BEYOND CHUNK-LOCAL EXTRACTION: CROSS-CHUNK GRAPH AUGMENTATION FOR GRAPHRAG

Anonymous authors
Paper under double-blind review

ABSTRACT

GraphRAG extends retrieval-augmented generation by organizing corpora as explicit knowledge graphs, enabling graph-based retrieval for complex question answering. However, existing frameworks extract entities and relations only within individual chunks, leaving cross-chunk relations—those whose evidence spans multiple passages—systematically absent from the index. Exhaustive LLM-based recovery of such relations is impractical due to the combinatorial explosion of chunk combinations. We present CROSSAUG, a GNN-guided CROSS-Chunk Graph AUGmentation method that enriches GraphRAG indices with cross-chunk relational structure as an offline step before query-time retrieval. CROSSAUG derives its training signal from self-supervised training pairs constructed from the base graph, uses a topology-aware GNN to score subgraphs for missingness, and applies evidence-grounded LLM completion only to the selected high-scoring regions. Experiments on three LLM-based GraphRAG frameworks across four multi-hop and long-document QA benchmarks show improvements across all framework–dataset–metric comparisons, confirming the benefit of cross-chunk graph augmentation for retrieval-based question answering. Our code is available at `https://anonymous.4open.science/r/CrossAug-72BB/`.

1 INTRODUCTION

Retrieval-augmented generation (RAG) is widely used for processing long documents, where documents are typically chunked into smaller segments for retrieval and generation. Recent GraphRAG frameworks extend this paradigm by organizing the corpus as an explicit graph in which entities are extracted from text and linked via relation triples, enabling graph algorithms and community summarization to support multi-hop retrieval and corpus-level queries (Edge et al., 2024; Gutiérrez et al., 2024). This direction has shown particular promise for long texts and complex QA, where flat vector retrieval frequently surfaces superficially relevant information while missing the critical evidence that such questions require.

However, the conventional chunk-based construction of GraphRAG introduces a fundamental blind spot. Existing frameworks typically extract entities and relations within individual chunks, implicitly assuming that document-level relational structure can be recovered by stitching together locally extracted triples. This assumption is fragile for long texts, where many important relations, such as alias resolution and causal chains, depend on evidence distributed across multiple chunks. For example, when HippoRAG2 (Gutiérrez et al., 2025) constructs a graph index for *Voodoo Planet*, the resulting graph contains locally correct triples such as *(Tau, approached, Lumbrilo)* and *(Tau, defeated, Lumbrilo)*, but these form only an event chain around Tau and Lumbrilo without specifying where the confrontation takes place, causing the framework to answer incorrectly. The cross-chunk locative relation *(Khatkan swamp, location of, final confrontation with Lumbrilo)* can be recovered only by jointly reading the chunk that describes the swamp with those describing the battle, a combination that a chunk-local extractor never sees. Community summarization addresses corpus-level queries but cannot recover such precise cross-chunk relational facts, and it may even discard the relational details it is meant to preserve (Han et al., 2026). A GraphRAG index may thus contain many locally correct edges while still missing the hidden cross-chunk relations required for long-document reasoning.

Most existing GraphRAG methods focus on improving how an already constructed graph is used for retrieval, ranking, summarization, or multi-hop reasoning. If the graph is incomplete at construction

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.

time, however, downstream algorithms can only operate over a partial structure (Yang et al., 2026). While LLMs could in principle recover missing cross-chunk relations, applying them exhaustively is impractical due to the combinatorial explosion of chunk combinations. We therefore introduce graph neural networks (GNNs) into the GraphRAG pipeline as a structural guide: by combining semantic node features with relational graph structure through edge-type-aware message passing (Gilmer et al., 2017; Schlichtkrull et al., 2018), a GNN can identify structural patterns characteristic of chunk-local extraction failures and locate where evidence-grounded LLM completion is most likely to recover useful relational structure.

In this paper, we introduce CROSSAUG, a GNN-guided CROSS-Chunk Graph AUGmentation method for GraphRAG. CROSSAUG is built around three design principles. First, a *self-supervised training-data construction module* creates training data by corrupting the base graph itself to match the incompleteness patterns introduced by chunk-local extraction. Second, a *topology-aware GNN scoring module* learns to assign missingness scores to local subgraphs, turning cross-chunk discovery from exhaustive entity enumeration into a targeted extraction problem with a budget hyperparameter. Third, a *GNN-guided LLM completion module* applies the LLM only to selected, evidence-grounded subgraphs and writes the validated triples back into the graph index. The additional reasoning cost is therefore paid once during offline augmentation, while the enriched relational structure can be reused by any downstream retrieval pipeline.

Our contributions are threefold:

- We identify the chunk-local extraction bottleneck in GraphRAG and formulate it as a problem of recovering cross-chunk entities and relations.
- We propose a self-supervised GNN scoring framework that reframes cross-chunk discovery as subgraph-level missingness detection, replacing intractable enumeration with targeted, topology-guided selection.
- We present a GNN–LLM collaboration pipeline that improves performance across three GraphRAG frameworks and four QA benchmarks.

2 RELATED WORK

GraphRAG has recently attracted considerable attention in the research community. MS-GraphRAG (Edge et al., 2024) is among the first GraphRAG frameworks to build a knowledge graph index directly from raw text for corpus-level question answering. It demonstrates strong improvements over flat vector retrieval for broad queries, although its graph construction requires substantial LLM inference.

Recent GraphRAG frameworks can be broadly grouped by how they construct the indexing graph. One line of work uses LLMs or OpenIE-style extraction to build semantic graphs with explicit entities and relations. HippoRAG and HippoRAG2 extract entities and (subject, predicate, object) triples from each chunk to build schema-free knowledge graphs, and then apply Personalized PageRank over entity, fact, and passage nodes to support associative multi-hop retrieval (Gutiérrez et al., 2024; 2025). LightRAG constructs a graph-enhanced index with entity and relation descriptions and combines low-level and high-level retrieval to improve both local factual access and broader contextual retrieval (Guo et al., 2025). GFM-RAG extends this line by training a general GNN over the extracted graph to enable topology-aware retrieval beyond similarity-based ranking (Luo et al., 2026). A second line of work reduces indexing cost by avoiding heavy LLM relation extraction. For example, LinearRAG builds a relation-free hierarchical index with lightweight entity extraction and semantic linking, avoiding explicit relation-triple generation during graph construction (Zhuang et al., 2026). E^2GraphRAG similarly focuses on efficient index construction with NLP-tool-based entity graphs and bidirectional entity–chunk links, with summaries used only as lightweight retrieval support (Zhao et al., 2025). These lightweight frameworks trade explicit semantic relation modeling for lower construction cost and higher scalability. The two families differ in their cost–quality trade-offs, but both construct graphs primarily from chunk-local evidence: LLM-based frameworks usually extract triples within individual chunks, while lightweight frameworks rely on local entity mentions, co-occurrence, or chunk-level indices. As a result, neither family directly addresses relations that can be recovered only by jointly reading evidence distributed across multiple chunks.

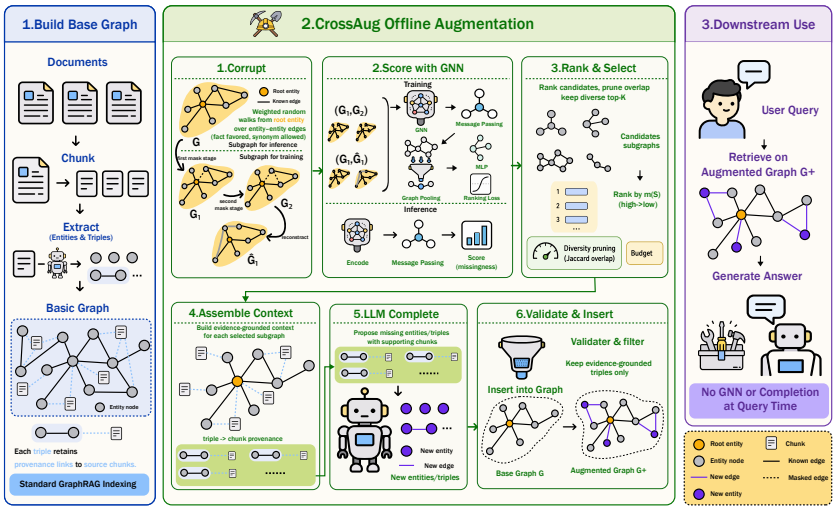

Figure 1: Overview of the CROSSAUG workflow. GNN-based missingness scoring selects incomplete subgraphs, and LLM completion extracts evidence-grounded triples before augmenting the graph.

3 METHOD

3.1 OVERVIEW

Given a long document collection, a standard GraphRAG pipeline splits the text into chunks, extracts entities and triples, and builds a graph index $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ for retrieval. Nodes include entity, passage, and auxiliary nodes; edges include fact edges from extracted triples, synonym edges, and entity–chunk edges. Although this graph supports graph-based retrieval, it remains incomplete because chunk-local extraction misses cross-chunk information. The goal of CROSSAUG is to recover relational information that is supported by the original text but absent from the graph, especially when its evidence is distributed across multiple chunks.

Figure 1 summarizes this three-module workflow. CROSSAUG comprises three modules. First, the *self-supervised training-data construction module* creates paired subgraph views through two-stage fact masking and incorrect fact reconstruction. The masked views train the model to recognize missing relational structure, while the reconstructed views discourage reliance on simple graph-size or density shortcuts. Second, the *topology-aware GNN scoring module* ranks subgraphs by their need for completion rather than predicting specific missing edges. Third, the *GNN-guided LLM completion module* supplies the LLM with the selected subgraphs, their known triples, and their supporting evidence chunks, and inserts the validated evidence-grounded triples into the graph index. This design uses the GNN for structural screening and reserves LLM calls for targeted completion.

3.2 SELF-SUPERVISED TRAINING-DATA CONSTRUCTION

Heterogeneous graph construction. CROSSAUG is designed for GraphRAG frameworks that build graphs from LLM-extracted triples. Such frameworks usually produce an entity-centric graph: entities are represented as nodes, extracted triples induce semantic relation edges between entities, and each relation retains its source chunk as textual provenance. CROSSAUG therefore assumes a minimal graph interface rather than a framework-specific schema: the base index should provide entity nodes, fact edges, and a way to trace extracted facts back to evidence chunks. Each node $v \in \mathcal{V}$ is assigned a semantic feature x_v from the existing retrieval index when available. When the base graph stores edge weights, CROSSAUG uses the raw weight w_e as an edge-strength signal, such as the number of extracted triples supporting a fact edge or the similarity score of an alias/synonym edge; otherwise, unweighted fact edges are assigned a default weight of 1. For each edge $e \in \mathcal{E}$, let $\tau_e \in \mathcal{T}$ denote its edge type, where $\mathcal{T}$ contains the fact, synonym, and entity–chunk types. This abstraction makes the method applicable to LLM-based GraphRAG indices beyond any specific implementation.

Weighted random-walk sampling. Because the full graph can be large, CROSSAUG samples local entity subgraphs around root entities instead of enumerating all entity or chunk pairs. An entity is eligible as a root if it passes a filter removing uninformative content (e.g., purely numeric strings) and has at least one entity neighbor connected by a fact or synonym edge. From each root r, CROSSAUG launches N_{walk}=20 weighted random walks of three entity hops over entity–entity edges; based on our empirical analysis of the base-graph statistics, we use three hops to provide sufficient multi-hop information and report sensitivity to N_{walk} in Appendix E. Let $e = (u, v)$ be a fact or synonym edge with raw graph weight w_e. Its sampling weight is

$$\tilde{w}_e = \min(w_e, c) \cdot \alpha_{\tau_e}, \tag{1}$$

where c clips large edge weights and α_{τ_e} is the multiplier for edge type τ_e. Fact edges are weighted more than synonym edges, so walks prefer semantically informative factual connections while allowing alias/synonym expansion. Thus,

$$P(v \mid u) = \tilde{w}_{(u,v)} / \sum_{v' \in \mathcal{N}(u)} \tilde{w}_{(u,v')}. \tag{2}$$

Here, $\mathcal{N}(u)$ denotes the entity neighbors of u connected by fact or synonym edges. Chunk nodes are not traversed during sampling. Evidence chunks are collected afterward from the provenance of fact edges and attached to their participating entities. This choice keeps the sampled region focused on entity-level relational structure and prevents early expansion into broad passage neighborhoods.

The sampled node set defines an unmasked local subgraph $G = (V_G, E_G)$. CROSSAUG retains only subgraphs with at least the required minimum number of nodes and at least one fact edge. This filtering removes degenerate samples that carry little training signal and ensures that each retained subgraph has enough relational structure for view construction and LLM completion.

Two-stage fact masking. Manually annotating which triples are missing from a long-document graph is expensive. We instead derive training supervision from the sampled subgraph G with fact-edge set F. In the first stage, we sample a fact-edge set $M_1 \subset F$ and mask it to obtain G_1. In the second stage, we sample an additional set $M_2 \subset F \setminus M_1$ and mask it from G_1 to obtain G_2:

$$G_1 = G \setminus M_1, \qquad G_2 = G_1 \setminus M_2 = G \setminus (M_1 \cup M_2). \tag{3}$$

For a per-stage mask ratio ρ_{mask}, each stage targets approximately $\rho_{\text{mask}}|F|$ fact edges. The implementation enforces at least one masked edge per stage and preserves the required minimum number of fact edges in G_2. The evidence-chunk nodes and their entity–chunk connections are retained in both views, allowing the GNN to compare the remaining graph structure with the textual evidence associated with the sampled region. The first ranking pair is $G_2 \succ G_1$, which means G_2 should receive a higher missingness score than G_1. The two views differ only by the fact edges masked in the second stage, so enforcing $G_2 \succ G_1$ supplies the primary supervision for identifying subgraphs that need LLM completion. The two-stage design also keeps the unmasked graph G, which is the object scored at inference, out of the training pairs, so no training view coincides exactly with an inference view even when the two share roots or local neighborhoods. We analyze sensitivity to the fact-mask ratio ρ_{mask} in Appendix E.

Incorrect fact reconstruction. The pair (G_1, G_2) still leaves a structural shortcut open: the scorer could rank views by edge count alone. We therefore use the second-stage masked edges M_2 to construct a third view, denoted by $\widehat{G}_1$. Starting from G_2, each edge in M_2 is replaced by an incorrect fact edge connecting one of its endpoints to a different entity already present in G: $\widehat{G}_1 = G_2 \cup \widetilde{M}_2$, where $\widetilde{M}_2$ contains the incorrect replacement edges. A replacement is rejected if it creates a self-loop, duplicates an existing fact edge, coincides with a synonym edge, or duplicates a replacement already generated for this view. The reconstructed view is retained only when it adds exactly as many fact edges as were masked in the second stage and introduces no new isolated nodes relative to G_1. Consequently, $\widehat{G}_1$ and G_1 contain the same nodes and the same number of edges while differing in which entities are connected. The second ranking pair is $\widehat{G}_1 \succ G_1$. Enforcing $\widehat{G}_1 \succ G_1$ therefore discourages shortcuts based on graph size or density alone and forces the scorer to use entity connectivity together with semantic representations. Two-stage masking produces (G_1, G_2) for missingness learning, while incorrect reconstruction produces $(G_1, \widehat{G}_1)$ for shortcut control.

3.3 TOPOLOGY-AWARE GNN MISSINGNESS SCORING

GNN encoder. For every sampled subgraph view, CROSSAUG constructs a heterogeneous graph representation for message passing with node features, typed edges, and node-type identifiers. Let x_v denote the input semantic feature vector of node v, and let W_x be a learned projection matrix that maps input features into the GNN hidden space. Because the textual content of entity nodes and chunk nodes is encoded by the same embedding model, the two node types yield features of the same dimensionality in a shared vector space and can therefore share the projection matrix W_x. CROSSAUG initializes each node representation by combining this projected feature with a learned embedding of the node type:

$$h_v^{(0)} = W_x x_v + \eta_{\phi(v)}, \tag{4}$$

where $\phi(v)$ denotes the node type of v and $\eta_{\phi(v)}$ is its learned type embedding. The node representations are then updated through L layers of message passing; we use $L{=}2$ in all experiments. For relational GNNs, the update uses edge-type-specific transformations:

$$h_v^{(\ell+1)} = \mathrm{GELU}\left(\sum_{\tau\in\mathcal{T}} \sum_{u\in\mathcal{N}_\tau(v)} W_\tau^{(\ell)} h_u^{(\ell)}\right), \tag{5}$$

where $\ell \in \{0, \ldots, L-1\}$ denotes the message-passing layer, $\tau \in \mathcal{T}$ is an edge type, $\mathcal{N}_\tau(v)$ denotes the neighbors of node v connected by edges of type τ, u is a node in this typed neighborhood, and $W_\tau^{(\ell)}$ is the layer-specific transformation matrix for type τ. Evidence chunks therefore affect entity representations through message passing, although entity-subgraph sampling does not traverse them.

Graph-level missingness prediction. For the unmasked sampled subgraph G used at inference, let V_G^{ent} denote its entity-node set. CROSSAUG aggregates only the updated representations of these nodes using mean pooling and maps the result to a scalar missingness score:

$$\mathrm{score}(G) = \mathrm{MLP}\left(\frac{1}{|V_G^{\mathrm{ent}}|} \sum_{v\in V_G^{\mathrm{ent}}} h_v^{(L)}\right). \tag{6}$$

Here, $h_v^{(L)}$ is the representation of node v after the final message-passing layer. During training, the same scoring function is applied to G_1, G_2, and $\widehat{G}_1$. Chunk nodes contribute indirectly by passing messages to entities, but they are excluded from the graph-level pooling mask. The resulting subgraph representation is permutation invariant.

Pairwise ranking objective. Let $\mathcal{P}_{\mathrm{missing}}$ contain all valid (G_1, G_2) training pairs, and let $\mathcal{P}_{\mathrm{recon}}$ contain all valid $(G_1, \widehat{G}_1)$ training pairs. Let $\sigma(a) = 1/(1+\exp(-a))$ denote the logistic sigmoid function. We directly optimize the two margin-free log-ranking losses:

$$\begin{aligned}
\mathcal{L}_{\mathrm{missing}} &= -\tfrac{1}{|\mathcal{P}_{\mathrm{missing}}|} \textstyle\sum_{(G_1,G_2)\in\mathcal{P}_{\mathrm{missing}}} \log\sigma(\mathrm{score}(G_2) - \mathrm{score}(G_1)), \\
\mathcal{L}_{\mathrm{recon}} &= -\tfrac{1}{|\mathcal{P}_{\mathrm{recon}}|} \textstyle\sum_{(G_1,\widehat{G}_1)\in\mathcal{P}_{\mathrm{recon}}} \log\sigma\Big(\mathrm{score}(\widehat{G}_1) - \mathrm{score}(G_1)\Big).
\end{aligned}$$

The final objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{missing}} + \lambda\mathcal{L}_{\mathrm{recon}}, \tag{7}$$

The objective thus optimizes relative missingness directly, without requiring an absolute label for any single subgraph. Here, λ controls the strength of the shortcut-suppression signal provided by the reconstruction loss $\mathcal{L}_{\mathrm{recon}}$.

Dynamic training and inference selection. At each training epoch, CROSSAUG randomly samples root-centered neighborhoods from the eligible roots and applies two-stage masks and incorrect fact reconstruction. The views G_1, G_2, and $\widehat{G}_1$ within a training group always share the same sampled graph G. This dynamic construction exposes the GNN to different local neighborhoods and training views while preventing differences in random-walk coverage from defining a ranking pair.

At inference time, the trained GNN scores each sampled candidate $G^{(i)}$ and ranks it by $\mathrm{score}(G^{(i)})$. Candidates with high node-level overlap with already-selected subgraphs—measured by Jaccard

similarity $J(G^{(i)}, G^{(j)}) = |V_{G^{(i)}} \cap V_{G^{(j)}}|/|V_{G^{(i)}} \cup V_{G^{(j)}}|$—are skipped to avoid redundant LLM calls on nearly identical evidence. Selection continues until the limit set by the LLM completion budget ratio ρ_{budget} is reached. The GNN module thereby replaces exhaustive enumeration of all possible cross-chunk combinations with targeted selection of a diverse, high-missingness subgraph set. This directs LLM completion toward the contexts where it is most likely to recover useful relational information, improving cost efficiency.

3.4 GNN-Guided LLM Completion and Graph Augmentation

For each selected subgraph, CrossAug assembles a context that includes the subgraph's root entity, the full list of entities in the local region, the triples already extracted for that region, and the evidence passages retrieved from the provenance of the participating fact edges. The GNN identifies regions that are likely to be incomplete, while the LLM performs completion over the actual entities, triples, and evidence passages in those regions.

The LLM is instructed to propose new entities and relations that are directly stated or unambiguously supported by the provided passages. Every proposed entity and relation must include a list of supporting chunk identifiers drawn from the evidence set; this citation requirement enforces cross-chunk grounding and prevents hallucinated graph expansion. Relations without valid chunk citations are discarded. The validated triples are then inserted into the base graph index using the same schema as the original extractor, so the augmented graph is structurally consistent with the base index and immediately compatible with any downstream retrieval pipeline. Because all augmentation is performed offline, the enriched graph serves as a static index for subsequent retrieval and question answering; neither the LLM completion step nor the GNN needs to be re-invoked at query time.

4 Experiments

Due to space limitations, we defer the cost and parameter-sensitivity analyses to Appendices D–E.

4.1 Experimental Settings

Datasets and Baselines. We evaluate our method on four benchmarks: three widely used multi-hop QA benchmarks—MuSiQue (Trivedi et al., 2022), 2WikiMultiHopQA (Ho et al., 2020), and HotpotQA (Yang et al., 2018)—and LiteraryQA (Bonomo et al., 2025), a long-document QA benchmark over literary texts. Following previous work (Gutiérrez et al., 2024), we sample 1,000 questions from each multi-hop benchmark to control experimental cost. For LiteraryQA, we use the first 50 books of the test set, comprising 1,358 queries in total; most books in this set exceed the context window of the backbone model. Detailed dataset statistics are provided in Appendix A.

We apply CrossAug to three GraphRAG frameworks that construct their graphs through LLM-based extraction: LightRAG (Guo et al., 2025), HippoRAG2 (Gutiérrez et al., 2025), and GFM-RAG (Luo et al., 2026). For each, we compare the original base framework against its CrossAug-augmented variant. We additionally report LinearRAG (Zhuang et al., 2026) for broader context; because its graph construction does not rely on LLM extraction, CrossAug does not apply to it.

Base Models. We adopt Qwen3-32B (Yang et al., 2025) as the backbone LLM; its 32,768-token context window provides sufficient capacity for the long evidence passages retrieved across our benchmarks. BGE-M3 (Chen et al., 2024) serves as the embedding model, and BAAI-bge-reranker-v2-m3 is used as the reranker where applicable. We set k=5 for top-k retrieval across all frameworks.

Metrics. We evaluate using Exact Match (EM) and F1, the standard metrics in this line of QA research (Gutiérrez et al., 2024; Guo et al., 2025; Luo et al., 2026).

4.2 Experimental Results

We first evaluate end-to-end QA accuracy. Within each framework, the base and CrossAug rows differ only in the graph index, with all retrieval settings held fixed. Table 1 reports means and standard deviations over three random seeds. CrossAug improves all 24 framework–dataset–metric

Table 1: Main QA results (mean ± standard deviation over three seeds). Bold denotes the best result in each metric.

Framework	Setting	MuSiQue		2Wiki		HotpotQA		LiteraryQA	
		EM	F1	EM	F1	EM	F1	EM	F1
LinearRAG	–	24.12±0.01	37.40±0.39	46.53±0.28	**65.05±0.18**	46.93±0.48	65.17±0.28	11.26±0.21	26.29±0.54
GFM-RAG	Base	19.20±0.62	29.88±0.44	52.57±0.32	59.98±0.37	39.23±0.21	49.76±0.24	12.97±0.34	27.71±0.22
	CROSSAUG	20.30±0.20	30.68±0.63	**54.00±0.26**	61.57±0.15	39.57±0.21	50.62±0.21	14.31±0.10	28.77±0.33
LightRAG	Base	20.73±0.21	28.42±0.34	28.00±0.44	36.65±0.29	45.23±0.45	58.69±0.15	11.24±0.15	26.11±0.11
	CROSSAUG	21.17±0.51	28.69±0.40	28.73±0.49	37.12±0.15	45.63±0.68	58.94±0.74	11.41±0.25	26.33±0.29
HippoRAG2	Base	27.93±0.25	39.74±0.26	51.23±0.06	57.63±0.17	54.10±0.26	68.35±0.42	17.47±0.11	32.68±0.37
	CROSSAUG	**29.53±0.55**	**41.19±0.57**	51.57±0.15	57.93±0.25	**54.60±0.10**	**68.64±0.07**	**18.23±0.56**	**33.83±0.59**

comparisons. The gains are generally larger on MuSiQue and LiteraryQA, the two more challenging benchmarks, suggesting that cross-chunk augmentation is particularly useful for demanding multi-hop and long-document reasoning.

Fine-grained Retrieval Analysis. We further examine where these gains originate by classifying every question whose EM outcome changes: false-to-true (F2T) cases, in which the base framework fails and the CROSSAUG-augmented variant succeeds, and true-to-false (T2F) cases, in which the reverse occurs. The left panel of Figure 2 decomposes the F2T cases by retrieval effect. *Order-only* cases leave the retrieved chunk set unchanged and merely re-rank it, promoting more useful evidence to positions where the LLM attends to it more. The next two categories involve a changed set: *Set-only* introduces non-gold bridging chunks that still support the reasoning chain, whereas *New gold* brings a gold chunk into the top-k. *Hidden facts* applies only to LightRAG, which inserts retrieved facts directly into the QA prompt, so CROSSAUG-extracted facts can serve as compact relational evidence in their own right; *Other* covers the remainder. LightRAG's gains therefore combine changed chunk retrieval with directly injected facts.

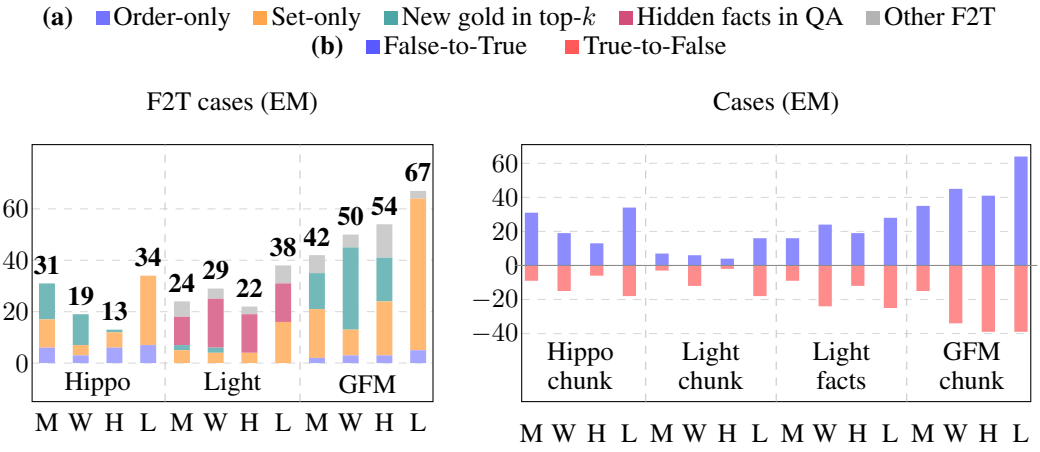

Figure 2: Fine-grained EM transitions. Left: F2T cases by retrieval effect. Right: F2T and T2F cases by change source. Labels Hippo/Light/GFM denote HippoRAG2/LightRAG/GFM-RAG; M/W/H/L denote MuSiQue/2Wiki/HotpotQA/LiteraryQA.

The right panel compares F2T with T2F cases by change source. For HippoRAG2 and GFM-RAG, chunk changes yield more F2T than T2F cases on every dataset. For LightRAG, the hidden facts injected into the prompt are favorable or balanced, but its chunk changes yield more T2F cases on 2Wiki and LiteraryQA. The benefit is therefore not uniform across change sources.

4.3 CASE STUDY

The aggregate gains leave open how they arise on individual questions, which we examine by tracing questions through the base and augmented indices. We present representative cases illustrating CROSSAUG's advantages in retrieval and downstream QA. We focus here on a cross-chunk locative

case in which CROSSAUG recovers a relation that no chunk-local triple can express, and give additional cases in Appendix C.

Recovering a cross-chunk locative relation. In *Voodoo Planet*, the question asks: *"Where does the confrontation between the Medic and Lumbrilo take place?"* The gold answer is *in a deadly swamp*. The base framework answers *"on a camping ground"*, whereas CROSSAUG answers that the confrontation occurs in the Khatkan poachers' camp within a swamp. The base graph contains locally correct triples such as *(Tau, approached, Lumbrilo)*, *(Tau, defeated, Lumbrilo)*, *(Lumbrilo, practices, new magic)*, and *(Voodoo Planet, features, showdown between Lumbrilo and Asaki)*. Even when these triples are connected through multi-hop reasoning, they trace only an event chain centered on the conflict between Tau and Lumbrilo, without specifying where the confrontation takes place. The base framework thus relies on incomplete context and misidentifies the location.

CROSSAUG extracts cross-chunk locative facts that connect the confrontation directly to the missing location, including *(khatkan swamp, location of battle, lumbrilo vs tau and dane)* and *(khatkan swamp, location of, final confrontation with lumbrilo)*. These facts are supported jointly by the chunk describing the swamp and camp and the chunks describing the battle. Once these cross-chunk locative links are added to the graph index, retrieval surfaces the relevant swamp and confrontation passages:

> "The shadow thing in the swamp moved … 'I'd rather get at the source.' There was a grim note in the Medic's reply. 'And to do that I want to look at their camp."'

This case illustrates the core failure mode targeted by CROSSAUG: the base graph contains true local event triples, but the cross-chunk relation needed to answer the question is absent from the index.

4.4 VALIDATION OF AUGMENTED TRIPLE QUALITY

We next validate the quality of the triples produced by CROSSAUG. To validate the quality of the triples produced by CROSSAUG, we randomly sampled 1,000 augmented triples from each dataset under HippoRAG2+CROSSAUG and used DeepSeek-V4-Pro as an automatic judge of factual correctness and chunk-citation correctness. The two criteria are judged independently: factual correctness asks whether the proposed relation is supported by the document, whereas chunk-citation correctness asks whether the cited chunk itself provides direct evidence for that relation. The distinction matters because a plausible triple can still carry an inadequate citation, while a relevant passage does not by itself guarantee that the extracted relation is correct.

We independently validated the judge on 200 sampled triples annotated by five non-author master's and doctoral students with relevant expertise and English proficiency. The annotators received a shared task description and representative examples, and their aggregated labels were compared with the automatic judgments using both F1 and Cohen's κ. Table 2 shows high correctness rates on both criteria and strong human–LLM agreement, particularly for factual correctness.

Table 2: Validation of the DeepSeek judge on generated triples; agreement metrics use 200 human-annotated samples. FC and CC denote factual and chunk-citation correctness.

Criterion	DeepSeek (%)	Human–LLM F1	Cohen's κ
FC	84.35	0.957	0.881
CC	81.54	0.887	0.751

This agreement supports using the judge as a scalable proxy for corpus-level quality analysis, and the absolute correctness rates indicate that most inserted triples are both factually supported and traceable to their cited evidence. However, this evaluation assesses the precision and provenance quality of the recovered triples rather than the completeness of recovery, and automatic validation may still be affected by judge bias or residual graph pollution.

Table 3: Missingness-type validation over 500 sampled, factually correct CROSSAUG triples. Counts and ratios use 497 valid judge annotations.

Missingness Type	Count	Ratio
Missing relation/fact	154	31.0%
Missing entity	117	23.5%
Partial information extraction	53	10.7%
Alias/coreference issue	46	9.3%
Relation-type refinement/error	85	17.1%
Other/ambiguous	42	8.5%

Observed Missingness Types. The two synthetic training patterns should resemble the in-

completeness that real indices exhibit, which we verify by categorizing what CROSSAUG recovers. We further analyze 500 CROSSAUG-recovered triples sampled from those judged factually correct in the preceding evaluation. For each triple, the judge compares it with the base graph and the supporting passages and assigns one dominant missingness type. Table 3 shows that missing facts and missing entities are the two most frequent patterns, together accounting for 54.5% of the valid annotations. Relation-type errors, partial extraction, and alias/coreference issues form further non-negligible groups. Although the scorer is trained only with fact masking and incorrect fact reconstruction, these generic constructions capture structural symptoms shared by the broader set of observed failure patterns. They do not cover the full error distribution, and targeted modeling of specialized or ambiguous errors remains future work.

4.5 ABLATION STUDY

We next study the importance of main components in CROSSAUG. First, the gains could stem from the extra LLM calls alone rather than from the scorer. We therefore replace the scorer with random subgraph selection under the same ρ_{budget}, so both settings issue the same number of calls and differ only in which subgraphs receive them. As shown in Table 4, this replacement degrades both EM and F1 on MuSiQue and LiteraryQA. The drops are 1.70 EM and 1.82 F1 on MuSiQue and 0.59 EM and 1.14 F1 on LiteraryQA. Topology-guided scoring therefore selects subgraphs whose completion is more useful downstream than a random selection of the same size.

Table 4: Ablation studies on HippoRAG2+ CROSSAUG using MuSiQue and LiteraryQA. The full CROSSAUG configuration is reported once; each remaining row changes one design component with the rest of the pipeline fixed. The w/o GNN variant replaces scorer-based selection with random selection under the same completion budget.

Component	Variant	MuSiQue		LiteraryQA	
		EM	F1	EM	F1
Full model	CROSSAUG	**30.10**	**41.77**	**18.78**	**34.31**
Selection	w/o GNN	28.40	39.95	18.19	33.17
Feature	Semantic only	28.10	40.03	17.82	32.90
	Topology only	29.40	41.33	18.11	33.65
Projection	Type-specific	28.20	40.19	17.75	33.35
Objective	$\mathcal{L}_{\text{missing}}$ only	29.30	41.28	18.40	33.81
	$\mathcal{L}_{\text{recon}}$ only	28.90	40.63	18.26	33.52

We next ask which design choices give the scorer that advantage. The remaining ablation rows in Table 4 vary three of them in turn. In the feature ablation, *semantic only* removes graph message passing, whereas *topology only* removes the semantic embeddings and retains the graph structure. *Topology only* outperforms *semantic only* on both datasets, yet both variants fall below the full CROSSAUG model, which combines the two sources, confirming that they are complementary. For projection, full CROSSAUG uses a shared projection and outperforms the *type-specific* variant on both datasets. Because entity and chunk texts pass through the same encoder, sharing the projection preserves their aligned feature space, while the node-type embeddings keep the two types distinguishable. The objective ablation separates the two ranking losses: $\mathcal{L}_{\text{missing}}$ alone outperforms $\mathcal{L}_{\text{recon}}$ alone on both datasets, confirming that missingness learning supplies the primary supervision, while both single-loss variants fall below the full combined objective, indicating that $\mathcal{L}_{\text{recon}}$ contributes complementary shortcut suppression.

5 CONCLUSION

We presented CROSSAUG, a GNN-guided cross-chunk graph augmentation method that addresses the chunk-local extraction bottleneck in GraphRAG. CROSSAUG operates through three modules: a self-supervised training-data construction module that derives supervision directly from the base graph without manual annotation, a topology-aware GNN scoring module that identifies structurally incomplete subgraphs, and a GNN-guided LLM completion module that produces evidence-grounded triples for the selected regions. Applied to three LLM-based GraphRAG frameworks across four QA benchmarks, CROSSAUG improves performance across all 24 framework–dataset–metric comparisons. Because LLM completion is restricted to a targeted subset of high-missingness subgraphs, the resulting augmented graph is produced once and can then be reused by any downstream retrieval pipeline without further recomputation.

ETHICS STATEMENT

CROSSAUG augments GraphRAG indices with LLM-generated triples. Despite our citation-grounding filter and the correctness rates above 81% reported in Section 4.4, residual extraction errors may propagate to downstream QA, and practitioners should verify the augmented triples in high-stakes settings. All four English QA benchmarks (MuSiQue, 2WikiMultiHopQA, HotpotQA, LiteraryQA) and all models used (Qwen3-32B, BGE-M3, BAAI-bge-reranker-v2-m3, DeepSeek-V4-Pro) are publicly released for research and are used in a manner consistent with their original intended purposes under their respective licenses. The underlying corpora are drawn from Wikipedia and public-domain literary works and contain no private personal information. Dataset statistics, train/test splits, and graph statistics are reported in Table 5; the model configurations, GNN scorer ($\sim 10^6$ parameters), A800-based infrastructure, and full indexing-time compute budget are reported in Appendix D and Table 6. The human validation involved five non-author graduate students who provided informed consent after receiving a shared task description; the task used only public text and collected no personal or sensitive information. We release the source code under a permissive open-source license.

AI USE STATEMENT

AI assistants were used only for editorial polishing and routine code suggestions, with all scientific claims authored and verified by the human authors.

REPRODUCIBILITY STATEMENT

Section 3 describes the graph construction, sampling, training objective, and inference procedure, and Section 4.1 specifies the datasets, baseline frameworks, backbone models, and evaluation metrics. Appendix B provides the completion prompt; Appendices D and E report the budget and hyperparameter analyses; and Appendix F documents the human evaluation protocol. These descriptions and the anonymous code link in the abstract are provided to support reproduction of our experiments.

REFERENCES

Tommaso Bonomo, Luca Gioffré, and Roberto Navigli. Literaryqa: Towards effective evaluation of long-document narrative qa. In *Proceedings of the 2025 conference on empirical methods in natural language processing*, pp. 34074–34095, 2025.

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. *arXiv preprint arXiv:2402.03216*, 4(5), 2024.

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.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. Pmlr, 2017.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tian Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. In *EMNLP (Findings)*, pp. 10746–10761, 2025.

Bernal J Gutiérrez, 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.

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'25. JMLR.org, 2025.

Haoyu Han, Li Ma, Yu Wang, Harry Shomer, Kai Guo, Yongjia Lei, Zhisheng Qi, Zhigang Hua, Bo Long, Hui Liu, et al. Rag vs. graphrag: A systematic evaluation and key insights. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 8966–8977, 2026.

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.

Linhao Luo, Zicheng Zhao, Reza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. Gfm-rag: graph foundation model for retrieval augmented generation. *Advances in Neural Information Processing Systems*, 38:36371–36405, 2026.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pp. 593–607. Springer, 2018.

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

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

Chang Yang, Chuang Zhou, Yilin Xiao, Su Dong, Luyao Zhuang, Yujing Zhang, Zhu Wang, Zijin Hong, Zheng Yuan, Zhishang Xiang, et al. Graph-based agent memory: Taxonomy, techniques, and applications. *arXiv preprint arXiv:2602.05665*, 2026.

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.

Yibo Zhao, Jiapeng Zhu, Ye Guo, Kangkang He, and Xiang Li. Eˆ 2graphrag: Streamlining graph-based rag for high efficiency and effectiveness. *arXiv preprint arXiv:2505.24226*, 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. In *International Conference on Learning Representations*, volume 2026, pp. 147053–147075, 2026.

A Dataset Statistics

Table 5 reports detailed statistics of the base and augmented graph indices.

Table 5: Dataset-level graph statistics for the base GraphRAG index and the CrossAug-augmented index. "2Wiki" denotes 2WikiMultiHopQA.

Statistic	LiteraryQA	MuSiQue	2Wiki	HotpotQA
Chunks	9,550	11,656	6,119	9,811
Base nodes	68,014	104,190	54,487	84,632
Base edges	225,908	331,233	169,455	262,256
Base triples	103,148	123,308	64,523	98,592
Added nodes	3,203	2,994	1,569	1,718
Added edges	27,284	20,093	9,781	9,839
Added triples	8,707	8,901	5,048	5,252

B PROMPT

Figure 3 presents the full prompt used for GNN-guided LLM completion.

Subgraph Completion Prompt

```
Subgraph incompleteness hint:
{
 "possible_missingness_types": [
  "missing_relations",
  "missing_entities",
  "missing_relations_and_entities"
 ],
 "instruction": "Treat this sampled subgraph as potentially incomplete.
Look for important gaps in both relations and entities, not just shallow
restatements of the known triples."
}

Subgraph root entity:
<ROOT_ENTITY>

Existing entities in the sampled subgraph:
<ENTITY_LIST>

Known triples in this sampled subgraph:
<KNOWN_TRIPLES>

Evidence passages:
<EVIDENCE_PASSAGES>

Silently follow this extraction strategy before producing the final JSON:

1. Summarize the current subgraph coverage.

2. Identify likely missing relations, likely missing entities, or both.

3. Prioritize deeper completions that:
  - connect multiple known triples;
  - resolve references across multiple passages; or
  - enrich the subgraph with important time, condition, prerequisite,
    ownership, role, sequence, or part-whole information.

4. When a new triple depends on multiple passages, attach all relevant
  chunk_ids instead of citing only one chunk.

5. Keep only facts that are directly stated or unambiguously supported by
the
  evidence passages.

Required output schema:
{
 "new_triples": [
  {
   "triple": ["subject", "predicate", "object"],
   "chunk_ids": ["chunk-id"]
  }
 ]
}

Return JSON only.
```

Figure 3: Prompt used by CROSSAUG for GNN-guided LLM completion. The selected subgraph, known triples, candidate entities, and evidence chunks are provided to the LLM, which is instructed to return only evidence-supported triples and entities.

C CASE STUDY

We provide additional representative examples in which CROSSAUG improves downstream QA performance. Each case illustrates a distinct failure mode of the base framework—misidentified entities, absent relational links, or insufficiently ranked evidence—and shows how cross-chunk augmentation addresses it by enriching the graph index before retrieval.

Recovering the correct object in long-document QA. In *Immensee*, the question asks: *"What did Erich give Elisabeth?"* The gold answer is *a bird* or *a new bird*. The base framework answers *"The gilded birdcage"*, while CROSSAUG answers *"A canary"*. The base retrieval returns chunks 0025, 0026, 0013, 0021, and 0027, which supply related narrative context but do not identify the gift. CROSSAUG instead generates hidden relational facts in the same narrative neighborhood, such as *(Eric, is, friend of Elisabeth)* and *(Elisabeth, accepted proposal from, Eric)*. These triples do not state the answer, but they make the Erich–Elisabeth relation explicit in the graph and shift retrieval toward the passage describing Erich's gift. As a result, CROSSAUG replaces chunk 0027 with chunk 0014, which already contains the base-extracted fact *(Eric, sent canary to, Elisabeth)* and the decisive textual evidence:

> "Elisabeth was standing by the window and sticking some fresh chick-weed in a gilded birdcage which he had not seen in the place before. In the cage was a canary … 'Your friend Eric sent it this noon from his estate as a present for Elisabeth."'

This example illustrates a common long-document failure mode in which the retrieved context contains a salient object near the answer but not the answer-defining relation. The birdcage is strongly associated with both the scene and Elisabeth, making it a plausible distractor for the base framework. Knowing that Elisabeth stands near a birdcage, or that Erich is associated with her elsewhere in the story, is not sufficient to infer what he gave her. The improvement therefore comes from a two-step effect: CROSSAUG first strengthens the graph neighborhood connecting giver and recipient, and the modified graph then surfaces chunk 0014, where the canary is explicitly linked to Erich as sender and Elisabeth as recipient. By shifting retrieval to this chunk, CROSSAUG prevents the LLM from answering with a nearby container and grounds the answer in the entity that actually satisfies the gift relation.

Using hidden facts to bridge retrieval. On MuSiQue, the question asks: *"What was the date that Pam got married in The Office?"* The gold answer is *October 8, 2009*. The base framework answers that the date is not mentioned in the provided information, whereas CROSSAUG answers *"October 8, 2009."* The base retrieval mainly returns general *Pam Beesly* and *The Office* passages. These passages state that Jim and Pam marry in the episode *Niagara*, but they do not directly surface the episode date:

> "Jim and Pam marry early in the season, at Niagara Falls, during the highly anticipated, hour long episode, 'Niagara'."

CROSSAUG extracts hidden facts that connect the marriage event to the episode, including *(niagara the office, features, pam beesly and jim halpert s marriage)* and *(the episode, features, jim and pam s marriage at niagara falls)*. These hidden facts add a more direct graph link between the marriage event and the episode. As a result, retrieval includes the *Niagara* episode page, which contains the exact answer:

> "Niagara" *The Office* episode … Original air date October 8, 2009.

The base triples can connect *Pam*, *Jim*, and the wedding event, but they do not reliably bridge that event to the episode page that stores the air date. This case shows that CROSSAUG does more than add triples to the index: the completed graph changes the starting points of graph retrieval, allowing downstream QA to reach evidence that the base graph fails to use.

Correcting an entity through better evidence retrieval. In *Youth*, the question asks: *"What is the name of the steamer that agrees to tow the crew?"* The gold answer is *Sommerville*. The base framework answers *"The Judea"*, confusing the damaged ship with the steamer that tows it.

CROSSAUG answers *"The Sommerville."* The base retrieval surfaces narrative passages from *Youth*, including chunks 0001, 0006, 0016, 0012, and 0027. These passages mention the ship *Judea* and earlier accidents but do not identify the towing steamer. CROSSAUG instead retrieves chunks 0020 and 0021, where the towing event is explicitly described:

> "When our skipper came back we learned that the steamer was the Sommerville, Captain Nash ... and that the agreement was she should tow us to Anjer or Batavia, if possible ... At noon the steamer began to tow."

The following chunk continues the same event:

> "The speed of the towing had fanned the smoldering destruction ... 'We had better stop this towing'."

The base answer is plausible precisely because *Judea* is the central vessel throughout the narrative. The question, however, asks for the steamer that performs the towing, not for the ship being towed. The base evidence links a prominent entity to the question without resolving the relational role that the question specifies. CROSSAUG retrieves the event-local passages that explicitly distinguish the vessel doing the towing from the one being towed.

Improving evidence ranking with a hidden date relation. Another MuSiQue example asks: *"What date saw the writing of the song where the devil went down to the state where WDXQ is located?"* The gold answer is *May 21, 1979*. The base framework answers *"1978"*, while CROSSAUG answers *"May 21, 1979."* Both frameworks retrieve passages about *The Devil Went Down to Georgia* and WDXQ, but CROSSAUG moves the most useful evidence from rank 2 to rank 1. The retrieved WDXQ passage establishes the state:

> "WDXQ (1440 AM) is a radio station licensed to Cochran, Georgia, United States."

The decisive song passage states:

> "The Devil Went Down to Georgia" ... Released May 21, 1979 ... Songwriter(s) Charlie Daniels, Tom Crain, "Taz" DiGregorio, Fred Edwards, Charles Hayward, James W. Marshall.

CROSSAUG introduces the hidden fact *(the devil went down to georgia, released in, may 21 1979)*, which links the song directly to its release date in the graph index. The base evidence can connect WDXQ to Georgia and the song to adjacent release context, but without the precise date relation the model anchors on an approximate nearby year such as *1978*. This example highlights a benefit distinct from the preceding cases: CROSSAUG need not introduce an entirely new document into retrieval. By anchoring retrieval to the hidden date triple, it promotes already-relevant evidence to a higher rank, letting the LLM ground its answer in the passage that contains the exact date.

D COST AND PERFORMANCE ACROSS PARAMETER SETTINGS

Since CROSSAUG adds LLM calls to indexing, we report both the size of this one-time overhead and how accuracy responds when it is varied. CROSSAUG exposes an LLM completion budget ratio ρ_{budget} that controls how many candidate subgraphs receive LLM completion calls during indexing. Table 6 reports the indexing cost at the default $\rho_{\mathrm{budget}}{=}0.10$ on MuSiQue. The HippoRAG2 base pipeline uses 10.84M prompt tokens, 27.22M completion tokens, and 23,312 LLM calls. On top of this pipeline, CROSSAUG adds 2,331 calls, 11.80M prompt tokens, and 4.80M completion tokens—10.0% of the base call count and 43.6% of its total token volume. Each CROSSAUG prompt is substantially longer than a base-pipeline prompt because it carries a local subgraph, multiple evidence passages spanning different chunks, the known triples, and entity context. The completion overhead is comparatively small, since the model outputs only evidence-supported triples.

Figure 4 shows how EM, F1, and completion-token cost respond as ρ_{budget} varies from 0.05 to 0.20 on MuSiQue. Among the tested settings, the default $\rho_{\mathrm{budget}}{=}0.10$ achieves the highest EM (30.10) and F1 (41.77). Reducing ρ_{budget} to 0.05 lowers completion-token usage to 2.52M but also

reduces EM and F1 to 28.50 and 39.94. Increasing ρ_{budget} to 0.15 or 0.20 consumes more tokens yet scores below the default on both metrics. Among the tested values, ρ_{budget}=0.10 is therefore the best operating point.

Table 6: Indexing-cost comparison on MuSiQue between the HippoRAG2 base pipeline and CROSSAUG. The CROSSAUG row reports the additional overhead introduced by CROSSAUG on top of the base pipeline. Prompt and completion token counts are reported in millions of tokens ($\times 10^6$).

Setting	**Prompt** ($\times 10^6$)	**Completion** ($\times 10^6$)	**LLM Calls**
Base	10.84	27.22	23,312
CROSSAUG (additional)	11.80	4.80	2,331

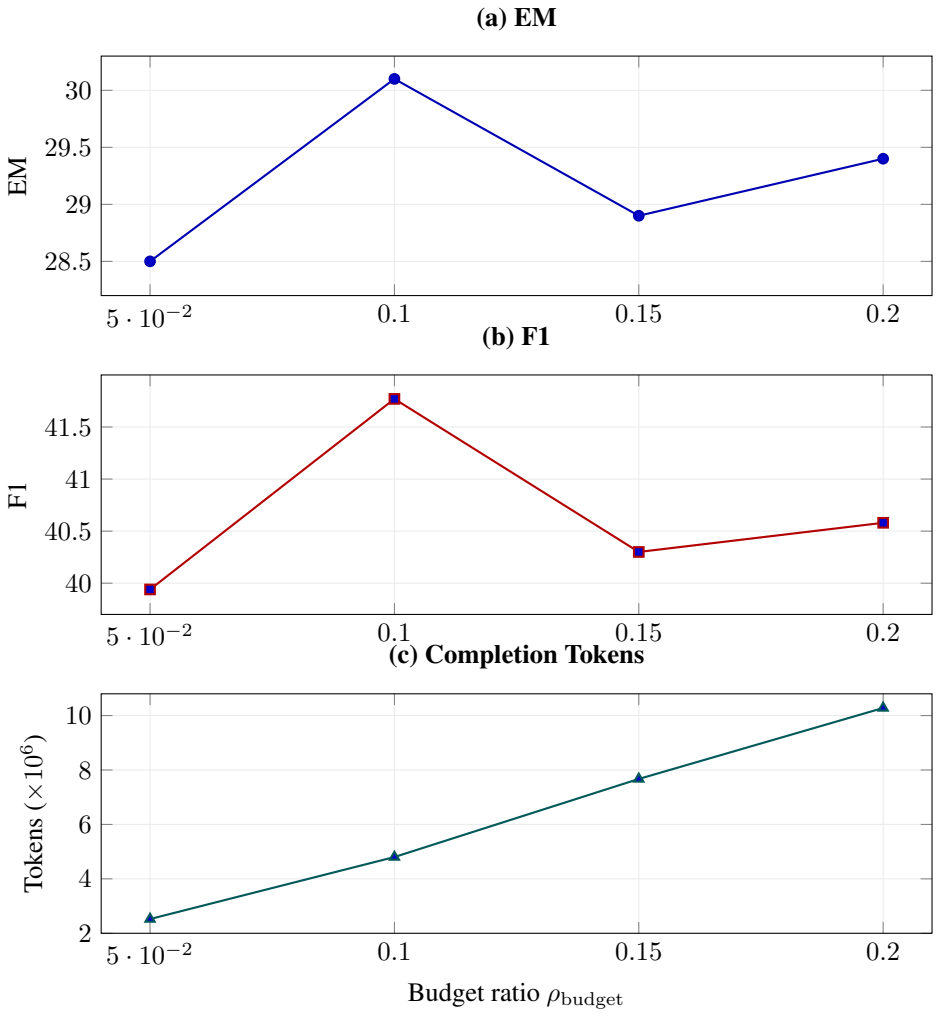

Figure 4: Sensitivity to the LLM completion budget ratio ρ_{budget} on MuSiQue. Panels report EM, F1, and completion-token usage.

E HYPERPARAMETER SENSITIVITY ANALYSIS

The budget ratio controls how much completion is performed, but not how well the scorer ranks candidates. That ranking depends on training, so we evaluate three GNN-training hyperparameters

on HippoRAG2+CROSSAUG using MuSiQue. We vary one hyperparameter at a time and otherwise use the default configuration: N_{walk}=20 random walks per root, weight λ=5 on the reconstruction loss $\mathcal{L}_{\text{recon}}$, and per-stage fact-mask ratio ρ_{mask}=0.20. Overall, performance remains stable under moderate variations around these defaults, indicating limited sensitivity within the practical range.

Table 7: Sensitivity to the number of random walks per root, N_{walk}, on HippoRAG2+CROSSAUG using MuSiQue.

N_{walk}	**EM**	**F1**
10	29.00	40.75
20	**30.10**	**41.77**
30	29.30	41.06
40	28.60	40.37
50	28.40	39.81
60	28.30	40.17

Table 7 shows that N_{walk}=20 performs best on both metrics. Increasing N_{walk} beyond 20 does not improve performance, indicating that 20 random walks per root are sufficient for effective subgraph sampling.

Table 8: Sensitivity to the weight λ on the reconstruction loss $\mathcal{L}_{\text{recon}}$ for HippoRAG2+CROSSAUG using MuSiQue. "Recon only" uses only $\mathcal{L}_{\text{recon}}$.

λ	**EM**	**F1**
0.1	27.30	39.08
1	28.10	39.68
5	**30.10**	**41.77**
10	29.80	41.39
15	29.60	40.79
20	**30.10**	41.48
100	29.60	41.42
Recon only	28.90	40.63

As shown in Table 8, λ=5 achieves the highest F1 and ties λ=20 for the highest EM. The *Recon only* setting is weaker than the default on both metrics, which indicates that $\mathcal{L}_{\text{missing}}$ and $\mathcal{L}_{\text{recon}}$ are best used jointly rather than in isolation.

Table 9: Sensitivity to the per-stage fact-mask ratio ρ_{mask} on HippoRAG2+CROSSAUG using MuSiQue.

Fact mask ρ_{mask}	**EM**	**F1**
0.05	29.70	41.30
0.15	30.00	41.42
0.20	**30.10**	**41.77**
0.25	**30.10**	41.55
0.30	30.00	41.54
0.35	29.90	41.31

Table 9 shows that the default ratio ρ_{mask}=0.20 achieves the highest F1 and ties ρ_{mask}=0.25 for the highest EM. Performance remains relatively stable from 0.15 to 0.35, while the smaller ratio of 0.05 is weaker on both metrics.

F HUMAN EVALUATION DETAILS

This appendix documents the protocol for the human evaluation summarized in Section 4.4 of the main text. The evaluation has two purposes: to estimate the absolute quality of CROSSAUG-augmented

triples and to verify that the DeepSeek-V4-Pro judge produces decisions consistent with informed human raters.

Annotator profile. Five non-author master's and doctoral students with relevant domain expertise and English proficiency served as annotators. Before annotation, each received a shared description of the task, representative high-quality examples, and an explanation of the study objectives, and each provided informed consent.

Sampling. From the pool of CROSSAUG-augmented triples produced by HippoRAG2+CROSSAUG across the four benchmarks, we sampled 200 triples for independent human validation.

Rubric. Each item presented to an annotator included the triple in *(subject, relation, object)* form, the cited chunk identifier(s), and the full chunk text. Annotators returned a strict binary judgment on two independent criteria:

- **Factual correctness**—the triple expresses a relation that is true with respect to the cited evidence and the document context.
- **Chunk-citation correctness**—at least one cited chunk contains direct textual evidence for the triple.

The instructions state that the two criteria are decided independently: a triple may be factually correct overall while its citation fails to provide direct support, and vice versa.

Procedure and quality control. The annotators worked separately, applying the same criteria and relying on the cited evidence rather than on external knowledge. Their aggregated labels were compared with the DeepSeek-V4-Pro decisions using F1 and Cohen's κ.

Agreement with the automatic judge. Agreement between the aggregated human labels and DeepSeek-V4-Pro is reported in Table 2. Human–LLM F1 is 0.957 for factual correctness and 0.887 for chunk-citation correctness, with Cohen's κ of 0.881 and 0.751 respectively. This strong agreement, particularly on factual correctness, supports the judge as a scalable proxy, though it does not rule out judge bias.

Risks and consent. The annotation task involved only publicly available text (Wikipedia excerpts and public-domain literary passages), collected no personal or sensitive information, and posed no foreseeable risk to annotators.