degree of 2.61–3.71 further distinguishes reachability from indiscriminate density: the index is broadly connected without requiring every event to become a high-degree hub.

Connection structure and retrieval. The indexing ablation in Table 2 holds the corpus, retrieval pipeline, and downstream model fixed within each benchmark. Hyperedge indexing outperforms triple indexing at every reported cutoff on all three datasets, with Recall@5 gains of 2.75 points on MuSiQue, 9.24 on 2WikiMultiHopQA, and 13.80 on HotpotQA. Expansion also improves recall across the three datasets, but its largest gain occurs on 2WikiMultiHopQA. The different patterns indicate that event-level representation and query-time expansion contribute differently across datasets. The connectivity statistics describe available cross-chunk routes, while the ablations establish the retrieval benefit of these two components within SAG.

C Qualitative Examples of Query-Time Hyperedge Activation

The intermediate states of the retrieval procedure in Section 3.4 form an auditable execution chain:

$$q \to Q \to \widehat{Q} \to R \to C \to \widehat{C} \to D_{\text{out}}. \tag{12}$$

Figures 4 and 5 pair the recorded intermediate outputs with their retrieval visualizations. The left-hand tables expose the extracted query entities, seed events returned by the entity-guided and direct event-retrieval paths, shared entities available for joining, and latent hyperedges activated during expansion. The right-hand diagrams visualize the corresponding query-centered neighborhoods. Both examples retrieve relevant evidence, but they differ in how effectively query-time dynamic hyperedge activation extends the initial result. The legend directly below the good-example diagram applies to both diagrams.

Query	How many times did the plague occur in the city where the painter of *The Bacchanal of the Andrians* died?
Gold Docs	1. The Bacchanal of the Andrians; 2. Black Death; 3. The Martyrdom of Saint Lawrence (Titian)
Extracted entity	The Bacchanal of the Andrians
Expanded entity	Venice
Expanded events	1. Venice; **2. The Martyrdom of Saint Lawrence (Titian)**; 3. Venice
Recalled events (Top-5)	**1. Black Death**; **2. The Bacchanal of the Andrians**; 3. Venice; **4. The Martyrdom of Saint Lawrence (Titian)**; 5. Venice

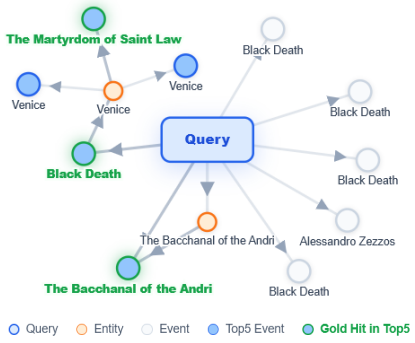


Figure 4: Example with a good retrieval result. The initial paths recover two gold events, and the shared entity *Venice* activates an additional latent hyperedge that recovers the third gold document. Bold titles mark gold documents in the corresponding table rows.

Query	When did the city where Greenwood Laboratory School is located become capital of the state where the screenwriter of *The Poor Boob* was born?
Gold Docs	1. Greenwood Laboratory School; 2. Margaret Mayo (playwright); 3. The Poor Boob; 4. Springfield, Illinois
Extracted entities	The Poor Boob; Greenwood Laboratory School
Expanded entity	None
Expanded events	None
Recalled events (Top-5)	**1. Greenwood Laboratory School**; 2. Maine; **3. The Poor Boob**; 4. Washington (state); 5. Oklahoma

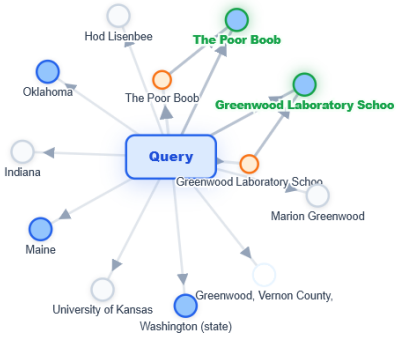


Figure 5: Example with a poor retrieval result. The initial paths recover two gold events, but no shared entity is available to activate latent hyperedges leading to the remaining evidence. Bold titles denote gold documents in the Top-5 result.

The good example directly illustrates query-time dynamic hyperedge activation. The initial entity-guided and direct event-retrieval paths recover *Black Death* and *The Bacchanal of the Andrians*. SAG then discovers the shared entity *Venice* through SQL joins and activates its incident latent hyperedges, bringing *The Martyrdom of Saint Lawrence (Titian)* into the final Top-5 result. The poor example exposes the complementary boundary. Although direct retrieval finds *Greenwood Laboratory School* and *The Poor Boob*, the returned events provide no retained shared entity that can activate a route toward *Margaret Mayo (playwright)* and *Springfield, Illinois*. The contrast isolates the role of dynamic hyperedges: they add useful cross-document evidence when the seed events reveal a discriminative bridge, while the absence of such a bridge leaves the result dependent on direct retrieval.

D Full Retrieval Results

Table 6: Full retrieval results (Recall@1/2/5/10) on three multi-hop benchmarks. All structure-augmented methods use BGE-Large-EN-v1.5 as the retriever; Qwen3.6-Flash is used as the reader.

	MuSiQue				2WikiMultiHopQA				HotpotQA			
	R@1	R@2	R@5	R@10	R@1	R@2	R@5	R@10	R@1	R@2	R@5	R@10
Simple Baselines												
BM25	24.59	32.96	42.11	49.69	36.85	54.30	63.78	67.90	38.75	56.45	73.05	85.40
Contriever	24.50	33.54	45.32	53.58	34.40	48.60	59.25	65.20	39.30	57.45	74.85	82.75
Large Embedding Models												
BGE-Large-EN-v1.5	29.42	41.68	56.32	64.08	41.48	62.28	69.83	73.75	45.45	76.20	89.20	93.35
GTE-Qwen2-7B	35.19	48.06	61.92	70.80	43.45	65.58	74.18	78.63	46.40	73.95	88.45	93.55
GritLM-7B	34.86	50.70	66.22	75.19	41.85	66.23	75.62	79.40	46.40	81.15	93.40	97.20
NV-Embed-v2	34.02	51.08	68.84	77.27	43.23	68.93	77.00	80.75	46.90	85.55	95.25	**97.70**
Structure-Augmented RAG												
HippoRAG 2	30.65	49.52	65.13	73.76	42.38	76.55	90.35	93.40	44.40	78.35	94.35	97.15
SAG (Ours)	**36.82**	**63.62**	**80.36**	**83.37**	**43.81**	**83.93**	**93.34**	**93.57**	**47.80**	**91.55**	**96.50**	**97.70**

E Full QA Results

For either EM or F1, Avg. is query-weighted by dataset size:

$$\text{Avg} = \frac{1000(\text{MuSiQue} + \text{2Wiki} + \text{HotpotQA}) + 293\,\text{NarrativeQA}}{3293}, \tag{13}$$

where each dataset name denotes its corresponding score.

Table 7: Full QA results (EM/F1) on all four benchmarks. All methods use Qwen3.6-Flash as the reader, and all structure-augmented methods use BGE-Large-EN-v1.5 as the retriever.

	Multi-Hop QA						Discourse Understanding			
	MuSiQue		2Wiki		HotpotQA		NarrativeQA		Avg.	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Simple Baselines										
BM25 (Robertson & Walker, 1994)	26.10	35.19	52.40	56.26	52.90	64.64	4.44	16.42	40.30	48.86
Contriever (Izacard et al., 2022)	25.70	34.39	45.20	50.62	52.80	64.06	6.83	18.93	38.18	46.95
Large Embedding Models										
BGE-Large-EN-v1.5 (Xiao et al., 2024)	33.00	42.80	56.10	61.49	62.40	74.81	6.83	20.66	46.62	56.24
GTE-Qwen2-7B (Li et al., 2023)	34.70	44.60	59.90	64.91	61.10	73.04	5.46	19.71	47.78	57.20
GritLM-7B (Muennighoff et al., 2025)	38.60	49.30	60.70	66.12	61.00	73.22	8.87	26.23	49.48	59.62
NV-Embed-v2 (Lee et al., 2025)	41.70	52.88	62.50	68.45	65.40	78.00	8.87	25.59	52.29	62.81
Structure-Augmented RAG										
GraphRAG (Edge et al., 2024)	43.20	54.14	64.70	72.08	63.70	76.29	9.91	27.24	52.99	63.92
LightRAG (Guo et al., 2025)	33.60	46.66	59.00	65.75	61.20	73.09	8.53	24.23	47.47	58.49
HyperGraphRAG (Luo et al., 2025)	43.60	54.78	65.70	73.06	64.60	76.74	9.13	25.71	53.62	64.41
HyperRAG (Feng et al., 2026)	40.90	51.80	69.40	76.89	64.40	77.19	9.22	25.95	53.88	64.84
HippoRAG 2 (Gutiérrez et al., 2025)	37.90	47.49	67.00	74.34	65.20	77.24	7.17	23.49	52.30	62.54
SAG (Ours)	**50.20**	**61.15**	**70.50**	**78.06**	**67.90**	**79.66**	**12.19**	**31.90**	**58.36**	**69.30**

F Hyperparameter Results

Table 8: Hyperparameter sweeps on MuSiQue. Expansion depth varies L; $L = 0$ corresponds to the w/o Expansion ablation in Table 2, while $L = 1$ is the default SAG setting. Candidate budget varies K_{cand}. In the dual-path sweep, K_{event} specifies the exact number of events selected by the LLM, and the semantic path fills the remaining $K_{\text{direct}} = K_{\text{out}} - K_{\text{event}}$ slots, with $K_{\text{out}} = 10$. All other settings remain at their defaults. Bold marks the best value within each group.

		MuSiQue			
Parameter	Configuration	R@1	R@2	R@5	R@10
Expansion depth	$L = 0$	35.70	57.75	69.41	74.76
	$L = 1$ (default)	**36.82**	63.62	80.36	83.37
	$L = 2$	36.17	**64.05**	80.47	84.00
	$L = 3$	36.03	63.42	80.40	83.79
	$L = 4$	36.27	63.83	**80.71**	**84.14**
Candidate budget	$K_{\text{cand}} = 50$	36.53	62.80	79.35	81.69
	$K_{\text{cand}} = 100$ (default)	36.82	63.62	80.36	83.37
	$K_{\text{cand}} = 200$	**36.99**	**65.12**	81.06	84.47
	$K_{\text{cand}} = 500$	36.46	64.72	**82.24**	**86.56**
Dual-path output	$K_{\text{event}} = 0$ (semantic only)	29.37	41.56	56.23	63.77
	$K_{\text{event}} = 2$	36.62	**64.56**	73.52	77.64
	$K_{\text{event}} = 4$	36.61	63.73	79.80	82.63
	$K_{\text{event}} = 5$ (default)	**36.82**	63.62	**80.36**	83.37
	$K_{\text{event}} = 6$	36.71	63.83	79.93	83.40
	$K_{\text{event}} = 8$	36.00	62.86	80.14	**83.57**
	$K_{\text{event}} = 10$	36.46	63.48	79.80	83.18

G Runtime, Token Consumption, and Scaling

Table 3 provides the measurements underlying the following analysis of offline construction, online cost–quality trade-offs, and LLM computation.

Offline construction cost. SAG's index requires 4.57 h and 28.26 M tokens to build. Relative to GraphRAG, LightRAG, and HyperGraphRAG, this reduces indexing time by 55.0–82.4% and indexing tokens by 72.6–86.9%. Among structure-augmented baselines, HippoRAG 2 has the lowest offline cost (2.30 h; 15.93 M tokens). SAG therefore reduces, but does not eliminate, the offline cost of structure augmentation.

Online cost–quality frontier. At $K_{\text{cand}} = 50$, SAG is simultaneously more accurate, faster end to end, and less token-intensive than LightRAG and HyperGraphRAG: it obtains 60.41 F1 in 22.68 s with 8,236 tokens, versus 46.66/30.10 s/9,974 tokens and 54.78/27.99 s/32,536 tokens, respectively. Against GraphRAG and HippoRAG 2, the comparison is instead a quality–latency trade-off: SAG gains 6.27 and 12.92 F1 points, respectively, but takes longer end to end. The stage breakdown explains where this time is spent. SAG's 15.93 s retrieval-and-selection stage is the bottleneck, whereas its 6.75 s QA stage is faster than those of all four structure-augmented baselines (8.50–23.54 s). Thus, SAG moves computation from answer generation to evidence construction rather than being the fastest system overall.

Return on LLM computation. The controlled substitutions show what the online selection cost buys. In Table 2, replacing the Qwen3.6-Flash selector with Qwen3-Reranker-8B lowers Recall@5 from 80.36% to 67.11%, a 13.25-point drop. In Table 8, setting $K_{\text{event}} = 0$ and thereby removing LLM-selected structural output lowers Recall@5 to 56.23%, 24.13 points below the default dual-path configuration. Within SAG, increasing K_{cand} from 50 to 100 adds only 0.74 F1 points while increasing per-query token consumption by 40.1% and end-to-end latency by 10.4%. Consequently, $K_{\text{cand}} = 50$ captures most of the measured quality at the more efficient operating point, while $K_{\text{cand}} = 100$ is the accuracy-oriented configuration used for the main results.

H LLM-as-a-Judge Metric Definitions

Table 9: LLM-as-a-judge evaluation, scored by Qwen3.7-Plus. All four metrics are percentages (higher is better). Each metric column reports MuSiQue / 2WikiMultiHopQA / HotpotQA / NarrativeQA, in that order. NarrativeQA has no annotated gold evidence chunks, so evidence recall is not defined for it.

Method	Answer correctness	Coverage	Context relevancy	Evidence recall
	MuSiQue / 2WikiMultiHopQA / HotpotQA / NarrativeQA			
GraphRAG	56.56 / 79.32 / 84.91 / 31.43	46.60 / 77.97 / 62.93 / 33.43	59.17 / 73.02 / 89.86 / 65.79	52.31 / 71.74 / 77.88 / –
LightRAG	53.07 / 72.52 / 80.98 / 27.25	43.54 / 71.15 / 60.88 / 31.16	47.55 / 63.65 / 83.57 / 60.48	63.91 / 75.89 / 87.49 / –
HyperGraphRAG	59.54 / 81.17 / 85.71 / 30.40	50.27 / 80.41 / 64.18 / 31.94	**65.51** / 80.27 / 89.55 / 58.71	73.58 / 83.71 / 91.74 / –
HyperRAG	56.01 / 84.93 / 85.32 / 31.15	48.06 / 84.23 / 63.16 / 30.65	58.74 / **80.30** / **91.13** / 57.44	68.23 / 87.90 / 92.37 / –
HippoRAG 2	50.09 / 83.93 / 84.91 / 22.19	43.20 / 82.42 / 63.31 / 22.65	45.75 / 74.58 / 86.65 / 46.16	64.77 / 88.45 / 93.65 / –
SAG (Ours)	**63.69** / **85.26** / **87.95** / **35.22**	**54.73** / **85.07** / **64.90** / **38.02**	59.54 / 76.02 / 89.50 / **71.93**	**78.87** / **89.44** / **95.87** / –

We follow the LLM-as-a-judge protocol of Xiang et al. (2025) and use Qwen3.7-Plus to evaluate every method. Table 10 summarizes the evaluation metrics. Each metric is normalized to a 0–1 scale and multiplied by 100 for presentation in Table 9; higher is better.

Table 10: Definitions of the LLM-as-a-judge evaluation metrics.

Metric	**Input**	**Description**	**Scoring formula**	**Output**
Answer correctness	Generated answer; ground-truth answer; question	Separate calls decompose the ground-truth answers into atomic facts. A third call matches the facts and counts true positives (TP), false positives (FP), and false negatives (FN).	$P = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$, $R = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$, $F_1 = \frac{2PR}{P+R}$	$[0, 1]$
Coverage	Generated answer; ground-truth answer; question	Separate calls decompose the generated and ground-truth answers into atomic facts. The judge then determines whether each ground-truth fact is covered by the generated answer.	$N_{\text{covered gold}}/N_{\text{gold}}$	$[0, 1]$
Context relevancy	Retrieved contexts; question	The judge assigns a score of 0 (irrelevant), 1 (partially relevant), or 2 (fully sufficient to answer). Two independent ratings are obtained.	$(s_1 + s_2)/4$, where $s_1, s_2 \in \{0, 1, 2\}$	$[0, 1]$
Evidence recall	Retrieved contexts; gold evidence chunks; question	The judge determines whether each gold evidence item can be inferred from the retrieved contexts.	$N_{\text{entailed}}/N_{\text{gold}}$	$[0, 1]$

I Indexing Prompt for Event and Entity Extraction

The offline indexing stage of Section 3.1 issues a single structured-output call per chunk to the indexing LLM (Qwen3.6-Flash in our deployment). The model returns exactly one event together with its typed entities, which are written as one latent hyperedge. Figure 6 reproduces the system prompt and a representative input/output pair. Note that the example input in the figure includes a `source_title` that disagrees with the passage (a fuzzing artifact of the benchmark corpus); the instruction to remain faithful to the passage makes the extractor robust to such noise.

System prompt

Role. You are a professional content extractor whose core task is to extract two types of structured information from raw documents: **events** and **entities**.

Event extraction principles.

- *Mandatory single event*: regardless of how many different topics the input contains, they must be integrated into one comprehensive event; splitting into multiple events is forbidden.
- Use short, complete subject–verb–object sentences to state facts; avoid mechanically copying the original text.
- *Exhaustive information coverage*: all core information from relevant fragments must be reflected in the event content; every sentence must have explicit textual support in the input fragments.
- *Faithful to the original text*: do not add or omit facts, do not infer on your own, and do not change the original subject.
- Preserve all key numbers and retain colloquial or relative time expressions in their original wording unless the input explicitly provides an exact date.
- Ignore advertisements, navigation text, boilerplate, source labels, and fragments without factual value.

Entity extraction principles.

- Extract entities necessary for understanding the event (people, organizations, locations, time, products, key data, etc.), especially subjects and objects.
- Full names, abbreviations, and aliases of the same entity must all be extracted; entities connected by "and/or/with" etc. must be split.
- Entity types must be selected from `entity_types`; the description must clarify the entity's specific role in the event.

Output requirements.

- Return exactly one event containing: `title` (a short, discriminative English title containing the key subject); `content` (a self-contained English factual account preserving important subjects, relations, numbers, and time expressions); `entities` (the entities required to retrieve the event); and `is_valid` (`true` only when the input contains an extractable factual event).
- For a valid event, `entities` must contain at least one entity. If no factual event can be extracted, set `is_valid` to `false`, use a short English placeholder title/content, and return an empty `entities` array.
- Each entity contains exactly: `type` (one of the entity types supplied in the input metadata), `name` (the entity's English name), and `description` (its specific role or relationship in this event).
- Return JSON only and follow the supplied response schema exactly: the top-level object contains `type` set to `"response"` and `data`; `data` contains only an `items` array holding the single event object.
- Generate the title, content, entity names, and entity descriptions in English.

User message (one request per chunk)

The request payload carries the chunk to extract, the closed entity-type inventory, and a short previous-chunk context:

- `items[0].idx`: 26; `items[0].title`: `Clement Attlee`; `items[0].text`: the raw passage associated with this item:

```
Clement Richard Attlee, 1st Earl Attlee, {'1': '', '2': '', '3': '', '4': ''} (3 January 1883 -- 8 October
 1967) was a British Labour politician who served as Prime Minister of the United Kingdom from 1945 to
 1951 and Leader of the Labour Party from 1935 to 1955. In 1940, Attlee took Labour into the wartime
coalition government and served under Winston Churchill, becoming the first person to hold the office
of Deputy Prime Minister of the United Kingdom. He went on to lead the Labour Party to an unexpected
landslide victory at the 1945 general election; forming the first Labour majority government, and a
mandate to implement its postwar reforms. The 12.0\% national swing from the Conservatives to Labour
was unprecedented at that time and remains the largest ever achieved by any party at a general
election in British political history. He was re-elected with a narrow majority at the 1950 general
election. In the following year, Attlee called a snap general election, hoping to increase his
parliamentary majority. However, he was narrowly defeated by the Conservatives under the leadership of
 Winston Churchill; despite winning the most votes of any political party in any general election in
British political history until the Conservative Party's fourth consecutive victory in 1992. Attlee
remains the longest-ever serving Leader of the Labour Party.
```

- `meta.entity_types`: the 11 closed types of Section 3.1 (`time`, `location`, `person`, `organization`, `group`, `subject`, `work`, `product`, `action`, `metric`, `tags`), each with a short definition and examples.
- `meta.source_type/source_title/source_summary`: document-level metadata (not necessarily aligned with the passage).
- `meta.previous_context`: the head of the previous chunk, used to resolve cross-chunk references.

Example output (full model response)

The complete, unabridged JSON returned by the model for the example chunk (all 15 entities shown):


```
{
  "type": "response",
  "data": {
    "items": [
      {
        "title": "Clement Attlee",
        "content": "Clement Richard Attlee, 1st Earl Attlee, {'1': \", '2': \", '3': \", '4': \"} (3
      January 1883 -- 8 October 1967) was a British Labour politician who served as Prime Minister of
      the United Kingdom from 1945 to 1951 and Leader of the Labour Party from 1935 to 1955. In 1940,
      Attlee took Labour into the wartime coalition government and served under Winston Churchill,
      becoming the first person to hold the office of Deputy Prime Minister of the United Kingdom. He
      went on to lead the Labour Party to an unexpected landslide victory at the 1945 general election;
       forming the first Labour majority government, and a mandate to implement its postwar reforms.
      The 12.0% national swing from the Conservatives to Labour was unprecedented at that time and
      remains the largest ever achieved by any party at a general election in British political history.
       He was re-elected with a narrow majority at the 1950 general election. In the following year,
      Attlee called a snap general election, hoping to increase his parliamentary majority. However, he
       was narrowly defeated by the Conservatives under the leadership of Winston Churchill; despite
      winning the most votes of any political party in any general election in British political
      history until the Conservative Party's fourth consecutive victory in 1992. Attlee remains the
      longest-ever serving Leader of the Labour Party.",
        "entities": [
          {"type": "person", "name": "Clement Attlee", "description": "British Labour politician who
      served as Prime Minister from 1945 to 1951 and Leader of the Labour Party from 1935 to 1955."},
          {"type": "time", "name": "1945 to 1951", "description": "The period during which Clement
      Attlee served as Prime Minister of the United Kingdom."},
          {"type": "time", "name": "1935 to 1955", "description": "The period during which Clement
      Attlee served as Leader of the Labour Party."},
          {"type": "time", "name": "1940", "description": "The year Attlee joined the wartime
      coalition government and became Deputy Prime Minister."},
          {"type": "time", "name": "1945", "description": "The year of the general election where
      Attlee led Labour to a landslide victory."},
          {"type": "time", "name": "1950", "description": "The year of the general election where
      Attlee was re-elected with a narrow majority."},
          {"type": "time", "name": "1951", "description": "The year Attlee called a snap general
      election and was narrowly defeated."},
          {"type": "organization", "name": "Labour Party", "description": "The political party led by
      Attlee from 1935 to 1955 and which he led to victory in 1945."},
          {"type": "location", "name": "United Kingdom", "description": "The country where Attlee
      served as Prime Minister and where the general elections took place."},
          {"type": "person", "name": "Winston Churchill", "description": "Prime Minister under whom
      Attlee served as Deputy Prime Minister in 1940, and the Conservative leader who defeated Attlee
      in 1951."},
          {"type": "metric", "name": "12.0%", "description": "The unprecedented national swing from
      the Conservatives to Labour in the 1945 general election."},
          {"type": "subject", "name": "Deputy Prime Minister of the United Kingdom", "description": "
      The office Attlee held starting in 1940, being the first person to hold it."},
          {"type": "subject", "name": "1945 general election", "description": "The election where
      Attlee led Labour to a landslide victory and the largest swing in British history."},
          {"type": "subject", "name": "1950 general election", "description": "The election where
      Attlee was re-elected with a narrow majority."},
          {"type": "subject", "name": "snap general election", "description": "The election called by
      Attlee in 1951, which resulted in his narrow defeat."}
        ],
        "is_valid": true
      }
    ]
  }
}
```


Figure 6: Prompt used for offline event–entity extraction. The system prompt enforces one comprehensive event per chunk with exhaustive, faithful coverage; the user message supplies the chunk, the closed entity-type inventory, and previous-chunk context; the model returns the event and typed entities that form one latent hyperedge.

J LLM Selector Prompt Input and Output

The LLM selector receives a multi-hop question and candidate relationship descriptions and returns exactly five useful relation indices. Figure 7 reproduces the prompt exchange in the same compact

visual style as Figure 6. The candidate descriptions are represented by their source indices, while the remaining prompt and response content is retained.

System prompt

I will provide you with a set of relationship descriptions from a knowledge graph. Select exactly 5 relationships that are most useful for answering this multi-hop question.

Return JSON with `thought_process` and `useful_relations` (a list of exactly 5 relation indices like `[id]`, ordered from most to least useful).

User message (target request)

Question: What month did the Tripartite discussions begin between Britain, France, and the country where, despite being headquartered in the nation called the nobilities commonwealth, the top-ranking Warsaw Pact operatives originated?

Relationship descriptions: `[0], [1], [2], ..., [99]`

Assistant message (target output)

```
"thought_process": "1. Identify the country where the 'top-ranking Warsaw Pact operatives originated'
    and which is 'headquartered in the nation called the nobilities commonwealth'. From relation
   [43], the 'Polish-Lithuanian Commonwealth' is referred to as the 'nobilities commonwealth'. The
   capital of the Warsaw Pact's Combined Command was in Warsaw, Poland (relation [3]). Thus, the
   country in question is Poland.\n2. Identify the 'Tripartite discussions' involving Britain,
   France, and Poland. Relation [22] describes 'Tripartite negotiations' involving the USSR, Britain,
    and France, but the question specifies Britain, France, and Poland. However, historical context
   often links these negotiations to the broader Allied talks. Let's re-read carefully. The question
    asks about discussions between Britain, France, and the country identified (Poland). Wait,
   relation [22] says 'mid-June, the main Tripartite negotiations started' and mentions 'Britain
   opposed such proposals...'. It doesn't explicitly list Poland as a party to the start of the
   negotiations in the text, but the historical event is the Anglo-Franco-Soviet talks. Let's look
   for a relation that mentions the start date. Relation [22] states: 'By the end of May, drafts
   were formally presented, and in mid-June, the main Tripartite negotiations started.'\n3.
   Determine the month. Relation [22] says 'mid-June'. So the month is June.\n4. Select the most
   useful relations to confirm the entities and the date.\n   - [43] identifies 'Poland' as the '
   nobilities commonwealth'.\n   - [3] identifies 'Poland' (Warsaw) as the headquarters of the
   Warsaw Pact operatives, confirming the country is Poland.\n   - [22] provides the date 'mid-June'
    for the start of the 'Tripartite negotiations' involving the relevant powers (Britain, France,
   USSR - though the question says Britain, France, and the country... actually, the question
   phrasing 'between Britain, France, and the country...' might be a trick or referring to a
   specific bilateral/multilateral context. However, given the options, [22] is the only one
   describing 'Tripartite discussions' with a date. The 'Tripartite' usually refers to UK, France,
   USSR. Does the question imply Poland was a party? Or is it referring to the negotiations
   regarding Poland? 'The discussion focused on potential guarantees to central and east European
   countries...'. The 'Tripartite' usually means the three great powers. Let's assume the question
   refers to the events in [22].\n   - Let's check if there are other 'Tripartite' mentions. [93]
   mentions 'Tripartite Pact' (Germany, Italy, Japan). Not relevant.\n   - So the key is linking '
   nobilities commonwealth' -> Poland -> Warsaw Pact HQ -> Poland. Then finding the Tripartite talks
    involving Britain and France (and implicitly the USSR, or perhaps the question implies the talks
    about the country?). The text in [22] says 'Tripartite negotiations started... mid-June'.\n   -
   Therefore, the relations needed are: [43] (Poland = nobilities commonwealth), [3] (Warsaw Pact HQ
    in Poland, confirming the origin of operatives), and [22] (Date of Tripartite negotiations). We
   need 5 relations.\n   - [1] also mentions Warsaw Pact established in Warsaw, Poland. Useful to
   reinforce [3].\n   - [12] mentions Solidarity in Poland, confirming Poland's identity in the
   Warsaw Pact context.\n   - So the set is [43], [3], [22], [1], [12].",
"useful_relations": ["[43]", "[3]", "[22]", "[1]", "[12]"]
```

Figure 7: LLM selector prompt exchange; candidate descriptions are identified by their source indices, while the remaining fields are shown in full.