# CORPUS-GUIDED DUAL-PATH PROPAGATION FOR GRAPH RETRIEVAL-AUGMENTED GENERATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Graph-based retrieval-augmented generation supports multi-hop retrieval by organizing corpus information into graphs. However, existing relation-free graph retrieval methods rely primarily on query–sentence similarity to search for evidence. This can exclude useful bridging evidence with low query similarity and activate incidental entities unrelated to the reasoning chain. In this paper, we propose a simple and effective approach called NexusRAG, which augments the relation-free Tri-Graph with a corpus-level entity neighborhood structure derived from joint entity co-occurrence and semantic similarity. NexusRAG employs this structure to guide two complementary propagation paths: neighborhood-constrained semantic propagation through sentences identifies the query-relevant entity frontier, while direct structural propagation between neighboring entities expands that frontier to structurally related entities. The propagated entity weights also inform neighborhood-aware passage initialization for Personalized PageRank. Experiments on three multi-hop QA benchmarks and a domain-specific subset of GraphRAG-Bench show that NexusRAG consistently outperforms existing approaches. On the GraphRAG-Bench subset, NexusRAG achieves the highest evidence recall in all question categories, exceeding baselines by 4.2–8.1 points.

## 1 INTRODUCTION

Retrieval-augmented generation (RAG) enables large language models (LLMs) to ground their responses in external evidence (Lewis et al., 2020; Gao et al., 2023). Conventional RAG systems typically retrieve passages based on their semantic similarity to the query. Although effective in finding directly relevant information, this approach can struggle with questions that require connecting evidence across multiple documents. In such cases, necessary intermediate evidence may have a low similarity to the original query, with its relevance becoming apparent only through connections to other passages. Independent passage retrieval can therefore overlook useful bridging evidence, leaving the generator without sufficient information to answer the question (Borgeaud et al., 2022; Izacard et al., 2023; Han et al., 2025; Zhang et al., 2025).

Graph-based retrieval-augmented generation (GraphRAG) addresses this limitation by explicitly modeling connections within a corpus to guide evidence retrieval (Edge et al., 2024; Procko & Ochoa, 2024; Zhang et al., 2025). Related approaches explore different forms of structural organization. RAPTOR (Sarthi et al., 2024) recursively clusters and summarizes text to build a hierarchical index, while Microsoft GraphRAG (Edge et al., 2024) constructs an entity graph and generates summaries of its communities. Other methods exploit explicit entity–relation graphs to support retrieval and reasoning (Gutiérrez et al., 2024; Gutiérrez et al., 2025; Guo et al., 2025; He et al., 2024; Luo et al., 2025). These structures help connect evidence across documents, allowing retrieval to account for relationships that independent passage scoring may overlook.

However, graph construction introduces its own sources of error. Inaccurate relation extraction can create misleading links between entities, while fragmented or inconsistent graph structures can hinder access to relevant evidence. Such errors can degrade the retrieved context and compromise downstream reasoning. GraphRAG-Bench (Xiang et al., 2026) evaluates graph construction, retrieval, and generation across four levels of task complexity: fact retrieval, complex reasoning, contextual summarization, and creative generation. In particular, its evidence recall metric measures the proportion of reference claims supported by the retrieved context, independently of the generated



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.

answer. This distinction highlights the importance of recovering complete supporting evidence when questions require integrating information across multiple passages.

These limitations motivate relation-free graph retrieval, which builds a useful corpus structure without explicitly extracting semantic relations between entities. LinearRAG (Zhuang et al., 2026), for example, constructs an entity-sentence-passage Tri-Graph using lightweight entity extraction and semantic linking. Starting from entities matched to the query, it iteratively propagates activation through the entity-sentence subgraph, guided by query-sentence similarity. The resulting entity scores, together with query-passage relevance, initialize Personalized PageRank (PPR) (Haveliwala, 2002) on the entity-passage subgraph to rank passages. Corpus sentences thus serve as contextual bridges between entities, supporting multi-hop retrieval without explicit relation extraction.

However, LinearRAG's sentence-mediated propagation ties entity activation to query-sentence similarity, which can lead to two failure modes. *Query-gated cutoff* occurs when a sentence connecting relevant entities has low query similarity, weakening the activation signal and potentially preventing propagation from reaching necessary evidence. *Spurious activation* occurs when a query-relevant sentence mentions incidental entities, allowing activation to spread to entities that do not support the answer. Together, these cases highlight a limitation of using sentence relevance to guide entity transitions: a useful bridge may have low query similarity, while a relevant sentence may contain irrelevant entities. This motivates complementing query-dependent sentence mediation with an explicit corpus-level entity neighborhood to guide propagation.

We propose NexusRAG, a corpus-guided graph retrieval framework that augments the relation-free Tri-Graph with a corpus-level neighbor prior derived from entity co-occurrence and semantic similarity. This prior captures entity relatedness without explicit relation extraction and guides a dual-path propagation process. The semantic propagation uses query-sentence similarity to identify a relevant entity frontier, with entity transitions constrained by the neighbor prior. The structural propagation then expands this frontier through structurally related entities. The resulting entity weights inform passage initialization for Personalized PageRank, incorporating neighborhood information into passage ranking.

Our main contributions are as follows:

- We identify two failure modes of sentence-mediated entity propagation, *query-gated cutoff* and *spurious activation*, highlighting the limitations of query–sentence relevance as a guide for entity transitions.
- We propose NexusRAG, a simple and effective approach which uses a corpus-level neighbor prior to guide dual-path propagation, recovering bridging evidence while suppressing spurious entity activation without explicit relation extraction.
- Experiments on three multi-hop QA benchmarks and a domain-specific subset of GraphRAG-Bench demonstrate consistent improvements over multiple strong baselines. NexusRAG achieves the highest evidence recall in all four categories of the GraphRAG-Bench subset, exceeding baselines by 4.2–8.1 points.

## 2 Related Work

**Retrieval-Augmented Generation (RAG)** grounds language models on retrieved external evidence (Lewis et al., 2020; Karpukhin et al., 2020; Gao et al., 2023). Dense passage retrieval is effective when relevant information is localized, but questions that require evidence from multiple documents motivate retrieval methods that explicitly organize or reason over corpus structure.

**Graph-based Retrieval-Augmented Generation (GraphRAG)** introduces relational structure to support multi-hop retrieval. We focus on three graph construction strategies that are most relevant to this work. One line of work organizes corpus information into hierarchical structures using clustering, community detection, or recursive summarization. Microsoft GraphRAG (Edge et al., 2024) identifies communities in an entity graph and generates summaries at multiple levels, while RAPTOR (Sarthi et al., 2024) recursively clusters and summarizes text to construct a hierarchical representation. E2GraphRAG (Zhao et al., 2025) also uses hierarchical representations to support multi-hop retrieval. These methods provide access to corpus information at different levels of abstraction, but their performance depends on the intermediate grouping or summary structures.

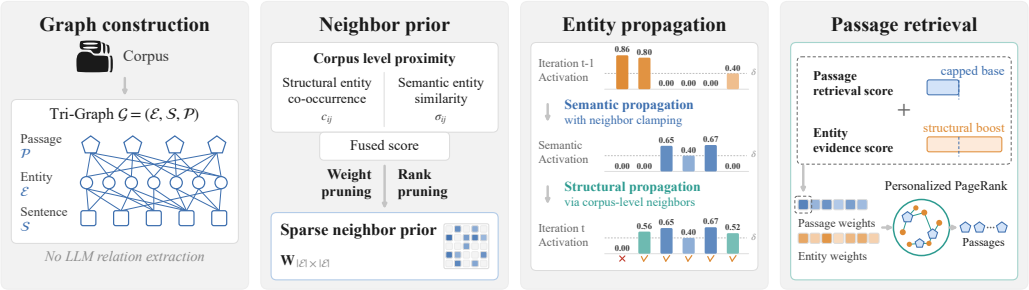



Figure 1: Overview of NexusRAG. **1) Graph Construction.** The corpus is indexed into a relation-free entity–sentence–passage Tri-Graph. **2) Neighbor Prior.** A sparse corpus-level entity neighbor matrix $\mathbf{W}$ is constructed from co-occurrence and semantic similarity with rank and weight pruning. **3) Entity Propagation.** Given a query $q$, neighbor-constrained semantic propagation identifies a query-relevant entity frontier, which structural propagation expands through $\mathbf{W}$ to obtain the activated entity set. **4) Passage Retrieval.** Cumulative entity weights and neighbor-aware passage weights initialize Personalized PageRank for top-$K$ passage retrieval.

Another line of work constructs explicit knowledge graphs by extracting entities and relations from text. HippoRAG (Gutiérrez et al., 2024) and HippoRAG2 (Gutiérrez et al., 2025) build knowledge graphs from extracted relations and apply Personalized PageRank for retrieval. LightRAG (Guo et al., 2025) combines entity–relation structures with hierarchical representations, while G-Retriever (He et al., 2024) and GFM-RAG (Luo et al., 2025) perform graph-based retrieval over structured textual knowledge. These approaches make relational information explicit, but relation extraction introduces additional indexing cost and can produce noisy or inconsistent relations.

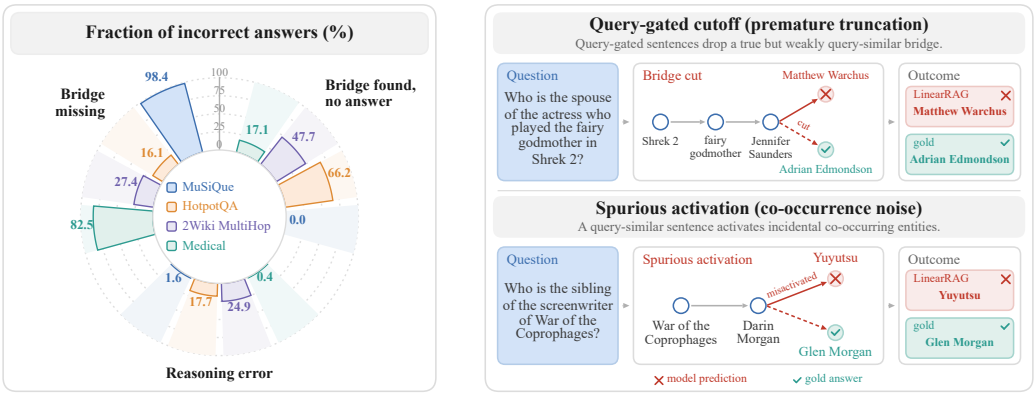
**Relation-free Graph Retrieval** avoids explicit relation extraction and derive graph structure directly from corpus observations. LinearRAG (Zhuang et al., 2026) constructs an entity–sentence–passage Tri-Graph using lightweight entity extraction and semantic linking. Its retrieval process activates entities through query-relevant sentences and aggregates passage importance with Personalized PageRank. NexusRAG retains this relation-free graph construction and introduces corpus-level neighbor weights computed from entity co-occurrence and semantic similarity. The same weights are used during entity propagation and passage initialization, providing a structural signal throughout retrieval.

**Reasoning-enhanced RAG** uses language models to decompose or refine complex queries rather than explicitly organizing the corpus into a graph. LogicRAG (Chen et al., 2026) and LAG (Xiao et al., 2025) decompose questions into subqueries with logical dependencies, while Chain-of-Note (Yu et al., 2024) generates intermediate notes before retrieving supporting documents. Self-RAG (Asai et al., 2024) incorporates self-reflection into the retrieval-generation process. These approaches use query-side reasoning to retrieve evidence for multi-step questions, whereas NexusRAG uses corpus-side structural information for graph retrieval.

## 3 METHOD

**Overview.** NexusRAG follows the relation-free GraphRAG paradigm established by LinearRAG (Zhuang et al., 2026), where entities serve as anchors for connecting evidence distributed across passages. The corpus is indexed as an entity–sentence–passage Tri-Graph $\mathcal{G}$, with entity nodes $\mathcal{E}$, sentence nodes $\mathcal{S}$, and passage nodes $\mathcal{P}$. Edges connect entities to the sentences and passages in which they appear, preserving contextual evidence within the original corpus structure.

Retrieval proceeds in two stages: 1) entity activation, where query-related entities are activated and propagated through the entity–sentence subgraph to identify intermediate evidence, with each active entity selecting its top-$\eta$ query-similar sentences and propagating activation to other entities in those sentences with scores proportional to the source activation and query–sentence similarity; 2) passage retrieval, where the activated entity evidence is transferred to the entity–passage subgraph to initialize Personalized PageRank, which ranks the top-$K$ passages for the LLM.




**(a) Bridge failures dominate retrieval errors** **(b) Query-dependent failure modes**

Figure 2: Retrieval failure analysis of LinearRAG. (a) Bridge entities account for a substantial fraction of retrieval errors. (b) Query-dependent sentence mediation can both suppress necessary bridge evidence and over-activate incidental entities.

This workflow places entity propagation at the center of multi-hop retrieval: the quality of the activated entities directly affects the evidence passed to passage initialization and subsequent PageRank ranking. We therefore revisit how query-dependent propagation determines entity importance and identify two failure modes: *query-gated cutoff* and *spurious activation*.

To address these limitations, we introduce NexusRAG with a query-independent neighbor prior that complements query-dependent sentence relevance with corpus-level entity structure. Figure 1 illustrates the overall framework. The neighbor prior supports a dual-path propagation process: the semantic propagation uses entity-level structural evidence to constrain transitions within query-relevant sentences, while the structural propagation expands activated entities through structurally related neighbors when sentence mediation is insufficient. The resulting entity evidence is then used for passage initialization and Personalized PageRank, providing a more reliable basis for multi-hop passage retrieval.

### 3.1 Evidence Retrieval Failures in LinearRAG

Multi-hop retrieval requires recovering intermediate evidence that connects successive reasoning steps. Results from GraphRAG-Bench (Xiang et al., 2026) reveal a less intuitive pattern: higher evidence recall does not necessarily lead to higher context relevance. With the passage retrieval budget fixed at $K = 5$, one would expect retrieving more gold evidence to improve relevance if query relevance were sufficient to identify useful evidence. The observed mismatch suggests that evidence required for multi-hop reasoning is not always aligned with the query relevance used to guide retrieval.

We therefore examine LinearRAG, where entity propagation is mediated by query–sentence relevance. With GPT-4o-mini as the answer generator, LinearRAG achieves accuracies of $38.2\%$, $69.5\%$, $65.0\%$, and $65.32\%$ on MuSiQue, HotpotQA, 2WikiMultiHopQA, and the Medical subset of GraphRAG-Bench, respectively. Among its incorrect answers, $98.4\%$, $82.3\%$, $75.1\%$, and $99.6\%$ are classified as retrieval misses, respectively (Figure 2(a)). This indicates that incomplete retrieval is a major source of failure and motivates examining how entity activation is propagated before passage ranking.

To locate the source of these retrieval misses, we further inspect the role of bridge evidence in the propagation process. For HotpotQA, 2WikiMultiHopQA, and Medical, we distinguish cases where the bridge entity is not retrieved from those where the bridge entity is retrieved but the answer passage is missing. The two cases account for $16.1\%$ and $66.2\%$ of incorrect answers on HotpotQA, $27.4\%$ and $47.7\%$ on 2WikiMultiHopQA, and $82.5\%$ and $17.1\%$ on Medical, respectively. These results show that retrieval failures are not limited to the final answer passage: in many cases, the intermediate bridge entity itself fails to receive sufficient activation for subsequent expansion.

We then examine LinearRAG's query-dependent sentence mediation (Figure 2(b)) and identify two limitations.

**Query-gated cutoff.** LinearRAG uses query–sentence similarity to control sentence-mediated propagation. However, a sentence that contains necessary bridge evidence may be only indirectly related to the query and therefore receive a low similarity score. Its contribution can then fall below the pruning threshold before the bridge entity is sufficiently activated for further propagation. This indicates that query relevance at the sentence level does not necessarily reflect the importance of the evidence carried by that sentence.

**Spurious activation.** The opposite problem can also occur for highly query-relevant sentences. A single sentence may contain several entities, but these entities can play very different roles in the underlying evidence chain. Because sentence-mediated propagation derives their activation from the same query–sentence relevance, entities that are incidental to the reasoning chain can receive propagation mass together with the relevant entity. This indicates that sentence-level relevance alone cannot distinguish which entities within a relevant sentence should receive stronger propagation.

Together, these observations show that query–sentence relevance alone is insufficient to govern entity propagation: it can both suppress necessary bridge evidence and over-activate incidental entities.

### 3.2 Dual-Path Entity Propagation via Neighbor Prior

To address these limitations, NexusRAG introduces a query-independent neighbor prior that captures corpus-level structural relatedness between entities. We incorporate this prior into two complementary propagation paths: the semantic propagation uses it to constrain entity transitions while retaining query relevance, and the structural propagation uses it to expand query-conditioned entities through structurally related neighbors.

**Query-independent Neighbor Prior.** We construct a query-independent neighbor prior from two complementary signals: sentence-level co-occurrence and semantic similarity between entity embeddings. Co-occurrence captures structural relatedness in the corpus: if two entities frequently appear in the same sentences, they are more likely to participate in the same local evidence or reasoning context. Semantic similarity links related entities in the embedding space, even when they rarely co-occur in the corpus.

The co-occurrence signal is naturally sparse, with nonzero counts only for entity pairs that appear in the same sentence. To obtain a sparse semantic signal, we use approximate nearest-neighbor (ANN) search (Malkov & Yashunin, 2020) to retain the top-$k$ most similar entities for each entity based on cosine similarity. We independently normalize the nonzero values of the two signals and combine them to obtain the neighbor weight between $e_i$ and $e_j$:

$$w_{ij} = \alpha \cdot c_{ij} + (1 - \alpha) \cdot \sigma_{ij}, \tag{1}$$

where $c_{ij}$ and $\sigma_{ij}$ denote the normalized co-occurrence and cosine-similarity scores, respectively, and $\alpha \in [0, 1]$ controls their relative contributions.

For each entity, we retain only neighbors whose fused scores exceed $\tau$ and rank among its top-$\kappa$ candidates. All remaining entries, including self-connections, are set to zero. The resulting sparse neighbor matrix, $\mathbf{W} = [w_{ij}]_{|\mathcal{E}|\times|\mathcal{E}|}$, is constructed offline and reused across queries.

At query time, we extract entities from $q$ using named entity recognition (NER) and match each extracted entity to its most similar entity in $\mathcal{E}$. Matched entities are initialized with the corresponding similarity scores, while all other entities receive zero activation. NexusRAG then alternates between semantic and structural propagation. Semantic propagation uses the neighbor prior to guide updates through query-relevant sentence evidence, while structural propagation expands activations through corpus-level connections. The process terminates when all activation scores become zero or the maximum number of iterations is reached. We denote the final iteration by $T$.

**Semantic propagation.** Sentence-mediated propagation can cause *spurious activation* by treating entities within a query-relevant sentence as equally relevant. NexusRAG uses the corpus-level neighbor prior to limit activation of weakly supported entities. At iteration $t$, we initialize $\tilde{a}_j^{(t)} = 0$ for every entity $e_j \in \mathcal{E}$. For each entity $e_i$ activated in the previous iteration, we select the top-$\eta$ sentences containing it according to their cosine similarity to the query. A selected sentence with similarity $\sigma_m$ induces a candidate propagation score $a_i^{(t-1)}\sigma_m$. We discard candidates below the

threshold $\delta$ and update the other entities in the sentence as follows:

$$\tilde{a}_j^{(t)} = \begin{cases} \max\left(\min\left(a_i^{(t-1)} \cdot \sigma_m,\, w_{ij}\right),\, \delta\right) & e_j \in \mathcal{N}(e_i), \\ \delta & e_j \notin \mathcal{N}(e_i). \end{cases} \tag{2}$$

Here, $w_{ij}$ and $\mathcal{N}(e_i)$ denote the neighbor weight and neighbor set defined by the corpus-level prior. Neighbor weights constrain propagation scores subject to a floor of $\delta$, while non-neighbors receive only this floor value. This attenuates incidental activation while preserving sentence-mediated connections. Updates are applied sequentially: when multiple admissible transitions reach the same entity, the latest assignment overwrites the previous score.

**Structural propagation.** Sentence-mediated propagation may stop when a sentence containing useful evidence has low query similarity, causing *query-gated cutoff*. Structural propagation addresses this failure by expanding semantically activated entities through the corpus-level neighbor graph, allowing activation to reach additional entities.

We initialize $a_j^{(t)} = \tilde{a}_j^{(t)}$ and process neighbor transitions from entities activated during semantic propagation. A structural transition is accepted only if its target has not yet been activated in the current iteration and its score meets the threshold:

$$a_j^{(t)} = \begin{cases} w_{ij} \cdot \tilde{a}_i^{(t)} & \tilde{a}_j^{(t)} = 0 \ \wedge \ w_{ij} \cdot \tilde{a}_i^{(t)} \geq \delta, \\ \tilde{a}_j^{(t)} & \text{otherwise.} \end{cases} \tag{3}$$

Thus, entities activated by semantic propagation retain their scores, while previously inactive entities receive the score of their first admissible structural transition. The resulting activations serve as inputs to the next iteration. We accumulate admitted activation scores across iterations to obtain $A_i$ for each entity $e_i \in \mathcal{E}$.

### 3.3 Neighbor-aware Passage Initialization

A passage can be relevant to the query through either direct semantic similarity or entities reached during propagation. Because query–passage similarity alone can be misleading, we use a bounded dense retrieval score as a baseline and augment it with cumulative entity evidence from neighbor-aware propagation. The initial score of passage $p_j$ is

$$P_{\text{init}}(p_j) = \exp(d_j) + \sum_{e_i \in \mathcal{E}} \frac{A_i \cdot \log\big(1 + \text{count}(e_i, p_j)\big)}{\max\big(\text{hop}(e_i), 1\big)}, \tag{4}$$

where $d_j \in [0, 1]$ is the normalized query–passage cosine similarity, $A_i$ is the cumulative activation weight of entity $e_i$, $\text{count}(e_i, p_j)$ counts its occurrences in passage $p_j$, and $\text{hop}(e_i)$ denotes the propagation hop at which it was last activated.

The exponential transformation preserves the ordering of dense similarities. The entity term favors passages containing strongly activated entities, rewards repeated mentions with diminishing returns through logarithmic weighting, and discounts cumulative activation according to the last activation hop. We use these passage scores together with the cumulative entity weights to initialize Personalized PageRank on the entity–passage subgraph. The top-$K$ passages in the resulting ranking are provided to the LLM for answering questions.

## 4 Experiments

We organize our experiments around three research questions: **Q1 (Effectiveness):** How does NexusRAG compare with the evaluated baselines in evidence recall and QA performance? **Q2 (Component Analysis):** How do the corpus-level neighbor prior, dual-path propagation, and neighbor-aware passage initialization contribute to performance? **Q3 (Sensitivity):** How do the fusion coefficient $\alpha$, pruning threshold $\tau$, and neighbor cap $\kappa$ affect the calculation of neighbor prior? Additional experiments on recall evaluation, efficiency and scalability, detailed sensitivity analyses, and case studies are provided in the appendix.

Table 1: Main results on four benchmarks across three generation backbones. All methods are evaluated with GPT-4o-mini; Qwen3.6-27B-FP8 and DeepSeek-V4-Flash provide additional comparisons between LinearRAG and NexusRAG. **Bold** and underline denote the highest and second-highest results, respectively. Columns report Contain-Acc. (%) (Con.), LLM-Acc. (%) (LLM.), and Avg. (%) (mean of Con. and LLM.). Only LLM-Acc. is used for the Medical dataset.

| Method | HotpotQA | | | 2Wiki | | | MuSiQue | | | Medical |
|---|---|---|---|---|---|---|---|---|---|---|
| | Con. | LLM. | Avg. | Con. | LLM. | Avg. | Con. | LLM. | Avg. | LLM. |
| ***Direct Zero-shot LLM Inference*** | | | | | | | | | | |
| llama-8B | 31.10 | 27.30 | 29.20 | 33.60 | 16.20 | 24.90 | 7.40 | 8.10 | 7.75 | 27.31 |
| llama-13B | 24.20 | 16.80 | 20.50 | 21.90 | 10.50 | 16.20 | 3.30 | 4.40 | 3.85 | 28.86 |
| GPT-3.5-turbo | 33.40 | 43.20 | 38.30 | 28.70 | 31.00 | 29.85 | 10.30 | 21.90 | 16.10 | 45.60 |
| GPT-4o-mini | 38.90 | 40.20 | 39.55 | 36.30 | 31.40 | 33.85 | 13.60 | 15.80 | 14.70 | 42.10 |
| Qwen3.6-27B-FP8 | 37.80 | 38.70 | 38.25 | 50.20 | 34.30 | 42.25 | 15.50 | 15.10 | 15.30 | 40.60 |
| ***Vanilla Retrieval-Augmented Generation*** | | | | | | | | | | |
| Retrieval (Top-1) | 46.30 | 49.10 | 47.70 | 36.60 | 31.70 | 34.15 | 17.80 | 21.10 | 19.45 | 48.01 |
| Retrieval (Top-3) | 53.00 | 56.00 | 54.50 | 44.90 | 39.70 | 42.30 | 25.10 | 27.50 | 26.30 | 59.07 |
| Retrieval (Top-5) | 55.70 | 58.60 | 57.15 | 48.60 | 43.00 | 45.80 | 26.10 | 29.60 | 27.85 | 61.68 |
| ***Graph-based Retrieval-Augmented Generation Methods*** | | | | | | | | | | |
| KGP | 61.50 | 60.90 | 61.20 | 31.60 | 30.00 | 30.80 | 25.60 | 30.10 | 27.85 | 54.22 |
| G-Retriever | 42.20 | 40.60 | 41.40 | 46.60 | 27.10 | 36.85 | 14.40 | 15.50 | 14.95 | 50.36 |
| RAPTOR | 55.90 | 58.30 | 57.10 | 50.10 | 42.10 | 46.10 | 23.30 | 27.40 | 25.35 | 55.75 |
| E2GraphRAG | 61.00 | 63.90 | 62.45 | 54.30 | 38.10 | 46.20 | 23.80 | 26.20 | 25.00 | 58.00 |
| LightRAG | 60.30 | 59.50 | 59.90 | 55.20 | 39.00 | 47.10 | 27.40 | 28.60 | 28.00 | 54.36 |
| HippoRAG | 57.00 | 59.30 | 58.15 | 66.10 | 59.90 | 63.00 | 29.30 | 24.10 | 26.70 | 55.04 |
| GFM-RAG | 62.70 | 65.60 | 64.15 | 66.80 | 59.60 | 63.20 | 29.90 | 34.60 | 32.25 | 56.07 |
| HippoRAG2 | 62.90 | 64.30 | 63.60 | 62.70 | 55.00 | 58.85 | 31.00 | 35.00 | 33.00 | 60.77 |
| ***Linear Graph Retrieval-Augmented Generation Methods*** | | | | | | | | | | |
| LinearRAG | 65.5 | 69.5 | 67.50 | 69.5 | 65.0 | 67.25 | 32.3 | 38.2 | 35.25 | 65.32 |
| LinearRAG + DeepSeek-V4-Flash | 76.0 | 86.7 | 81.35 | 83.3 | 86.8 | 85.05 | 46.7 | 57.5 | 52.10 | 66.00 |
| LinearRAG + Qwen3.6-27B-FP8 | 69.3 | 85.0 | 77.15 | 80.0 | 83.1 | 81.55 | 45.6 | 54.5 | 50.05 | 70.22 |
| NexusRAG (ours) | **70.1** | **72.9** | **71.50** | **72.2** | **68.7** | **70.45** | **35.5** | **41.0** | **38.25** | **73.91** |
| NexusRAG (ours) + DeepSeek-V4-Flash | **79.8** | **88.1** | **83.95** | **85.9** | **87.6** | **86.75** | **50.0** | **60.4** | **55.20** | **78.90** |
| NexusRAG (ours) + Qwen3.6-27B-FP8 | **72.2** | **88.7** | **80.45** | **81.4** | **86.0** | **83.70** | **47.1** | **58.4** | **52.75** | **73.96** |

### 4.1 Experimental Setting

**Datasets.** We evaluate NexusRAG on three multi-hop QA benchmarks (HotpotQA, 2WikiMultiHopQA, and MuSiQue) and the Medical subset of GraphRAG-Bench (Xiang et al., 2026). For the three multi-hop benchmarks, we adopt the 1,000-question validation subsets used by LinearRAG and HippoRAG (Zhuang et al., 2026; Gutiérrez et al., 2024). For Medical, we evaluate 2,062 questions following the GraphRAG-Bench evaluation protocol.

**Baselines.** We compare NexusRAG with Vanilla RAG (Top-1, Top-3, and Top-5) and the following graph-based retrieval methods: KGP (Wang et al., 2024), G-Retriever (He et al., 2024), RAPTOR (Sarthi et al., 2024), E$^2$GraphRAG (Zhao et al., 2025), LightRAG (Guo et al., 2025), HippoRAG (Gutiérrez et al., 2024), GFM-RAG (Luo et al., 2025), and HippoRAG2 (Gutiérrez et al., 2025). LinearRAG (Zhuang et al., 2026) serves as our direct baseline. Details of all baselines are provided in Appendix C.

**Metrics.** Following Zhuang et al. (2026), we evaluate QA performance using two metrics. *Contain-Match Accuracy (Contain-Acc.)* measures the proportion of generated responses containing the reference answer. *LLM-as-a-Judge Accuracy (LLM-Acc.)* evaluates answer correctness against the reference, allowing for valid paraphrases and formatting differences. We report both metrics on the three multi-hop QA benchmarks and only LLM-Acc. on Medical, whose reference answers are typically long and descriptive. For retrieval quality, we adopt *Context Relevance* and *Evidence Recall* from GraphRAG-Bench (Xiang et al., 2026). Context Relevance assesses the alignment between the

Table 2: Retrieval quality evaluation results (%) following the GraphRAG-Bench protocol across four question categories. Recall and relevance are reported for fact retrieval, complex reasoning, contextual understanding, and creative generation. The highest and second-highest results are shown in **bold** and underlined, respectively.

| Method | Fact Retrieval | | Complex Reasoning | | Contextual | | Creative Generation | |
|---|---|---|---|---|---|---|---|---|
| | Recall | Relevance | Recall | Relevance | Recall | Relevance | Recall | Relevance |
| Vanilla RAG (Top-5) | 86.24 | 63.71 | 84.97 | **84.11** | 84.14 | **89.94** | 44.88 | 58.73 |
| RAPTOR | 85.40 | 69.38 | 89.70 | 53.20 | 88.86 | 58.73 | 72.70 | 52.71 |
| $E^2$GraphRAG | 87.84 | 69.74 | 87.08 | 62.67 | 89.17 | 71.63 | 60.26 | 35.84 |
| LightRAG | 80.32 | 41.27 | 82.91 | 42.79 | 85.71 | 43.11 | 81.34 | 45.17 |
| GFM-RAG | 90.08 | 57.90 | 85.03 | 33.06 | 78.62 | 40.14 | 83.51 | 22.87 |
| HippoRAG | 87.25 | 52.44 | 83.80 | 42.19 | 83.46 | 49.13 | 81.66 | 45.03 |
| LinearRAG | 88.86 | **86.09** | 87.03 | 81.58 | 89.13 | 87.89 | 89.08 | **72.74** |
| NexusRAG (ours) | **94.30** | 81.17 | **94.62** | 68.52 | **94.36** | 79.41 | **97.16** | 61.66 |

retrieved context and the question, while Evidence Recall measures the proportion of reference claims supported by the retrieved context. Retrieval results are reported at the end of the Q1 evaluation.

**Implementation.** We use all-mpnet-base-v2 (Song et al., 2020) for extracting entity, sentence, and passage embeddings, with spaCy (Honnibal et al., 2020) en_core_web_trf for entity recognition on the general-domain datasets and en_core_sci_scibert for the Medical corpus. We retrieve $K = 5$ passages by default. Table 1 reports results with GPT-4o-mini, DeepSeek-V4-Flash, and Qwen3.6-27B-FP8, with each model serving as both generator and judge in its corresponding setting. For the neighbor prior, we set $\alpha = 0.5$, $\tau = 0.5$, and $\kappa = 5$, with the ANN candidate count set to $k = \kappa$. Retaining at most $\kappa$ neighbors per entity limits index storage and the computation required for direct neighbor expansion. The propagation and PPR parameters follow LinearRAG's reported configuration. Parameter sensitivity is examined in Section 4.4.

### 4.2 Main Results and Evidence Retrieval (Q1)

Table 1 presents the main comparison. Relative to LinearRAG, NexusRAG obtains higher Avg. scores on all four datasets, while the additional backbones show the same direction of change.

**Observation 1.** The main results show that several graph-based retrieval already provides an advantage over conventional RAG, while the relation-free LinearRAG further demonstrates that explicit relation extraction is not necessary to obtain effective graph-based retrieval. Nevertheless, LinearRAG still leaves room for improvement. On GPT-4o-mini, NexusRAG consistently improves the overall Avg. over LinearRAG on all three datasets with both metrics, by +4.0% on HotpotQA, +3.0% on MuSiQue, and +3.2% on 2Wiki. Both Contain-Acc. and LLM-Acc. increase on these datasets; for example, HotpotQA improves by +4.6% and +3.4%, respectively. On Medical, where only LLM-Acc. is reported, NexusRAG further improves it by 8.6%. These results indicate that although relation-free graph retrieval already alleviates limitations of conventional RAG, the entity propagation used by LinearRAG remains an important source of retrieval error.

**Observation 2.** The same trend holds with Qwen3.6-27B-FP8, suggesting that the improvement is not tied to a particular generation backbone. NexusRAG improves Avg. over LinearRAG by +3.30% on HotpotQA, +2.15% on 2Wiki, and +2.70% on MuSiQue, while LLM-Acc. on Medical increases by +3.74%. The consistency across datasets and backbones suggests that NexusRAG improves the retrieval process itself rather than merely exploiting differences in the generator.

To evaluate retrieval quality independently of generation, we follow the GraphRAG-Bench protocol on the Medical dataset, which contains four question categories of increasing difficulty: Fact Retrieval, Complex Reasoning, Contextual, and Creative Generation. The results are reported in Table 2.

**Observation 3.** Table 2 reports two retrieval metrics for the baselines, LinearRAG, and NexusRAG: *recall* (evidence recall, the fraction of all gold evidence retrieved) and *relevance* (context relevancy, the relevance of the retrieved content to the query). NexusRAG obtains the highest evidence recall in all four categories, with differences of 4.2–8.1 points relative to the next-highest result in each category. The context-relevance scores vary across question categories, from 61.66 on Creative

Generation to 81.17 on Fact Retrieval. This variation is consistent with the retrieval objective: recovering bridge evidence can add content that is necessary for the evidence chain but less directly aligned with the query wording.

**Observation 4.** (evaluator comparison). Table 1 shows higher LLM-Acc. when the generator and evaluator are the same model (e.g., 88.7 on HotpotQA for NexusRAG with Qwen3.6-27B-FP8). To separate generation from evaluation, we fix the GPT-4o-mini predictions and re-judge the same outputs with three evaluators: GPT-4o-mini, DeepSeek-V4-Flash, and Qwen3.6-27B-FP8. Table 3 reports the resulting LLM-Acc. scores. NexusRAG scores higher than LinearRAG for each evaluator on each dataset. Absolute scores differ across evaluators, reflecting differences in evaluator calibration.

Table 3: LLM-Acc. (%) of the same GPT-4o-mini predictions re-judged by three evaluators. Highest results are in **bold**.

| Method | Evaluator | Hotpot | 2Wiki | MuSiQue | Med. |
|---|---|---|---|---|---|
| LinearRAG | GPT-4o-mini | 69.5 | 65.0 | 38.2 | 65.3 |
| | DeepSeek-V4-Flash | 78.1 | 73.8 | 42.8 | 60.5 |
| | Qwen3.6-27B-FP8 | 78.4 | 74.9 | 42.9 | 74.3 |
| NexusRAG | GPT-4o-mini | **72.9** | **68.7** | **41.0** | **73.91** |
| | DeepSeek-V4-Flash | **80.2** | **74.9** | **44.6** | **63.06** |
| | Qwen3.6-27B-FP8 | **81.9** | **76.7** | **46.6** | **79.29** |

### 4.3 Ablation Study (Q2)

The neighbor prior $\mathbf{W}$ is consulted at three points of the pipeline, and we ablate each in turn. **(1)** w/o Neighbor removes $\mathbf{W}$ altogether. **(2)** w/o Structural propagation drops its *expansion* use. **(3)** w/o Neighbor Clamp drops its *gating* use, so non-neighbor transitions are no longer held at the $\delta$ floor. **(4)** w/o Neighbor-aware Init drops its readout, initializing passages without the neighbor-aware entity weights of Eq. 4. Results are in Table 4.

**Observation 5.** Each use of $\mathbf{W}$ contributes to the final score. Removing Neighbor Clamp, Structural propagation, or Neighbor-aware Init changes average LLM-Acc. by $-1.9$, $-2.1$, and $-1.4$ points, respectively, while removing $\mathbf{W}$ altogether changes it by $-3.3$ points. The three uses therefore provide complementary contributions, with the full prior giving the largest change among the ablations.

Table 4: Ablation across the four benchmarks (Qwen3.6-27B-FP8).

| | HotpotQA | | 2Wiki | | MuSiQue | | Med. |
|---|---|---|---|---|---|---|---|
| Variant | Con. | LLM. | Con. | LLM. | Con. | LLM. | LLM. |
| NexusRAG | **72.2** | **88.7** | **81.4** | **86.0** | **47.1** | **58.4** | **73.96** |
| w/o Neighbor Clamp | 71.2 | 86.6 | 80.0 | 83.4 | 46.2 | 56.9 | 72.58 |
| w/o Structural propagation | 71.4 | 87.7 | 80.8 | 84.8 | 45.0 | 55.0 | 71.10 |
| w/o Neighbor | 69.8 | 85.6 | 79.8 | 82.8 | 45.7 | 54.6 | 70.81 |
| w/o Neighbor-aware Init | 71.3 | 87.6 | 80.9 | 84.9 | 46.1 | 56.3 | 72.50 |

### 4.4 Parameter Sensitivity (Q3)

We vary $\alpha$, $\tau$, and $\kappa$ one at a time on HotpotQA with the Qwen3.6-27B-FP8 backbone (Figure 3); full details are in Appendix G.

**Observation 6.** Across the swept $\alpha$ range, LLM-Acc. varies by at most 1.2 points; within $[0.1, 0.9]$ it remains in the range 88.1–88.9, while $\alpha = 0$ and $\alpha = 1$ give 87.7 and 88.0, respectively. Across the swept $\tau$ range, LLM-Acc. changes from 89.0 at $\tau = 0.4$ to 87.9 at $\tau = 0.9$. The density drops by about $66\times$ between $\tau = 0.5$ and $\tau = 0.6$, while LLM-Acc. changes by 0.2 points. $\kappa$ varies within 88.1–88.9 in the tested range.

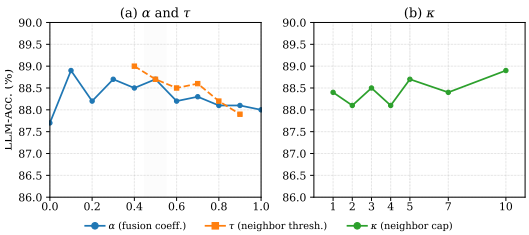



Figure 3: Sensitivity study of the neighbor-mechanism parameters on the HotpotQA dataset.

## 5 Conclusion

We propose NexusRAG, a relation-free graph retrieval framework guided by a corpus-level neighbor prior derived from entity co-occurrence and semantic similarity. The prior guides dual-path entity propagation and passage initialization for Personalized PageRank to mitigate query-gated cutoff and spurious activation. Experiments on three multi-hop QA benchmarks and a domain-specific subset of GraphRAG-Bench show that NexusRAG consistently outperforms the evaluated baselines across retrieval paradigms. On the GraphRAG-Bench subset, NexusRAG achieves the highest evidence recall in all four question categories, exceeding the next-best result in each category by 4.2–8.1 points.

## AI USE STATEMENT

Large language models (LLMs) were used to aid in writing and polishing the manuscript, including sentence rephrasing, grammar checking, and improving readability and flow. The LLM was not involved in ideation, research methodology, or experimental design; all concepts, analyses, and results are developed and conducted by the authors. We take full responsibility for the entire manuscript, including any text refined with LLM assistance, and have ensured it adheres to ethical guidelines without plagiarism or scientific misconduct.

## ETHICS STATEMENT

The four datasets used in the experiments: HotpotQA, 2WikiMultiHopQA, MuSiQue and Medical, are widely used public benchmarks. Our research strictly adheres to the ICLR Code of Ethics, particularly regarding data privacy, transparency, and responsible computing practices. No human participants were involved in this study.

## REPRODUCIBILITY STATEMENT

We provide the full experimental configuration in Table 6 (Appendix), and the complete neighbor construction, dual-path propagation, and passage initialization procedures are specified in Eq. 1–4. The implementation follows the experimental settings described in Section 4. Code will be released upon publication.

## 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)*, pp. 9112–9141, 2024.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning (ICML)*, 2022.

Shengyuan Chen, Chuang Zhou, Zheng Yuan, Qinggang Zhang, Zeyang Cui, Hao Chen, Yilin Xiao, Jiannong Cao, and Xiao Huang. You don't need pre-built graphs for RAG: Retrieval augmented generation with adaptive reasoning structures. In *Conference on Artificial Intelligence (AAAI)*, 2026.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The Faiss library. *arXiv preprint arXiv:2401.08281*, 2024.

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

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

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

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 *International Conference on Machine Learning (ICML)*, 2025.

Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, et al. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309*, 2025.

Taher H. Haveliwala. Topic-sensitive PageRank. In D. Lassner, D. D. Roure, and A. Iyengar (eds.), *Proceedings of the Eleventh International World Wide Web Conference (WWW 2002)*, pp. 517–526, Honolulu, Hawaii, USA, 2002. ACM. doi: 10.1145/511446.511513. URL https://dl.acm.org/doi/10.1145/511446.511513.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength natural language processing in Python, 2020.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *The Journal of Machine Learning Research (JMLR)*, 2023.

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 *emnlp*, pp. 6769–6781, 2020.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. Gfm-rag: Graph foundation model for retrieval augmented generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yu A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020. doi: 10.1109/TPAMI.2018.2889473.

Tyler Thomas Procko and Omar Ochoa. Graph retrieval-augmented generation for large language models: A survey. In *2024 Conference on AI, Science, Engineering, and Technology (AIxSET)*, pp. 166–169. IEEE, September 2024. doi: 10.1109/AIxSET62544.2024.00030.

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

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33: 16857–16867, 2020.

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

Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. Knowledge graph prompting for multi-document question answering. In *Conference on Artificial Intelligence (AAAI)*, 2024.

Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. When to use graphs in RAG: A comprehensive analysis for graph retrieval-augmented generation. In *International Conference on Learning Representations (ICLR)*, 2026.

Yilin Xiao, Chuang Zhou, Qinggang Zhang, Su Dong, Shengyuan Chen, and Xiao Huang. LAG: Logic-augmented generation from a cartesian perspective. *arXiv preprint arXiv:2508.05509*, 2025.

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 *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. Chain-of-note: Enhancing robustness in retrieval-augmented language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 14672–14685, November 2024.

Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Junnan Dong, Hao Chen, Yi Chang, and Xiao Huang. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*, 2025.

Yibo Zhao, Jiapeng Zhu, Ye Guo, Kangkang He, and Xiang Li. E2graphrag: 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 (ICLR)*, 2026.

# A Implementation Details and Design Discussion

The complete retrieval procedure of NexusRAG is given in Algorithm 1. The propagation state maintains a per-entity activation score $a_i^{(t)}$ for the current iteration. During each iteration, $\tilde{a}_i^{(t)}$ records the semantic activation before structural expansion, while $a_i^{(t)}$ denotes the resulting activation after both paths. In contrast, the separate cumulative activation weight $A_i$ sums all admitted activation scores received by entity $e_i$ over the retrieval process. Thus, $a_i^{(t)}$ determines whether an entity can continue propagating, whereas $A_i$ determines how strongly the entity contributes to final passage retrieval.

**Algorithm 1** NexusRAG Retrieval

**Require:** Corpus Tri-Graph $\mathcal{G}$, query $q$, thresholds $\tau, \delta$, max hops $maxT$, fusion coefficient $\alpha$, neighbor cap $\kappa$, top-$\eta$ sentences

Construct co-occurrence and semantic similarity score
Normalize nonzero co-occurrence and semantic similarity scores
Compute $w_{ij}$ using the normalized score using Eq. 1
Construct $\mathbf{W}$ by retaining neighbors with $w_{ij} \geq \tau$ that rank within the top-$\kappa$ candidates for each entity
Apply NER and match each query entity to its most similar entity in $\mathcal{E}$ to obtain seed set $E_0$
Initialize $a_i^{(0)}$ with the matching similarity for $e_i \in E_0$, and set all other entries to 0
Initialize $A_i \leftarrow a_i^{(0)}$ for each $e_i \in \mathcal{E}$
Initialize $\mathrm{hop}(e_i) \leftarrow 1$ for each $e_i \in E_0$
**Dual-Path Entity Propagation**
**for** $t = 1, \ldots, maxT$ **do**
    Initialize $\tilde{a}_j^{(t)} \leftarrow 0$ and $a_j^{(t)} \leftarrow 0$ for all $e_j \in \mathcal{E}$
    **for** each active $e_i \in \mathcal{E}$ with $a_i^{(t-1)} \geq \delta$ **do** ▷ Semantic propagation
        Select top-$\eta$ query-similar sentences containing $e_i$
        Compute admitted semantic activations using Eq. 2
    **end for**
    **for** each active $e_i \in \mathcal{E}$ and $e_j \in \mathcal{N}(e_i)$ **do** ▷ Structural propagation
        Compute the final activation $a_j^{(t)}$ using Eq. 3
    **end for**
    Accumulate admitted activations into $A_j$ for each activated $e_j$
    Update $\mathrm{hop}(e_j) \leftarrow t$ for each activated $e_j$
    **if** $a_j^{(t)} = 0$ for all $e_j \in \mathcal{E}$ **then**
        **break**
    **end if**
**end for**
Initialize $P_{\mathrm{init}}(p_j)$ for each passage $p_j$ using similarity and accumulated activation score $A_i$ of entities ▷ Eq. 4
Run PPR on the entity–passage subgraph using $P_{\mathrm{init}}(p_j)$ and the accumulated activation score
**return** Top-$K$ passages

## A.1 Complexity

Constructing sparse co-occurrence scores costs $O\big(\sum_{s\in\mathcal{S}} |E_s|^2\big)$, linear in $|\mathcal{S}|$ since each sentence $s$ holds $|E_s| \approx 4$ entities, where $E_s$ is the set of entities in $s$; the similarity term is built by ANN search instead of an all-pairs computation: building the Faiss (HNSW) index (Douze et al., 2024; Malkov & Yashunin, 2020) costs $O(|\mathcal{E}| \log |\mathcal{E}| \cdot d_{\mathrm{emb}})$ and querying it for the top-$k$ neighbors of each entity costs $O(|\mathcal{E}| \log |\mathcal{E}| \cdot d_{\mathrm{emb}})$ in total (each ANN query is $O(\log |\mathcal{E}| \cdot d_{\mathrm{emb}})$), materializing only $O(|\mathcal{E}| \cdot k)$ similarity pairs, where $d_{\mathrm{emb}}$ is the embedding dimension. The resulting neighbor matrix $\mathbf{W}$ is stored in CSR format with storage $O(|\mathcal{E}| \cdot \kappa)$ after the top-$\kappa$ truncation. Per-query retrieval remains linear in the number of entities, matching LinearRAG's inference-time scaling, while the offline construction is reduced from quadratic to near-linear in $|\mathcal{E}|$.

### A.2 Why Neighbors and Why Two Paths?

The neighbor prior addresses the two failure modes and also affects answer localization.

- **Query-gated cutoff.** When the query is lexically distant from bridging sentences, the semantic propagation alone would assign minimal scores (capped at $\delta$), so genuine but weakly query-similar bridges are passed over. The structural propagation provides a fallback path through structural neighbors, ensuring that high-co-occurrence entities are not discarded due to surface-form mismatch.
- **Spurious activation.** Without a structural prior, a query-similar sentence can activate multiple co-occurring entities, including entities that are not on the relevant reasoning path. The neighbor clamp in Eq. 2 bounds the transition by $w_{ij}$; non-neighbor transitions that pass the $\delta$ threshold receive the lower bound $\delta$, after which they cannot participate in the next propagation.
- **Answer localization.** For center-entity queries where the question revolves around a single entity, the answer is frequently found among that entity's contextual partners. The neighbor clamp in Eq. 2 assigns non-neighbor transitions that pass the admission gate the minimum score $\delta$, so multi-source corroboration can still surface a genuinely co-occurring partner.

**Why operate on semantic activations?** Propagating from entities with nonzero $\tilde{a}^{(t)}$ focuses structural expansion on entities newly activated by semantic propagation, allowing structural propagation to recover related entities that the semantic propagation may miss. This preserves query relevance from the semantic propagation while using the neighbor structure as a complementary expansion signal.

**Why two paths?** The semantic and structural propagations address two aspects of retrieval: the semantic propagation retains query-dependent sentence mediation while the neighbor weights limit transitions through structurally weak pairs (Section 3.2); the structural propagation adds transitions along pre-computed neighbor edges without using sentence–query similarity. The structural propagation is seeded by semantic activation (Eq. 3), so it extends the frontier reached by sentence mediation rather than introducing an independent source of query entities. Together, the two paths suppress query-gated cutoff and spurious activation.

### A.3 Formal Properties of the Clamp and the structural propagation

**Property 1 (Terminal non-neighbor activation).** For any admitted transition under Eq. 2, a neighbor receives a score bounded by its structural weight $w_{ij}$, whereas a non-neighbor receives exactly the floor score $\delta$. Consequently, an admitted non-neighbor can contribute to the current evidence set and to the cumulative PPR weight, but it cannot obtain a propagation score above the floor and therefore cannot serve as an unconstrained source of further activation, except when $\sigma_m = 1$.

Under Eq. 2, a neighbor receives

$$\max\Big(\min(a_i^{(t-1)}\sigma_m, w_{ij}), \delta\Big)\,,$$

whereas a non-neighbor receives exactly $\delta$. Thus, non-neighbor transitions are restricted to the floor score and cannot obtain a higher propagation score through this branch, although their $\delta$ contributions remain available to the cumulative PPR weight $A_j$ and hence to passage initialization.

**Property 2 (Query-independent reachability).** For every entity with at least one co-occurring partner, normalization retains a neighbor with $w_{ij} \geq \alpha$, and that neighbor is activated by Eq. 3 whenever the frontier entity's activation satisfies $\tilde{a}_i^{(t)} \geq \delta/\alpha$, regardless of any query–sentence similarity.

The row maximum of the normalized co-occurrence term is 1, so the fused weight of the maximum-weight co-occurrence partner is at least $\alpha$; this gives the $0\%$ zero-neighbor fraction whenever $\tau \leq \alpha$ (Appendix F). The firing condition $w_{ij}\tilde{a}_i^{(t)} \geq \delta$ of Eq. 3 does not use query–sentence similarity, so the maximum-weight structural neighbor can remain reachable even when its bridging sentence has low query similarity.

**Remark (duality).** Query-gated cutoff and spurious activation are the false-negative and false-positive modes of the same admission decision. The two propagation paths address these failure modes in

different ways: semantic propagation constrains sentence-mediated transitions using the corpus-level neighbor prior to reduce spurious activation, while structural propagation expands through structurally supported neighbors without relying on query-sentence similarity for the transition, helping recover potentially missed evidence.

## B DATASETS

Our experimental evaluation is conducted on four datasets: three established multi-hop QA benchmarks, HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (2Wiki) (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022), plus one domain-specific Medical set from GraphRAG-Bench (Xiang et al., 2026).

**(i) HotpotQA (Yang et al., 2018):** A benchmark comprising $97k$ question-answer instances designed to evaluate multi-hop reasoning capabilities. Each question requires models to synthesize information from multiple documents, with up to 2 gold-standard supporting passages provided alongside numerous irrelevant documents.

**(ii) 2WikiMultiHopQA (2Wiki) (Ho et al., 2020):** A multi-hop reasoning benchmark containing $192k$ questions that necessitate information integration across multiple Wikipedia articles. Each instance requires evidence synthesis from either 2 or 4 specific articles, testing models' ability to perform structured cross-document reasoning and maintain coherent information flow.

**(iii) MuSiQue (Trivedi et al., 2022):** A multi-hop QA benchmark featuring $25k$ question-answer pairs that demand 2-4 sequential reasoning steps. Each question requires coherent multi-step logical inference across multiple documents.

**(iv) Medical:** A specialized subset derived from GraphRAG-Bench (Xiang et al., 2026), constructed from structured clinical data sourced from the National Comprehensive Cancer Network (NCCN) guidelines. These guidelines provide standardized treatment protocols, drug interaction hierarchies, and diagnostic criteria. The dataset encompasses four tasks of increasing complexity: fact retrieval, complex reasoning, contextual summarization, and creative generation. GraphRAG-Bench spans 4,076 questions across these difficulty levels in total; the Medical subset used in our evaluation comprises 2,062 questions.

## C BASELINE DETAILS

We compare NexusRAG against the same set of retrieval and GraphRAG baselines as LinearRAG (Zhuang et al., 2026), plus LinearRAG itself as the direct baseline.

**(i) Vanilla RAG** retrieves the top-$K$ passages by dense similarity and feeds them directly to the generator, providing a retrieval-augmented lower bound without any graph structure.

**(ii) KGP** (Wang et al., 2024) builds a knowledge graph over multiple passages, with edges encoding their semantic and lexical similarity; at retrieval, it introduces an LLM-driven graph-traversal agent to navigate the graph and progressively collect supporting passages.

**(iii) G-Retriever** (He et al., 2024) combines graph neural networks with LLMs by formulating subgraph retrieval as a Prize-Collecting Steiner Tree optimization problem, for conversational question answering on textual graphs while mitigating hallucination and enhancing scalability.

**(iv) RAPTOR** (Sarthi et al., 2024) builds a hierarchical tree by applying clustering algorithms and abstractive summarization, facilitating representation at multiple semantic granularities.

**(v) E2GraphRAG** (Zhao et al., 2025) uses spaCy to extract entities and LLMs to summarize passage groups into a hierarchical tree with encoded nodes at indexing; at retrieval, it hits relevant entities in their multi-hop neighborhood and collects associated passages for ranking, otherwise performing dense retrieval over the whole tree.

**(vi) LightRAG** (Guo et al., 2025) employs a two-tier framework that incorporates graph-based representations within textual indexing, merging fine-grained entity–relation mappings with coarse-grained thematic structures.

**(vii) HippoRAG** (Gutiérrez et al., 2024) is a training-free graph-enhanced retriever that uses the Personalized PageRank algorithm with query concepts as seeds for single-step or multi-hop retrieval across disparate documents.

**(viii) GFM-RAG** (Luo et al., 2025) implements a GraphRAG paradigm by constructing graphs from documents and using a graph-enhanced retriever to retrieve relevant documents.

**(ix) HippoRAG2** (Gutiérrez et al., 2025) extends HippoRAG with enhanced paragraph integration and contextualization, optimizing seed-node selection and PageRank reset probabilities while maintaining factual-memory capabilities.

**(x) LinearRAG** (Zhuang et al., 2026) is the direct baseline used for comparison; it performs linear-time retrieval over an entity–sentence–passage tri-graph via Personalized PageRank, without the entity-neighbor modeling introduced in this work.

## D Machine Configuration

Retrieval and indexing experiments were conducted on the hardware in Table 5. Generation and evaluation were performed using three setups: GPT-4o-mini and DeepSeek-V4-Flash via their respective APIs, with Qwen3.6-27B-FP8 running on the same hardware below.

Table 5: Detailed machine configuration used in our experiments.

| Component | Specification |
|---|---|
| GPU | NVIDIA RTX A6000 |
| CPU | Intel(R) Xeon(R) Platinum 8260 CPU @ 2.30GHz |
| CUDA | 12.8 (Driver 570.172.08) |

## E Hyperparameter Configurations

Table 6 reports the full hyperparameter configuration. The propagation and PPR parameters $maxT, \delta, \eta, d, \omega_p$ are inherited from LinearRAG (Zhuang et al., 2026) and held fixed across all datasets. The neighbor-mechanism parameters are set as: fusion coefficient $\alpha = 0.5$, neighbor threshold $\tau = 0.5$, and maximum neighbors per entity $\kappa = 5$. The neighbor cap $\kappa = 5$ bounds the per-entity degree to limit memory and propagation compute; its role as a recall-versus-cost tradeoff is characterized in Section 4.4.

Table 6: Hyperparameter configuration. $maxT$: max propagation hops; $\delta$: pruning threshold; $\eta$: sentences per entity per iteration; $d$: PPR damping; $\omega_p$: passage node weight. The first group is fixed to LinearRAG's reported values; $\alpha, \tau, \kappa$ are the neighbor-mechanism parameters (the structural propagation uses the raw neighbor weight $w_{ij}$ with no extra decay coefficient).

| **Dataset** | $maxT$ | $\delta$ | $\eta$ | $d$ | $\omega_p$ | $\alpha$ | $\tau$ | $\kappa$ |
|---|---|---|---|---|---|---|---|---|
| HotpotQA | 3 | 0.4 | 1 | 0.5 | 0.05 | 0.5 | 0.5 | 5 |
| 2Wiki | 3 | 0.4 | 1 | 0.5 | 0.05 | 0.5 | 0.5 | 5 |
| MuSiQue | 5 | 0.1 | 4 | 0.5 | 0.05 | 0.5 | 0.5 | 5 |
| Medical | 3 | 0.5 | 3 | 0.5 | 0.05 | 0.5 | 0.5 | 5 |

## F Neighbor Matrix Statistics

**Remark (inclusive neighbor construction).** Threshold-based admission has a bounded form: high-weight neighbors retain their transition scores up to their neighbor weights, low-weight edges are admitted with attenuation, and non-neighbors fall to the lower bound $\delta$ (Section 3.2). In particular, when $\tau \leq \alpha$ every entity retains at least its maximum-weight co-occurrence partner; normalization guarantees $w_{ij} \geq \alpha \geq \tau$ for the row maximum, so before rank truncation, every entity with at least one co-occurring partner has at least one candidate with fused weight at least $\alpha$. At the adopted $\kappa = 5$ and $\tau = 0.5$, the resulting effective graph empirically has 0

Table 7: Neighbor graph statistics on the Qwen3.6-27B-FP8 backbone, after the top-$\kappa$=5 truncation (the effective graph used by propagation). Avg. #neighbors = average neighbors per entity in the effective graph; Zero-neighbor (%) = fraction of entities with no effective neighbors. The top half reports per-dataset statistics at the adopted threshold $\tau = 0.5$; the bottom half reports per-threshold statistics on HotpotQA.

| **Dataset** | $\tau$ | **Avg. #neighbors** | **Zero-neighbor (%)** |
|---|---|---|---|
| Per dataset at $\tau = 0.5$ (adopted) | HotpotQA | 1.99 | 0.00 |
| | 2Wiki | 1.99 | 0.00 |
| | MuSiQue | 2.01 | 0.00 |
| | Medical | 2.29 | 0.00 |
| Per threshold on HotpotQA | 0.4 | 4.38 | 0.00 |
| | 0.5 | 1.99 | 0.00 |
| | 0.6 | 0.03 | 97.54 |
| | 0.7 | 0.01 | 99.02 |
| | 0.8 | 0.01 | 99.23 |
| | 0.9 | 0.01 | 99.46 |

Table 7 reports neighbor graph statistics for the Qwen3.6-27B-FP8 backbone after the top-$\kappa$=5 truncation, i.e., the effective graph that the propagation rule actually uses; we report per-dataset statistics at the adopted $\tau = 0.5$ and per-threshold statistics on HotpotQA for $\tau \in \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Higher $\tau$ yields sparser graphs and a gentle, near-monotone accuracy decline: at $\tau \geq 0.6$ the zero-neighbor fraction on HotpotQA jumps to 97–99% and LLM-Acc. falls from 88.5 to 87.9, whereas the denser $\tau = 0.4$ graph (4.38 neighbors per entity) is the most accurate (89.0). The $\sim 66\times$ density collapse between $\tau = 0.5$ and $\tau = 0.6$ costs only 0.2 points, so those removed edges are low-weight and rarely firing; the further 0.6 points lost up to $\tau = 0.9$ come from the higher-weight edges that remain. This is expected: the neighbor graph is a prior on *plausible* partners, not all of which matter for a given query, so accuracy need not track edge density; distant low-weight partners can be deleted almost for free, while the few higher-weight partners that remain are more likely to fire, so their removal costs visibly more. This also reconciles the sweep with the ablation in Section 4.3: the contribution concentrates in the few higher-weight edges, which is why removing almost all low-weight edges costs only 0.2 points while removing the mechanism entirely costs several. We adopt $\tau = \alpha = 0.5$ : every entity retains its maximum-weight co-occurrence partner by construction, low-weight edges are admitted and self-attenuate by their weight, and the average neighbor count of 1.99–2.29 stays well below the cap, so $\kappa$=5 rarely truncates.

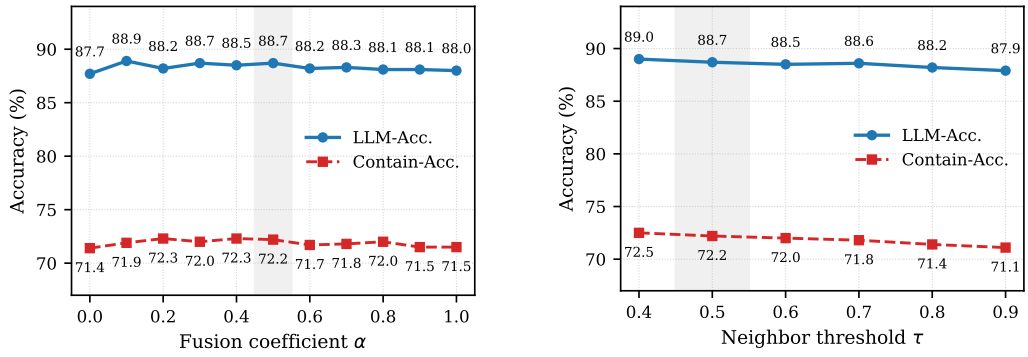
# G Detailed Parameter Sensitivity

## G.1 Impact of Fusion Coefficient $\alpha$ and Neighbor Threshold $\tau$

We first examine the fusion coefficient $\alpha$ and the neighbor threshold $\tau$, sweeping one parameter at a time on HotpotQA with the Qwen3.6-27B-FP8 backbone and the other fixed at its adopted value (Figure 4); $\alpha = \tau = 0.5$ is fixed for all datasets.

**Observation 7.** Across the swept $\alpha$ range $[0.0, 1.0]$ at $\tau = 0.5$, LLM-Acc. varies by at most 1.2 points; within the fusion range $\alpha \in [0.1, 0.9]$ it stays at 88.1–88.9 (Contain-Acc. 71.5–72.3), while the pure boundaries $\alpha = 0$ (87.7/71.4) and $\alpha = 1$ (88.0/71.5, touching the Contain-Acc. lower edge) sit at or just below that band, so each single signal retains most of the benefit yet never exceeds the fused interior. Across the swept $\tau$ range $[0.4, 0.9]$ at $\alpha = 0.5$, LLM-Acc. declines gently from 89.0 at $\tau = 0.4$ to 87.9 at $\tau = 0.9$ (88.7, 88.5, 88.6, 88.2 at $\tau = 0.5$, 0.6, 0.7, 0.8), a total range of 1.1 points, while Contain-Acc. is highest at the dense $\tau = 0.4$ setting (72.5) and otherwise stays within 71.1–72.2.

An edge fires only when $\tilde{a}_i^{(t)} \geq \delta / w_{ij}$, so a low-weight edge contributes little while present: raising $\tau$ from 0.5 to 0.6 removes mostly such edges, collapsing the graph from 1.99 to 0.03 neighbors per entity ($\sim 66\times$) for 0.2 LLM-Acc. points. From $\tau = 0.6$ to 0.9 the graph holds 0.03 to 0.01 neighbors per entity, yet accuracy falls a further 0.6 points, so the edges remaining at the highest thresholds account for most of the contribution. Semantic propagation continues under the $\delta$ lower




(a) Sensitivity to $\alpha$.

(b) Sensitivity to $\tau$.

Figure 4: Sensitivity to the fusion coefficient $\alpha$ and the neighbor threshold $\tau$ on the Qwen3.6-27B-FP8 backbone (HotpotQA; one parameter varied at a time, the other fixed at its adopted value).

bound throughout. We adopt $\tau = \alpha = 0.5$, which retains the maximum-weight co-occurrence partner of every entity by construction. The ablation in Section 4.3 establishes the same point independently: removing the neighbor mechanism degrades accuracy, showing that both signals contribute.

### G.2 Impact of Neighbor Cap $\kappa$

The neighbor cap $\kappa$ bounds the per-entity degree of the effective graph and thereby limits the neighbor-matrix memory footprint and the per-iteration propagation work ($O(|\mathcal{E}| \cdot \kappa)$, with the multi-hop reachable set scaling with $\kappa$ as a branching factor). At the adopted $\tau = 0.5$ the effective graph averages only 1.99–2.29 neighbors per entity (Table 7), well below the cap, so $\kappa = 5$ rarely truncates in practice: it serves as a hard upper bound on memory and propagation cost rather than as an active recall parameter.

We verify this directly by sweeping $\kappa \in \{1, 2, 3, 4, 5, 7, 10\}$ on HotpotQA (Qwen3.6-27B-FP8), holding $\alpha = \tau = 0.5$ fixed (Figure 5).

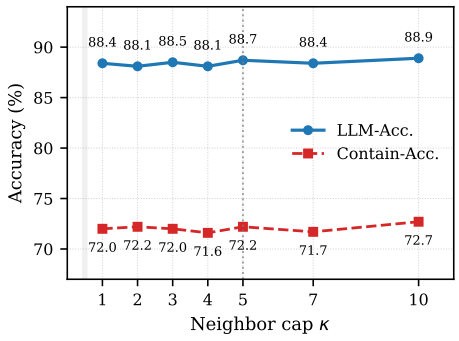



Figure 5: Sensitivity to the neighbor cap $\kappa$.

**Observation 8.** Over $\kappa = 1, 2, 3, 4, 5, 7, 10$, LLM-Acc. takes the values 88.4, 88.1, 88.5, 88.1, 88.7, 88.4, 88.9 and Contain-Acc. 72.0, 72.2, 72.0, 71.6, 72.2, 71.7, 72.7; the adopted $\kappa = 5$ setting (88.7/72.2, from Table 1) falls within this 0.8-point range. The variation is non-monotonic rather than trended: the largest cap $\kappa = 10$ gives the largest value in this sweep yet exceeds the adopted setting by only 0.2 LLM-Acc. points, while $\kappa = 2$ and $\kappa = 4$ give the lowest, 0.6 points below it. This is consistent with the effective graph averaging only 1.99–2.29 neighbors per entity, so a cap as small as $\kappa = 3$ already rarely truncates and $\kappa$ bounds compute rather than recall; $\kappa = 5$ lies inside this flat region and is fixed across all datasets.

## H Retrieval-level Recall Evaluation

We report retrieval-level recall on the two benchmarks with annotated bridge evidence. Table 8 reports Answer-Recall@$K$ and Bridge-Recall@$K$ at two retrieval depths. The structural propagation increases Answer-Recall at both depths on both datasets, with gains up to 2.0 points. Bridge-Recall also increases, with gains of 2.27/3.51 points on HotpotQA and 0.69/1.53 points on 2WikiMultiHopQA at $K$=2/5, respectively, providing a finer-grained view of evidence-chain retrieval.

Finally, the retrieval difference is also reflected in generation: the judge-independent Contain-Acc. rises in the same direction (HotpotQA +4.6pp, 2Wiki +2.7pp, Table 1), and on the Qwen backbones NexusRAG's Contain-Acc. exceeds its own Answer-Recall on 2WikiMultiHopQA (81.4 vs. 76.7; Table 1 and Table 8), showing that the retrieval difference remains visible in the generated answers.

Table 8: Retrieval-level recall@$K$ (%). Bridge-Recall (Br) and Answer-Recall (Ans) at each depth $K$. Highest values are in **bold**.

| Method | HotpotQA | | | | 2Wiki | | | |
|---|---|---|---|---|---|---|---|---|
| | $K$=2 | | $K$=5 | | $K$=2 | | $K$=5 | |
| | Br | Ans | Br | Ans | Br | Ans | Br | Ans |
| LinearRAG | 87.71 | 78.1 | 89.98 | 82.8 | 80.36 | 72.0 | 81.6 | 76.3 |
| NexusRAG | **89.98** | **78.7** | **93.49** | **84.8** | **81.05** | **72.7** | **83.13** | **76.7** |

# I Case Studies: Two Failure Modes of Query-Gated Propagation

We concretize the two failure modes of Section 3.1 with two real multi-hop instances on which NexusRAG answers correctly and LinearRAG does not. In each case the gold reasoning chain is shown as *Support Context* and the two systems' retrieved context is compared side by side (Tables 9 and 10). Case I.1 illustrates *query-gated cutoff* in a representative case: LinearRAG never anchors the bridge entity, so the answer-bearing passage is missing from its context entirely. Case I.2 illustrates *spurious activation*: LinearRAG does complete the first hop, but the bridging sentence names two co-occurring entities and the walk moves to the other co-occurring entity, so the second hop terminates on the wrong branch and the answer passage is again never retrieved. In both cases, NexusRAG recovers a passage that LinearRAG does not return.

Table 9: Case study (HotpotQA). LinearRAG retrieves topically related poet / civil-rights passages, one of which mentions Maya Angelou only in passing, but misses the bridge article *Ford Hall Forum* (which contains the answer sentence) and answers incorrectly, whereas NexusRAG's structural propagation retains the *Ford Hall Forum* passage and answers correctly.

| | |
|---|---|
| **Question** | "What is unique about the forum an American poet, memoirist, and civil rights activist spoke at?" |
| **Ground Truth** | the oldest free public lecture series in the United States |
| **Support Context** | ["American poet, memoirist & civil-rights activist"] → "Maya Angelou" → "spoke at Ford Hall Forum"<br>["Ford Hall Forum"] → "the oldest free public lecture series in the United States" |
| **LinearRAG** | ***Retrieved context:***<br>1) ✗ "Cleanth Brooks (poet & critic)": . . . with *The Southern Review* in 1935; won the Pulitzer Prize for fiction and for poetry . . .<br>2) ✗ "Grosvenor, Duke of Westminster": . . . arts & charity patron; mentions Maya Angelou only in passing . . .<br>3) ✗ "Derick Wade Burleson (poet)": . . . American academic and writer; *Ejo: Poems, Rwanda 1991–94* . . .<br>4) ✗ "Beamus Pierce / Eagle Feather (Smithsonian trustee)": . . . board of trustees for the Smithsonian's National Museum of the American Indian . . .<br>5) ✗ "American Pit Bull Terrier": . . . competes in dog sports and conformation shows . . .<br>*(The bridge article* Ford Hall Forum *is absent from the retrieved context.)*<br>***Prediction:***<br>✗ It is the United Nations. |
| **NexusRAG (ours)** | ***Retrieved context:***<br>1) ✓ "Ford Hall Forum (in a civil-rights biography)": . . . *the Ford Hall Forum is the oldest free public lecture series in the United States* . . . past speakers include Maya Angelou, Noam Chomsky, Martin Luther King Jr. . . .<br>2) ✗ "Cleanth Brooks (poet & critic)": . . . with *The Southern Review* in 1935; won the Pulitzer Prize for fiction and for poetry . . .<br>3) ✗ "Derick Wade Burleson (poet)": . . . American academic and writer; *Ejo: Poems, Rwanda 1991–94* . . .<br>4) ✗ "Langston Hughes gospel show": . . . traditional Christmas carols are sung in gospel style . . . originally written by Langston Hughes . . .<br>5) ✗ "Beamus Pierce / Eagle Feather (Smithsonian trustee)": . . . board of trustees for the Smithsonian's National Museum of the American Indian . . .<br>***Prediction:***<br>✓ It is the oldest free public lecture series in the United States. |

### I.1 Case 1: Query-Gated Cutoff Drops the Bridge

The question asks what is distinctive about a forum addressed by an American poet, memoirist, and civil-rights activist. Its gold reasoning chain is two-hop: the activist resolves to *Maya Angelou*, who spoke at the *Ford Hall Forum*, and that forum is defined as the oldest free public lecture series in the United States. The relevant bridge is therefore the entity *Ford Hall Forum*, whose defining sentence ("the oldest free public lecture series") is lexically distant from the query's "American poet, memoirist, and civil rights activist".

LinearRAG answers incorrectly. Its retriever returns several topically adjacent poet / civil-rights passages, including one that mentions Maya Angelou only in passing (the Grosvenor biography), but never surfaces the *Ford Hall Forum* article that contains the answer sentence. Because the answer-bearing passage is absent from the context, the generator cannot recover it and instead outputs the unrelated "United Nations". This is an instance of *query-gated cutoff* (Section 3.1): the query-gated propagation fails to carry activation across the weakly query-similar bridge to *Ford Hall Forum*.

NexusRAG answers correctly. In addition to the query-gated sentence-mediated propagation, its neighbor mechanism performs a *structural* expansion (Eq. 3) along pre-computed entity–entity edges that are sentence-independent. This expansion reaches the weakly query-similar bridge *Ford Hall Forum* before the query-gated propagation removes it from the final context. The case is consistent with the role of structural propagation in preserving a weakly query-similar bridge during subsequent passage ranking. The generator then reads off "the oldest free public lecture series in the United States." This illustrates the role of the structural propagation in retrieving bridges whose sentences have low query similarity.

### I.2 Case 2: Spurious Activation Diverts the Second Hop

The second instance is a 2WikiMultihopQA question on which LinearRAG *does* complete the first hop and retrieves the bridging passage, yet still fails, even though it names the correct bridge entity in its own output, because the second hop moves to an entity that merely co-occurs with the bridge in the same sentence. The two systems share four of their five retrieved passages; Table 10 compares the two contexts.

The question is two-hop: *Karl Pius* resolves to his mother *Infanta Blanca of Spain*, and her article gives the date of birth. Both hops hinge on a single sentence in passage 349, which states that Karl Pius "was the tenth and youngest child of archduke *Leopold Salvator . . . and infanta Blanca of Spain*"; that is, the sentence names *both* parents, only one of whom is the mother the query asks for.

LinearRAG answers incorrectly, and its own output shows where the chain breaks. It retrieves passage 349, and its generation explicitly identifies "*Archduke Leopold Salvator* (1863–1931), he is the *father*" and searches for "Blanca" with a date, so hop-1 succeeded. What it never retrieves is the mother's own article. Its only passage not shared with NexusRAG is passage 630, the biography of *Leopold Salvator*: the second-hop expansion followed an entity that shares the bridging sentence rather than the queried one. This is *Spurious activation* (Section 3.1): a query-similar sentence activates *every* entity it contains, and with no structural prior to tell which co-occurrence realizes the queried relation, the walk assigns higher activation to the wrong parent. Its downstream consequence is a retrieval gap: the answer passage is missing, so the generator can only report that the date is "not mentioned in the text" (it even notes the father's 15 October 1863 as a tempting but wrong substitute).

NexusRAG answers correctly, and the decisive element is the semantic propagation's neighbor clamp (Eq. 2, Section 3). A transition from an activated entity is admitted only up to the weight of the pre-computed neighbor edge, so an entity that merely shares the bridging sentence is capped rather than freely inherited. *Leopold Salvator* (the father) is not in the neighbor set of *Karl Pius* at the adopted setting (top-$\kappa$=5; Table 7), so the sentence-mediated path propagating through the shared sentence of passage 349 assigns him the lower bound $\delta$ instead of the full transition score;

Table 11: Indexing and retrieval timing on the four benchmarks (seconds).

| | HotpotQA | 2Wiki | MuSiQue | Medical |
|---|---|---|---|---|
| **LinearRAG** | | | | |
| Index(s) | 1057.42 | 541.67 | 1109.22 | 179.86 |
| Precompute(s) | 0.00 | 0.00 | 0.00 | 0.00 |
| Retrieval(s) | 0.252 | 0.214 | 0.240 | 0.216 |
| **NexusRAG (ours)** | | | | |
| Index(s) | 991.31 | 549.20 | 1072.60 | 181.85 |
| Precompute(s) | 24.87 | 12.95 | 23.96 | 1.91 |
| Retrieval(s) | 0.260 | 0.213 | 0.255 | 0.216 |

Table 10: Case study (2WikiMultiHopQA). LinearRAG retrieves the hop-1 bridging passage (349) and correctly identifies the mother, *Infanta Blanca of Spain*, but its second-hop expansion follows the *father Leopold Salvator*, the entity that co-occurs with the mother in that very sentence, and returns his biography (630) instead; the mother's own article (147), which states her date of birth, is never retrieved, and LinearRAG answers "not mentioned in the text". NexusRAG returns the same hop-1 passage *and* retains passage 147, and answers correctly. Passage IDs are the retriever's own indices; ✓ marks passages that carry a gold fact.

| | |
|---|---|
| **Question** | "What is the date of birth of Archduke Karl Pius of Austria, Prince of Tuscany's mother?" |
| **Ground Truth** | 7 September 1868 |
| **Support Context** | ["Archduke Karl Pius of Austria, Prince of Tuscany"] $\xrightarrow{\text{mother}}$ "Infanta Blanca of Spain"<br>["Infanta Blanca of Spain"] $\xrightarrow{\text{date of birth}}$ "7 September 1868" |
| **LinearRAG** | ***Retrieved context:***<br>1) ✓ *hop-1 bridge* (passage 349): "…archduke *karl pius* of austria, prince royal of hungary and bohemia, prince of tuscany (4 december 1909 – 24 december 1953) …he was the tenth and youngest child of archduke *leopold salvator*, prince of tuscany *and infanta blanca of spain*."<br>2) ✗ Habsburg / film biography chunk (passage 348).<br>3) ✗ *wrong branch* (passage 630, retrieved only by LinearRAG): "archduke *leopold salvator*, prince of tuscany …(*15 october 1863* – 4 september 1931), was the son of archduke karl salvator of austria …"; the *father*'s article; it contains a date of birth, but not the mother's.<br>4) ✗ Bolivian-mountains / film-director chunk (passage 350).<br>5) ✗ Habsburg princesses chunk (passage 148): names "*infanta blanca of spain*" again, but only as a parent, with no date.<br>*(The article of* Infanta Blanca of Spain*, which carries "7 september 1868", is absent from the retrieved context.)*<br>***Prediction:***<br>✗ "…the text lacks the info …the father's birth date is *15 October 1863*, but that is factually incorrect for the mother …I will stick with 'Not mentioned in the text'." |
| **NexusRAG (ours)** | ***Retrieved context:***<br>1) ✓ *hop-1 bridge* (passage 349): the same passage as above, establishing *Karl Pius* → mother *Infanta Blanca of Spain*.<br>2)–4) ✗ the same three distractors (passages 348, 350, 148).<br>5) ✓ *hop-2 answer passage* (passage 147, returned only by NexusRAG): "*infanta blanca of spain* (*7 september 1868* – 25 october 1949) was the eldest child of carlos, duke of madrid …in 1889 she married archduke leopold salvator of austria."<br>***Prediction:***<br>✓ 7 September 1868. |

his biography (passage 630) therefore does not enter the top-5 through this transition. Because the mother's article (147), which carries the date of birth, already sits at the edge of the dense retriever's ranking, it is retained once the incidental father is no longer amplified, and the generator reads off "7 September 1868". The retrieval difference is consistent with the lower activation assigned to the incidental father under the structural propagation.

## J Efficiency and Scalability

We also measure the runtime overhead introduced by the additional neighbor structure, since graph-based propagation can be slow at serving time. Table 11 reports the per-benchmark timing to address this: it separates the one-time preprocessing cost from the base graph index cost and from the average online retrieval cost (mean wall-clock time per sample over the test split), to separate the one-time preprocessing cost from the base indexing and online retrieval costs.

The measurements show that the additional cost is concentrated in preprocessing. The one-time `Precompute(s)` for NexusRAG ranges from about 2 s on Medical to roughly 25 s on HotpotQA and MuSiQue, while LinearRAG has no neighbor stage. The base `Index(s)` differs by only a few percent between the two methods, consistent with their shared graph index construction. `Retrieval(s)` differs by a few percent on three benchmarks and by about 6% on MuSiQue (e.g., 0.260 s vs 0.252 s on HotpotQA, 0.216 s vs 0.216 s on Medical). The structural propagation reuses the cached neighbor matrix and adds a vectorized step. The reported results therefore add a one-time preprocessing stage without changing the basic online retrieval procedure.

We now extend this per-benchmark timing study to the large-scale ATLAS-Wiki corpus, using the 5M and 10M token subsets introduced by LinearRAG (Zhuang et al., 2026).

Table 12: Indexing cost on ATLAS-Wiki (5M and 10M tokens). `Total Index (s)` = base graph index (`Index(s)`) + neighbor precompute (`Precompute(s)`). LinearRAG has no neighbor precompute stage, so its total equals `Index(s)`; NexusRAG adds the one-time neighbor construction. HippoRAG figures are measured in this work (Qwen3.6-27B-FP8 via vLLM).

| Dataset | Method | Index(s) | Precompute(s) | Total Index (s) | Token ($\times 10^6$) | |
|---|---|---|---|---|---|---|
| | | | | | Prompt | Completion |
| 5M | HippoRAG | 71019.97 | N/A | 71019.97 | 18.14 | 9.63 |
| | LinearRAG | 4188.49 | N/A | 4188.49 | 0 | 0 |
| | NexusRAG (ours) | 3949.79 | 106.93 | 4056.72 | 0 | 0 |
| 10M | HippoRAG | 100245.49 | N/A | 100245.49 | 37.10 | 20.26 |
| | LinearRAG | 8110.30 | N/A | 8110.30 | 0 | 0 |
| | NexusRAG (ours) | 8077.13 | 228.87 | 8306.00 | 0 | 0 |

Table 12 reports the one-time indexing cost of a complete rebuild from scratch (caches cleared before each run). LinearRAG has no neighbor precompute stage, so its total equals `Index(s)`, while NexusRAG adds the one-time neighbor construction that enables the structural propagation; HippoRAG likewise has no precompute stage, so its total equals its indexing time (measured here with Qwen3.6-27B-FP8 via vLLM). The prompt and completion tokens denote LLM input/output during indexing. NexusRAG and LinearRAG share the same base graph-construction pipeline, so the gap between their `Index(s)` columns ($3949.79$ versus $4188.49$ s at 5M and $8077.13$ versus $8110.30$ s at 10M, at most $6\%$) is run-to-run variation rather than a structural cost of the neighbor mechanism, and the comparison should be read on `Total Index (s)`.

**Observation 9.** On ATLAS-Wiki, the neighbor precompute costs $106.93$ s at 5M tokens and $228.87$ s at 10M, i.e., $2.6\%$ and $2.8\%$ of NexusRAG's total indexing cost, so the one-time overhead of the structural propagation stays below $3\%$ at both scales. Doubling the corpus grows the total cost by $2.05\times$ for NexusRAG and $1.94\times$ for LinearRAG, essentially the same rate, while the precompute itself grows $2.14\times$, so the neighbor stage has the same scaling behavior as the base index over these two corpus sizes. Against an LLM-extraction pipeline the difference is architectural rather than incremental: HippoRAG spends $18.14 \times 10^6$ prompt and $9.63 \times 10^6$ completion tokens at 5M and $37.10 \times 10^6$ and $20.26 \times 10^6$ at 10M, whereas NexusRAG and LinearRAG consume no tokens at indexing time, and HippoRAG's total index time exceeds NexusRAG's by a factor of $17.5$ at 5M and of $12.1$ at 10M.