EVIPORT: FROM PASSAGE RELEVANCE TO EVIDENCE SUFFICIENCY IN BUDGETED MULTI-HOP RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-augmented generation (RAG) grounds Large Language Models (LLMs) in external evidence, but multi-hop questions require a complementary set of passages rather than several individually relevant ones. Under a tight context budget, one-shot dense retrieval can repeatedly retrieve evidence for the same readily retrievable subproblem while missing semantically distant bridge or terminal facts. Existing multi-hop approaches improve evidence discovery through iterative retrieval, query decomposition, or structured corpus reasoning, but generally do not explicitly allocate limited context slots across dependent evidence needs. We introduce EviPort, a framework that represents multi-hop questions as dependency-aware evidence ports without task-specific training or corpus graph construction. Passage-derived answers ground downstream ports, enabling each evidence need to trigger an independent full-corpus search before evidence assembly under a fixed reader budget. Across HotpotQA, 2WikiMultiHopQA, and MuSiQue, EviPort consistently improves both evidence retrieval and downstream question answering over strong dense, iterative, and graph-based baselines. These results support framing multi-hop retrieval as budgeted evidence-set construction rather than independent passage ranking.

1 INTRODUCTION

Retrieval-augmented generation (RAG) equips language models with external evidence by retrieving a small set of passages before generation (Lewis et al., 2020; Karpukhin et al., 2020; Izacard & Grave, 2021). Modern dense, late-interaction, and instruction-tuned retrievers make this pipeline highly effective when a question and its supporting passage are directly aligned (Xiong et al., 2021a; Khattab & Zaharia, 2020; Izacard et al., 2022; Muennighoff et al., 2025; Lee et al., 2025; Zhang et al., 2025). Multi-hop questions, however, often require evidence distributed across passages: one passage identifies an intermediate entity, another describes a property of that entity, and a final passage may support a comparison or conclusion (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022).

The standard one-shot dense retrieval objective is pointwise. Given question q, corpus $\mathcal{D}$, and budget K, it returns the K passages with the largest individual scores $s(q, d)$. Multi-hop answering, by contrast, imposes a set-level requirement: the selected set must jointly cover several distinct and sometimes dependent information needs. Consequently, a Top-K list can contain passages that are all topically relevant yet functionally redundant. A later-hop passage may be difficult to retrieve from q because the entity that identifies it is itself an unknown intermediate answer. Figure 1 illustrates that this failure can arise before final selection: redundant early-hop passages enter the original-query ranking, whereas terminal evidence remains unidentified until intermediate answers are grounded.

Prior work addresses multi-hop retrieval through learned iterative retrieval (Xiong et al., 2021b; Khattab et al., 2021; Qi et al., 2021), reasoning-guided search (Trivedi et al., 2023), question decomposition (Wolfson et al., 2020; Perez et al., 2020; Ye et al., 2025), or structured corpus methods such as RAPTOR (Sarthi et al., 2024), HippoRAG (Jiménez Gutiérrez et al., 2024), HippoRAG 2 (Jiménez Gutiérrez et al., 2025), PropRAG (Wang & Han, 2025), and BrowseNet (Santhana Krishnan et al., 2026).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

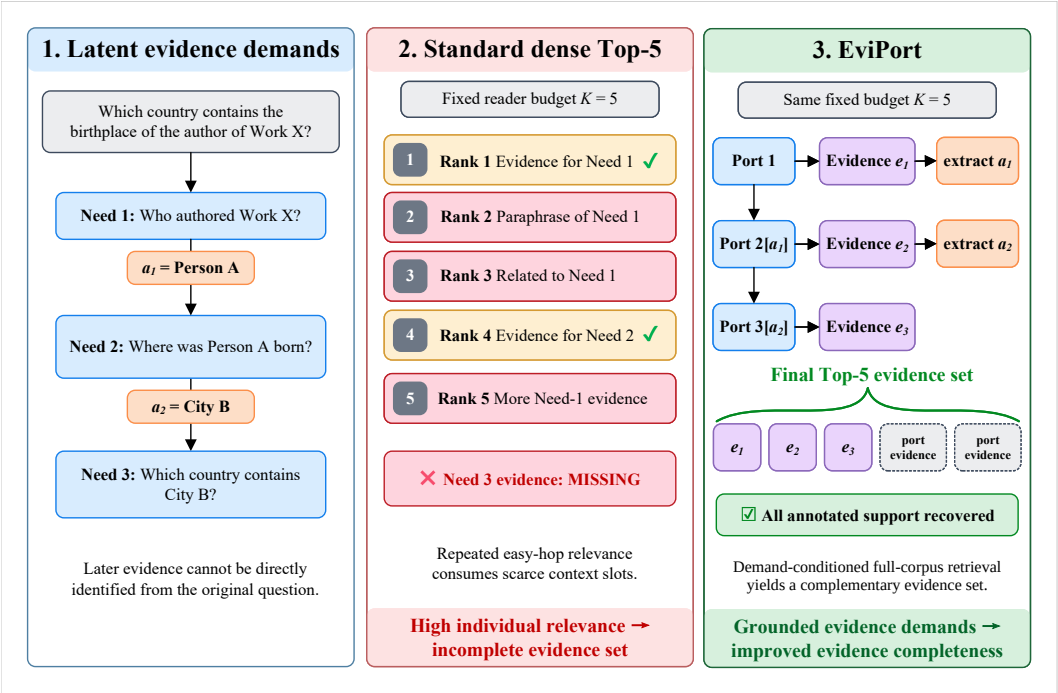


Figure 1: Pointwise passage ranking can spend a fixed reader budget on redundant early-hop evidence, whereas EviPort uses grounded evidence demands to retrieve and assemble complementary support.

These approaches establish that structure and intermediate reasoning are useful, but generally do not explicitly formulate the fixed-budget problem studied here: *which complementary evidence needs should receive the limited positions in the final reader context?* Recent evidence-conditioned query-refinement methods address a related problem by deciding when a query should be resolved or recursively decomposed (Kim et al., 2026). EviPort addresses a complementary axis: it plans an explicit evidence-demand structure once, dynamically instantiates dependent queries with passage-grounded outputs, and allocates the final context across the resulting rankings.

Our contributions are threefold:

- We characterize the mismatch between *pointwise relevance* and *setwise evidence sufficiency*, showing that strong first-hit performance need not imply complete annotated-evidence coverage. We therefore evaluate first-hit retrieval, fractional recall, and complete-evidence behavior separately.
- We propose EviPort, a multi-hop retrieval framework requiring no task-specific training or corpus graph. It represents questions as dependency-aware evidence demands, resolves them through passage-grounded binding and independent full-corpus retrieval, and assembles a port-balanced Top-K context.
- Across three benchmark protocols, EviPort improves retrieval and end-to-end QA over strong dense, iterative, and graph-based baselines. Frozen 2Wiki interventions provide transfer evidence for grounded binding and full-corpus port retrieval beyond the development dataset.

2 Related Work

Dense and iterative multi-hop retrieval. Dense passage retrieval encodes questions and passages into a shared space and independently ranks passages by vector similarity (Karpukhin et al., 2020; Xiong et al., 2021a; Izacard et al., 2022); late interaction preserves token-level matching while retaining efficient retrieval (Khattab & Zaharia, 2020). Multi-hop dense retrievers condition later retrieval on earlier passages or compact an accumulated state (Xiong et al., 2021b; Khattab et al., 2021; Qi et al., 2021), while IRCoT interleaves retrieval with natural-language reasoning so that

newly derived information guides subsequent search (Trivedi et al., 2023). EviPort differs from free-form reasoning-guided retrieval by explicitly representing evidence needs as dependency-aware demands and allocating limited context slots across port-specific rankings.

Question decomposition and adaptive retrieval. Question decomposition converts a compositional query into simpler subquestions and may learn decompositions from explicit annotations or indirect supervision (Wolfson et al., 2020; Perez et al., 2020; Dua et al., 2022). Q-DREAM additionally models subquestion dependencies and optimizes retrieval-space alignment (Ye et al., 2025). More broadly, active and adaptive retrieval methods decide whether, when, or how to retrieve (Jiang et al., 2023; Jeong et al., 2024; Asai et al., 2024), while Hi-Q grows an evidence-conditioned query tree by resolving supported query units and recursively refining unsupported ones (Kim et al., 2026). EviPort instead focuses on the interaction between passage-grounded dependency instantiation and fixed-budget evidence-set construction: dependent queries are not fully specified until upstream evidence is selected, and their retrieved passages compete across needs rather than only within a single relevance ranking.

Structured and graph-based RAG. Hierarchical systems recursively organize passages and abstractions (Sarthi et al., 2024). GraphRAG and LightRAG construct graph-based corpus representations (Edge et al., 2024; Guo et al., 2025), whereas StructRAG selects task-dependent structured representations for knowledge-intensive reasoning (Li et al., 2025). HippoRAG (Jiménez Gutiérrez et al., 2024) and HippoRAG 2 (Jiménez Gutiérrez et al., 2025) integrate extracted structures, passages, and Personalized PageRank for associative retrieval. PropRAG retrieves over context-rich proposition paths (Wang & Han, 2025), and BrowseNet traverses a graph of chunks using a query subgraph (Santhana Krishnan et al., 2026). These methods construct or exploit graph-based or task-dependent structured representations of the corpus. EviPort leaves the corpus unchanged apart from a standard passage-embedding index and creates a transient, query-specific structure online.

Evidence-set construction under a budget. Joint evidence retrievers directly model compatibility among candidate passages rather than selecting each passage in isolation (Yadav et al., 2021). Classical diversified retrieval and summarization similarly penalize redundancy or optimize coverage through maximal marginal relevance, submodular objectives, or determinantal point processes (Carbonell & Goldstein, 1998; Lin & Bilmes, 2011; Kulesza & Taskar, 2012; Gillenwater et al., 2019). EviPort shares their set-level motivation but differs in the source of diversity: it does not diversify a fixed candidate pool by lexical or embedding-space dissimilarity. Each dependency-bound evidence port instead launches a new full-corpus retrieval, allowing semantically distant passages to enter the candidate pool before fixed-budget selection.

3 Problem Formulation

Let q be a question, $\mathcal{D} = \{d_i\}_{i=1}^{N}$ a passage corpus, and K the maximum number of passages that can be provided to the reader. Let the nonempty set $E^\star(q) \subseteq \mathcal{D}$ contain the annotated supporting passages for q. The retrieval task is to construct a set $S_K(q) \subseteq \mathcal{D}$ subject to $|S_K(q)| \leq K$.

Conventional one-shot dense retrieval ranks passages with a learned relevance function $s(q, d)$ and returns

$$R_q^K = \arg\operatorname*{topK}_{d \in \mathcal{D}} s(q, d), \tag{1}$$

Because each passage is scored independently against the question, this formulation does not explicitly account for redundancy among selected passages or dependencies between evidence requirements. High individual relevance scores therefore need not yield complete annotated-evidence coverage under a limited budget.

We characterize evidence coverage at three complementary levels:

$$\text{AnyEvidence@}K(q) = \mathbb{1}[S_K(q) \cap E^\star(q) \neq \emptyset], \tag{2}$$

$$\text{Recall@}K(q) = \frac{|S_K(q) \cap E^\star(q)|}{|E^\star(q)|}, \tag{3}$$

$$\text{FullEvidence@}K(q) = \mathbb{1}[E^\star(q) \subseteq S_K(q)]. \tag{4}$$

Here $\mathbb{1}[\cdot]$ is the indicator function. AnyEvidence@K measures whether at least one annotated supporting passage is retrieved, Recall@K measures the retrieved fraction of the annotated support, and FullEvidence@K tests whether the entire annotated set is present. FullEvidence@K is an annotated-evidence-completeness proxy, not a guarantee that the retrieved context is sufficient or necessary for answering the question.

Our objective is to improve annotated-evidence coverage under a fixed reader budget K. Rather than relying only on pointwise relevance, we construct a retrieval set that addresses distinct and potentially dependent evidence requirements induced by the question.

4 EviPort: Dependency-Aware Evidence Construction

4.1 Framework Overview

EviPort compiles a multi-hop question into dependent retrieval demands and executes them under a fixed reader budget. Figure 2 summarizes four query-time stages: evidence-demand planning, sequential grounded execution, port-wise full-corpus candidate generation, and port-balanced budget assembly. Panels 2 and 3 separate the per-port execution cycle for visualization; at runtime, grounding, retrieval, and judging are interleaved in dependency order. An LLM provides question-only planning and grounded passage judging, while the dense retriever and index remain frozen. The plan remains fixed after generation; passage-extracted upstream values fill symbolic slots in downstream ports during execution. Appendix A gives the complete execution procedure.

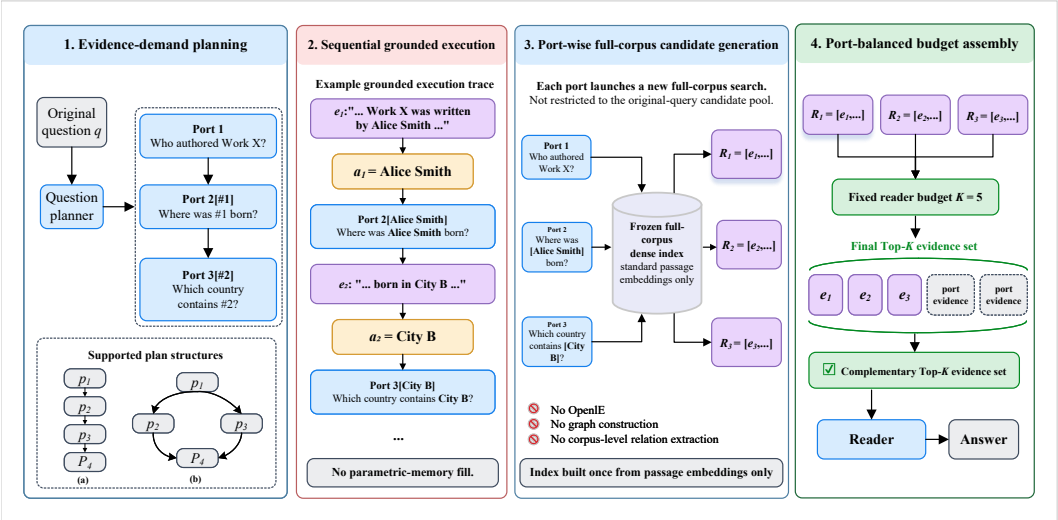


Figure 2: EviPort query-time pipeline: evidence-demand planning, sequential grounded execution, independent full-corpus retrieval for each instantiated port, and port-balanced Top-K assembly.

4.2 Evidence-Demand Planning

The planner receives only q and produces an ordered acyclic plan

$$\mathcal{P}(q) = (p_1, \ldots, p_m), \qquad p_j = (u_j, \mathrm{pa}(j)), \tag{5}$$

where u_j is an atomic searchable question and $\mathrm{pa}(j) \subseteq \{1, \ldots, j-1\}$ contains its predecessor ports. A downstream template may refer to an upstream answer using a symbolic slot such as `#1`. The planner is prompted to generate a minimal plan intended to cover the required evidence, with one evidence-bearing fact per port and no final synthesis-only step. Our current implementation permits one to four ports and accepts any predecessor pattern whose references point backward.

Planning is corpus-independent: the planner cannot copy gold intermediate answers or adapt a decomposition to known supporting passages. Thus, plan generation is independent of evaluation labels.

4.3 SEQUENTIAL GROUNDED EXECUTION

Ports execute in dependency order. At port j, predecessor slots are first replaced by passage-extracted answers:

$$A_j = \{i \mapsto a_i \mid i \in \mathrm{pa}(j)\}, \qquad \tilde{u}_j = \mathrm{Bind}(u_j, A_j). \tag{6}$$

Thus, each symbolic slot `#i` is replaced by the answer associated with predecessor i, including when a port has multiple parents. The instantiated query then searches the corpus,

$$R_j^{(0)} = \underset{d \in \mathcal{D}}{\arg\,\mathrm{topL}}\, s(\tilde{u}_j, d), \tag{7}$$

and a passage judge receives q, $\tilde{u}_j$, and the first $J \leq L$ candidates. It selects an evidence passage e_j, extracts a short answer a_j, and returns a support flag z_j. Descendant ports may use a_j only when it was extracted from the selected passage; the executor never fills a missing identity from parametric memory. If z_j is false, the frozen implementation records an unsupported binding and continues without retry or substitution. Appendix A specifies this deterministic failure behavior, and Appendix C gives the judge prompt and validation rules.

4.4 PORT-WISE FULL-CORPUS CANDIDATE GENERATION

Each instantiated port launches an independent search over $\mathcal{D}$. Grounding can therefore move a downstream query into a semantic region that is poorly represented in the original-question ranking; candidate generation is not restricted to a shared initial pool. EviPort uses the same frozen passage index for every port and requires no corpus graph, relation extraction, or entity linking.

The judge refines only the ordering within its port list. The selected passage is placed first and the remaining candidates preserve their dense order:

$$\widehat{R}_j = [e_j] \,\|\, \left(R_j^{(0)} \setminus \{e_j\}\right). \tag{8}$$

Thus, each $\widehat{R}_j$ remains an independently generated candidate list associated with one evidence demand.

4.5 PORT-BALANCED BUDGET ASSEMBLY

Pooling all candidates and sorting by a single score can let one easy or lexically salient demand consume the context budget. The frozen EviPort allocator instead traverses $\widehat{R}_1, \ldots, \widehat{R}_m$ round by round, adding the next unseen passage from each list until K passages are selected. If duplicates or short lists leave open slots, they are filled from the original-question dense ranking R_q.

Round-robin is a deterministic, completeness-oriented heuristic rather than a learned global optimizer. It preserves representation across evidence demands while retaining dense retrieval as a fallback; Section 6 compares alternative allocation rules.

4.6 COMPUTATION AND INDEX REQUIREMENTS

EviPort adds one query embedding, one nearest-neighbor search, and one candidate-judging LLM call per executed port, plus one planning call. It reuses a standard passage-embedding index and requires no LLM-based corpus extraction, entity linking, proposition generation, or graph materialization. The method therefore incurs higher online query cost than one-shot dense retrieval but no graph-construction or graph-update cost. Section 6.5 and Appendix H report retrieval-to-context latency; one-time index construction is outside that timing boundary.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Datasets. We evaluate on HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022). Following the 1,000-question evaluation protocol used by

HippoRAG 2 (Jiménez Gutiérrez et al., 2025), we use a fixed 1,000-question set for each benchmark, together with the original paragraph retrieval units and supporting-passage identifiers. The HotpotQA, 2WikiMultiHopQA, and MuSiQue corpora contain 9,811, 6,119, and 11,656 passages, respectively. Retrieval is corpus-wide rather than restricted to each question's annotated supporting and distractor passages.

Because MuSiQue-1K traces were repeatedly inspected during mechanism development, we treat it as a development/audit set rather than untouched held-out evidence. HotpotQA and 2WikiMultiHopQA provide frozen cross-dataset transfer evaluations. Appendix B details the hop distribution, frozen transfer protocol, and shared reader configuration.

Baselines. We compare lexical BM25 (Robertson & Zaragoza, 2009); Contriever (Izacard et al., 2022); Qwen3-Embedding-8B (Zhang et al., 2025); NV-Embed-v2 (Lee et al., 2025); IRCoT (Trivedi et al., 2023); GraphRAG (Edge et al., 2024); HippoRAG (Jiménez Gutiérrez et al., 2024); and HippoRAG 2 (Jiménez Gutiérrez et al., 2025). EviPort uses NV-Embed-v2 as its frozen dense backbone. Graph-based baselines retain their native retrieval algorithms, but every retrieved five-passage context is evaluated by the same reader protocol.

Metrics. Retrieval quality is measured by passage Recall@5. Where passage identifiers are directly available, we additionally analyze AnyEvidence@5 and FullEvidence@5. End-to-end answer quality is measured by normalized exact match (EM) and token F1. All methods receive at most five passages in the final reader context.

Implementation. The frozen EviPort configuration uses NV-Embed-v2 for query–passage retrieval and Qwen3.5-9B (Qwen Team, 2026) for planning, passage judging, and reading. Each port retrieves $L = 5$ passages; the judge considers the top three; and the final allocator returns $K = 5$ unique passages. No online method component is given access to gold decompositions, supporting-passage identifiers, intermediate answers, or final answers. Across HotpotQA, 2WikiMultiHopQA, and MuSiQue, all reported EM/F1 values use the same Qwen3.5-9B reader checkpoint, one-shot prompt, temperature-zero decoding, five-passage budget, answer normalization, and scoring code. Exact planner and judge prompts, validation rules, and decoding limits are given in Appendix C.

5.2 Experimental results

Table 1: Retrieval and end-to-end QA on three multi-hop benchmarks. Bold marks the best score per dataset.

	HotpotQA-1K			2Wiki-1K			MuSiQue-1K		
Method	R@5	EM	F1	R@5	EM	F1	R@5	EM	F1
BM25	74.8	52.2	62.4	65.5	48.0	50.7	43.2	20.4	26.8
Contriever	75.0	51.0	61.5	58.2	38.0	40.6	47.1	23.9	29.6
Qwen3-Embedding-8B	95.0	60.5	73.8	72.8	53.9	58.2	61.7	29.2	35.7
NV-Embed-v2	94.8	60.3	72.6	76.5	56.4	60.8	69.3	33.1	40.5
IRCoT	95.3	61.6	73.7	91.0	46.3	52.1	74.1	32.5	38.9
GraphRAG	–	54.5	67.9	–	50.6	57.4	–	27.7	38.2
HippoRAG	78.4	51.4	61.9	90.1	63.5	70.5	52.3	25.3	31.5
HippoRAG 2 + NV	95.8	60.7	74.5	90.7	64.8	71.5	74.0	36.0	45.0
EviPort	**96.7**	**65.2**	**81.4**	**92.3**	**70.7**	**79.3**	**81.7**	**45.3**	**55.0**

Table 1 shows that EviPort achieves the highest scores among the evaluated systems on all three datasets. Relative to HippoRAG 2 + NV, EviPort improves Recall@5 by 0.9, 1.6, and 7.7 points on HotpotQA, 2WikiMultiHopQA, and MuSiQue, respectively. The corresponding EM gains are 4.5, 5.9, and 9.3 points, while F1 improves by 6.9, 7.8, and 10.0 points under the common reader protocol.

With the shared NV-Embed-v2 backbone, EviPort gains 1.9, 15.8, and 12.4 Recall@5 points on HotpotQA, 2WikiMultiHopQA, and MuSiQue. Identical passage embeddings rule out backbone strength as the explanation; the pattern is consistent with query compilation, grounded binding, and

fixed-budget evidence construction. The hop-stratified QA results in Appendix Table 6 show that the improvement persists from two-hop through four-hop questions.

6 Analysis

We use the MuSiQue-1K development/audit set to answer four questions about EviPort's behavior: whether gains come from first-hit discovery or complete evidence recovery; whether they persist under tighter context budgets; which retrieval mechanisms create the gain; and how much performance is lost during final allocation. Figure 3 provides the overview, while the following subsections isolate each question with matched interventions and transfer controls. Exact budget- and hop-stratified values are reported in Appendix Table 5.

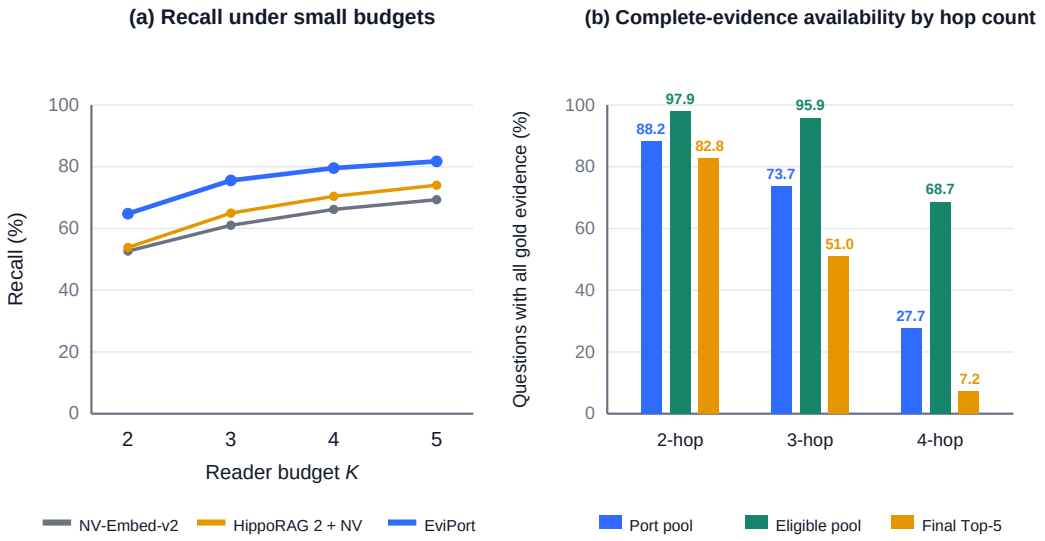


Figure 3: Budget sensitivity and evidence-set diagnostics on MuSiQue. **(a)** Recall across reader budgets K=2–5. **(b)** Complete annotated-evidence availability across retrieval and selection stages, stratified by hop count.

6.1 First-hit recall versus complete evidence

EviPort improves evidence-set completion rather than first-hit discovery. These results use the complete 1,000-question MuSiQue development/audit set, not a smaller subset. NV-Embed-v2, HippoRAG 2 + NV, and EviPort retrieve at least one annotated passage for 98.7, 98.6, and 98.5 percent of questions, respectively. Their FullEvidence@5 rates are much farther apart: 37.4, 46.9, and 60.2 percent. The main gain therefore comes from recovering complementary support after a first hit, not from converting complete misses into hits. Appendix Table 5 reports these values together with the budget- and hop-stratified recall computed from the same per-question retrieval traces.

FullEvidence remains an annotated-evidence-completeness proxy rather than a semantic sufficiency certificate. Nevertheless, readers are substantially more accurate when the complete annotated set is present, while conditional accuracy is similar across retrievers (Appendix Table 8). This connects EviPort's higher evidence-set completion rate to its aggregate QA improvement without claiming that annotated completeness is necessary or sufficient for a correct answer.

6.2 Evidence coverage under tight budgets

The advantage appears before the context budget reaches five passages. Figure 3(a) evaluates prefixes of the same executed rankings, changing the reader budget but not retrieval computation.

EviPort leads both comparators for every K from two to five; the largest relative separation is already visible at $K = 2$, where its Recall advantage over HippoRAG 2 + NV is 10.95 points. The gain is therefore not an artifact of receiving more context slots. Figure 3(b) shows the complementary limitation: complete evidence degrades sharply with hop count, and only 7.23 percent of four-hop questions retain all annotated support in the final Top-5. Appendix Table 5 provides the exact curves and hop-stratified values.

6.3 Grounded ports expand the candidate space

Grounded binding is the dominant candidate-generation mechanism. Table 2 changes one component at a time. Replacing upstream answers with an unresolved placeholder lowers Recall@5 from 81.74 to 63.22 and FullEvidence@5 from 60.20 to 29.10. Restricting bound ports to the original-question Dense-100 pool costs a further 3.10 Recall points, showing that grounded queries recover evidence beyond a large shared initial pool. By comparison, removing within-port candidate judging costs only 1.25 Recall points; judging is useful but secondary to binding and independent retrieval.

Table 2: Mechanism interventions on MuSiQue-1K. ΔR is EviPort Recall@5 minus the intervention.

Variant	R@5	Full@5	ΔR
No grounded binding	63.22	29.10	18.52
bound ports in Dense-100	78.64	55.30	3.10
No ranking judge	80.49	57.40	1.25
Plan-order truncation (no balancing)	47.74	11.80	34.00
EviPort	**81.74**	**60.20**	–

Table 3: Frozen transfer of the candidate-generation interventions on 2Wiki-1K.

Condition	R@5	Full@5	ΔR
NV-Embed-v2	76.53	50.40	15.75
No grounded binding	72.85	42.80	19.43
bound ports in Dense-100	86.98	69.80	5.30
EviPort	**92.28**	**81.90**	–

As shown in Table 3, the same interventions transfer to the frozen 2Wiki evaluation: removing binding costs 19.43 Recall points, and restricting bound ports to Dense-100 costs 5.30 points. These effects support the candidate-generation mechanism outside the inspected MuSiQue audit. Appendix Table 12 separately estimates the remaining headroom in planning, binding, and reading.

6.4 Fixed-budget allocation remains a bottleneck

No tested selector dominates every evidence-set objective. For the selector rows in Table 4, plans, bindings, and candidate lists are identical; only final fusion changes. The iterative control instead matches EviPort's full-corpus search and LLM-call counts. Every QA result uses the common reader and five-passage budget.

Table 4: Candidate fusion and matched iterative controls on MuSiQue-1K.

Method	R@5	Full@5	EM	F1
Search/call-matched iterative	78.53	53.00	41.50	50.83
RRF over port lists	80.38	57.60	44.10	53.21
Port heads + dense fill	**81.86**	59.10	**46.80**	**56.15**
EviPort round-robin	81.74	**60.20**	45.30	55.02

The matched iterative baseline is 3.21 Recall, 7.20 FullEvidence, 3.80 EM, and 4.19 F1 points below EviPort, so additional retrieval rounds alone do not explain the gains. Among selectors,

port-heads+dense-fill slightly improves fractional recall and QA, whereas round-robin retains 1.10 points more FullEvidence. We therefore use round-robin as a completeness-oriented default, not as a universally optimal allocator.

Candidate generation remains the main source of improvement over one-shot dense retrieval, but allocation is not solved. The broad diagnostic pool contains all annotated evidence for 92.40 percent of questions, whereas only 60.20 percent retain the complete set after five-slot selection (Appendix Table 11). Because that pool includes the Dense Top-100 fallback, it is an availability upper bound rather than the exact prefix traversed online. Appendix Tables 9–11 report the selector, hop-specific intervention, and failure-decomposition results behind this conclusion.

6.5 Efficiency

EviPort reuses the dense passage index and requires no OpenIE or graph-construction stage, but it performs more online computation than one-shot dense retrieval. On a prospectively fixed 100-question subset, EviPort is 1.53× slower than the search-and-call-matched iterative control while recovering more complete evidence. Appendix H specifies the matched search and model-call counts, timing boundary, repeated measurements, and retrieval results.

7 Limitations and Future Directions

Two directions follow from the current results. First, EviPort avoids corpus-graph construction by performing more work at query time; its latency grows with the number of ports, and the present evaluation covers corpora with thousands rather than millions of passages. Batching port searches, caching shared prefixes, and testing approximate retrieval at larger scale are natural next steps. Second, the frozen executor intentionally uses a single plan without retry or topology repair, so an unsupported upstream extraction can affect dependent ports. Confidence-aware binding, local replanning, and adaptive allocation could improve robustness on the four-hop cases that remain difficult here.

8 Conclusion

We introduced EviPort, a framework for constructing multi-hop evidence sets without corpus-graph construction. EviPort represents a question as dependent evidence ports, binds later ports with passage-grounded upstream answers, retrieves from the full corpus for each instantiated demand, and allocates context slots across port-specific rankings. Across three multi-hop QA benchmarks, this approach improves both retrieval and end-to-end answering over strong dense, iterative, and graph-based baselines. The candidate-pool analysis also exposes the next bottleneck: making all annotated supporting evidence available within a broad diagnostic pool does not guarantee that a five-passage selector will retain it. These results point to two complementary directions for multi-hop retrieval: demand-conditioned candidate generation and stronger set-level allocation.

AI use statement

Generative AI tools were used to assist with exploratory code development, experimental-design feedback, implementation, figure ideation, analysis and drafting, and language editing. All experimental configurations, numerical results, citations, and scientific claims were independently reviewed and verified by the authors. The authors take full responsibility for the final content.

Ethics statement

This work studies retrieval algorithms on established question-answering benchmarks and does not involve human-subject data collection. Like other RAG systems, EviPort may retrieve inaccurate, biased, private, or malicious source text when deployed on uncontrolled corpora. Evidence provenance, access control, and downstream verification remain necessary in high-stakes applications.

Reproducibility statement

Appendices A–C document the complete execution procedure, evaluation protocol, prompts, validation rules, and frozen hyperparameters. Appendices D–H provide the numerical values underlying Figure 3, reader sensitivity and completeness analyses, additional mechanism controls, oracle diagnostics, and runtime measurements. The supplementary archive provides anonymized source code for evidence-port planning, grounded execution, transfer evaluation, and bootstrap analysis, together with the paper source and aggregate result tables. Benchmark corpora and pretrained checkpoints are obtained from their official distributions and are not redistributed. Gold decompositions, evidence identifiers, intermediate answers, and final answers remain isolated from all online method components.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR*, pp. 335–336, 1998.

Dheeru Dua, Shivanshu Gupta, Sameer Singh, and Matt Gardner. Successive prompting for decomposing complex questions. In *Proceedings of EMNLP*, pp. 1251–1265, 2022.

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.

Jennifer Gillenwater, Alex Kulesza, Zelda Mariet, and Sergei Vassilvitskii. A tree-based method for fast repeated sampling of determinantal point processes. In *Proceedings of ICML*, pp. 2260–2268, 2019.

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

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of COLING*, 2020.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of EACL*, pp. 874–880, 2021.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of NAACL*, pp. 7036–7050, 2024.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of EMNLP*, pp. 7969–7992, 2023.

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*, 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*, pp. 21497–21515, 2025.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*, 2020.

Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of SIGIR*, pp. 39–48, 2020.

Omar Khattab, Christopher Potts, and Matei Zaharia. Baleen: Robust multi-hop reasoning at scale via condensed retrieval. In *Advances in Neural Information Processing Systems*, volume 34, pp. 27670–27682, 2021.

Jueun Kim, Sungho Park, and Wook-Shin Han. Hi-Q: Hierarchical evidence-guided query refinement for multi-hop question answering. *arXiv preprint arXiv:2608.30468*, 2026.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012. doi: 10.1561/2200000044.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-Embed: Improved techniques for training LLMs as generalist embedding models. In *International Conference on Learning Representations*, 2025.

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

Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, and Yongbin Li. StructRAG: Boosting knowledge intensive reasoning of LLMs via inference-time hybrid information structurization. In *International Conference on Learning Representations*, 2025.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of ACL*, pp. 510–520, 2011.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. In *International Conference on Learning Representations*, 2025.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. Unsupervised question decomposition for question answering. In *Proceedings of EMNLP*, pp. 8864–8880, 2020.

Peng Qi, Haejun Lee, Tg Sido, and Christopher D. Manning. Answering open-domain questions of varying reasoning steps from text. In *Proceedings of EMNLP*, pp. 3599–3614, 2021.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019.

Pavan Kumar Santhana Krishnan, Kiran Kumar Nakka, C. Reddy, Divyateja Pasupuleti, Prakhar Agarwal, Harpinder Singh, Anshu Avinash, and Nirav Bhatt. BrowseNet: Graph-based associative memory for contextual information retrieval. In *International Conference on Learning Representations*, 2026.

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*, 2024.

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.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of ACL*, pp. 10014–10037, 2023.

Jingjin Wang and Jiawei Han. PropRAG: Guiding retrieval with beam search over proposition paths. In *Proceedings of EMNLP*, pp. 6212–6227, 2025.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. BREAK it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198, 2020.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2021a.

Wenhan Xiong, Xiang Lorraine Li, Srinivasan Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oğuz. Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations*, 2021b.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. If you want to go far go together: Unsupervised joint candidate evidence retrieval for multi-hop question answering. In *Proceedings of NAACL*, pp. 4571–4581, 2021.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of EMNLP*, 2018.

Linhao Ye, Lang Yu, Zhikai Lei, Qin Chen, Jie Zhou, and Liang He. Optimizing question semantic space for dynamic retrieval-augmented multi-hop question answering. In *Proceedings of ACL*, pp. 17814–17824, 2025.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

A EviPort Algorithm

Given q, $\mathcal{D}$, port retrieval depth L, and final budget K:

1. Generate a validated ordered plan $\mathcal{P}(q) = (p_1, \ldots, p_m)$ using only q.
2. For $j = 1, \ldots, m$:
   (a) form $A_j = \{i \mapsto a_i \mid i \in \mathrm{pa}(j)\}$ and bind it into u_j to obtain $\tilde{u}_j$;
   (b) retrieve $R_j^{(0)} = \mathrm{TopL}_{d \in \mathcal{D}}\, s(\tilde{u}_j, d)$;
   (c) judge the top candidates and return selected passage e_j, answer string a_j, and support flag z_j;
   (d) if z_j is false, record an unsupported binding, retain e_j, and continue without retry or parametric-memory substitution;
   (e) construct $\widehat{R}_j = [e_j] \,\|\, (R_j^{(0)} \setminus \{e_j\})$.
3. Traverse $\widehat{R}_1, \ldots, \widehat{R}_m$ round by round, adding unseen passages until K are selected.
4. Fill any remaining positions from the original-query dense ranking R_q.

B Evaluation Protocol Details

Each benchmark uses a fixed 1,000-question evaluation set. The HotpotQA, 2WikiMultiHopQA, and MuSiQue corpora contain 9,811, 6,119, and 11,656 original paragraph units, respectively. The MuSiQue audit contains 518 two-hop, 316 three-hop, and 166 four-hop questions. Because its traces were repeatedly inspected during mechanism development, we treat MuSiQue as a development/audit set rather than untouched held-out evidence. The EviPort configuration was then frozen before HotpotQA and 2Wiki transfer evaluation. In particular, the 2Wiki binding and Dense-100 interventions were specified before their outputs were inspected and use all 1,000 frozen question identifiers.

The reader is Qwen3.5-9B with temperature zero, seed 20260904, and a maximum of 2,048 generated tokens. Every EM/F1 value in Table 1 uses the same reader checkpoint, one-shot QA prompt, five-passage budget, normalization, and scorer; retrieval algorithms and embedding backbones remain method-specific. Gold evidence is used only for offline scoring or in explicitly identified oracle diagnostics.

C Prompt Templates and Frozen Hyperparameters

Question-only planner. The planner receives only the original question. The operative instruction is:

```
Create the smallest evidence topology needed.
Do not answer the question or fill unknown facts
from memory. Each step asks for one fact supported
by one passage. A later step may refer to an earlier
short answer using #1, #2, etc. Use 1--4 steps and
allow parallel branches. Return topology, step id,
question, dependencies, and output type as JSON.
```

Grounded passage judge. For the original question, current bound subquestion, and three candidate passages, the judge is instructed:

```
Choose the single passage that best directly supports
the current subquestion. Extract a short answer only
from that passage. Do not use outside knowledge. If
none supports an answer, set supported=false but still
choose the most promising passage. Return best_passage,
answer, and supported as JSON.
```

Planner and judge decoding use temperature zero. The planner is capped at 650 generated tokens and the judge at 120. The frozen retriever uses NV-Embed-v2, retrieves five passages per port, judges the top three, and assembles five unique passages for the reader. Plans contain at most four ports. The planner's returned `output_type` field is not consumed by retrieval, judging, binding, or allocation and is therefore omitted from the method formalization. The frozen method does not revise, insert, delete, or backtrack over ports after planning.

When the judge returns `supported=false`, the executor still retains its most promising passage and uses the returned answer string, which may be empty. The same string is inserted into any descendant slot; there is no retry, model-memory fill, or topology repair. The selected passage remains eligible for final allocation. Search and judge counts remain fixed, but an unsupported value can propagate to later ports.

Every generated plan passes a deterministic syntax validator before retrieval. The validator keeps at most four non-empty steps, renumbers them consecutively, derives dependencies from explicit `#n` references, and requires every reference to point to an earlier step. An invalid forward reference is replaced by the neutral phrase "the required entity"; an empty plan falls back to a single port containing the original question. On MuSiQue-1K, one plan (0.1%) required an invalid-reference repair and none used the empty-plan fallback.

D COMPLETE SMALL-BUDGET RETRIEVAL RESULTS

Figure 3 summarizes the small-budget trend and the loss of complete evidence as hop count grows. Table 5 reports the exact aggregate and hop-stratified values underlying that figure. It also distinguishes AnyEvidence@5, which is already saturated for all three methods, from fractional recall and complete-evidence recovery.

FullEvidence is structurally unattainable whenever the passage budget is below the number of distinct annotated supporting passages. Such cells should not be interpreted as a retriever failure. Prefix evaluation changes the final reader budget while retaining the frozen executed plan; it does not measure a reduced online computation budget.

Table 5: MuSiQue-1K evidence coverage by method, passage budget, and annotated hop count (percent). Any@5 is reported for the aggregate rows.

Method	Group	N	Any@5	R@1	R@2	R@3	R@4	R@5	Full@5
NV-Embed-v2	All	1000	98.70	35.36	52.68	61.03	66.18	69.33	37.40
HippoRAG 2 + NV	All	1000	98.60	32.60	53.85	64.98	70.43	74.01	46.90
EviPort	All	1000	98.50	35.32	64.80	75.56	79.57	81.74	60.20
NV-Embed-v2	2-hop	518	–	43.73	62.64	70.75	75.39	77.41	55.98
	3-hop	316	–	30.17	48.10	58.65	64.35	68.78	25.95
	4-hop	166	–	19.13	30.27	35.24	40.96	45.18	1.20
HippoRAG 2 + NV	2-hop	518	–	40.64	67.08	77.90	83.01	85.42	72.20
	3-hop	316	–	27.53	45.57	59.18	64.56	70.04	28.16
	4-hop	166	–	17.17	28.31	35.69	42.32	45.93	3.61
EviPort	2-hop	518	–	44.59	81.76	87.36	89.29	90.83	82.82
	3-hop	316	–	29.32	52.85	72.57	78.06	80.59	50.95
	4-hop	166	–	17.77	34.64	44.43	52.11	55.57	7.23

E READER SENSITIVITY AND EVIDENCE COMPLETENESS

The main paper reports aggregate retrieval and QA. This section tests whether the gains persist across hop counts, under a longer generation cap, and after changing passage order. It then measures the association between complete annotated evidence and answer accuracy.

E.1 PERFORMANCE BY HOP COUNT

Table 6 breaks QA down by annotated hop count. EviPort improves EM and F1 in every group, including the four-hop questions for which complete-evidence recovery is hardest.

Table 6: End-to-end QA by annotated hop count.

Method	Group	N	EM	F1
NV-Embed-v2	2-hop	518	40.73	47.31
HippoRAG 2 + NV	2-hop	518	46.91	56.03
EviPort	2-hop	518	53.09	63.04
NV-Embed-v2	3-hop	316	30.06	40.96
HippoRAG 2 + NV	3-hop	316	28.16	39.68
EviPort	3-hop	316	44.94	55.00
NV-Embed-v2	4-hop	166	15.06	18.14
HippoRAG 2 + NV	4-hop	166	16.87	20.95
EviPort	4-hop	166	21.69	30.04

E.2 GENERATION-CAP AND ORDERING SENSITIVITY

To test whether the aggregate gains depend on the 2,048-token generation cap, we froze a prospective, hop-stratified 300-question sample before generation (155 two-hop, 95 three-hop, and 50 four-hop questions) and reran all methods with a shared 4,096-token cap. EviPort exceeds NV-Embed-v2 by 9.0 EM and 10.98 F1 points in Table 7. Reversing the *same selected EviPort set* changes EM/F1 by only $+2.67/+1.77$ points, so the QA advantage is not explained by the default presentation

order alone. This experiment tests a longer generation cap and ordering sensitivity; it is not a paired comparison against a 2,048-token setting on the same 300 questions.

Table 7: Prospective 4,096-token reader and ordering control on 300 MuSiQue questions. All methods use five passages, the same reader and prompt, temperature zero, and seed 20260913.

Context	EM	F1	Truncated	Parse fallback
NV-Embed-v2	35.00	43.74	40	17
HippoRAG 2 + NV	36.67	46.63	42	23
EviPort	44.00	54.71	28	7
EviPort, reverse order	**46.67**	**56.48**	30	14

E.3 Association between annotated completeness and QA

Table 8 tests whether FullEvidence@5 is informative about downstream answering rather than merely an annotation-centric retrieval score. For every retriever, answer accuracy is much higher when the complete annotated set is present. Conditional on FullEvidence, EM ranges from 60.30 to 63.10 and F1 from 69.92 to 71.19 across retrievers. The similar conditional accuracy is consistent with EviPort's aggregate QA gain being associated with more frequent complete-support contexts. These observational associations support FullEvidence as a predictive completeness proxy, while showing that it is neither necessary nor sufficient for a correct answer.

Table 8: QA conditioned on whether all annotated supporting passages occur in the five-passage context. These observational associations do not establish intervention effects.

Method	Full rate	EM|Full	EM|Incomplete	F1|Full	F1|Incomplete	ΔF1
NV-Embed-v2	37.40	63.10	15.18	71.19	22.10	49.09
HippoRAG 2 + NV	46.90	60.55	14.31	69.92	23.06	46.86
EviPort	60.20	60.30	22.61	70.77	31.20	39.57

F Additional Mechanism Controls

This section defines the controls summarized in main-paper Tables 2 and 4, then examines whether their behavior changes with hop count. For the no-ranking-judge intervention, each port commits its dense Top-1 passage; the model is retained only to extract the passage-supported value required by later ports. The intervention therefore removes candidate promotion rather than the extraction needed for dependency binding. Plan-order truncation concatenates port lists and cuts at $K = 5$; its loss shows that this policy is unsuitable for these candidate lists, not that round-robin dominates every fusion rule.

The port-heads+dense-fill and reciprocal-rank-fusion controls in Table 4 reuse exactly the same plans, bindings, and port candidate lists as EviPort. Reciprocal-rank fusion uses $k = 60$, fixed before evaluation; port heads + dense fill reserves one passage per port before consulting the original-query dense ranking.

F.1 Search-and-call-matched iterative control

The iterative control receives the original question but never EviPort's port text or dependencies. It first uses one LLM call to produce a concise search query. For exactly P rounds, where P is the frozen EviPort plan length for that question, it embeds the current query, searches the full corpus, shows the top three passages to the same Qwen3.5-9B model, selects one passage-grounded fact, and generates the next query for the most important unresolved fact. The final context applies the same five-passage limit. Consequently, the 1,000-question comparison has 1,000 initial query calls and 2,525 retrieval/reasoning calls, exactly matching EviPort's one planner call per question and 2,525 port executions. Temperature is zero; initial and round outputs are capped at 80 and 180 tokens. The control matches search and LLM-call counts, not token volume, wall-clock time, or FLOPs.

The retained 1,000-question contexts were evaluated with the same Qwen3.5-9B reader, one-shot prompt, temperature-zero decoding, 2,048-token output cap, and answer scorer used for EviPort.

Table 4 shows that explicit evidence demands improve both annotated evidence recovery and downstream QA over this search-and-call-matched control.

F.2 Alternative frozen selection policies

Table 9 compares policies over the same frozen candidate lists as the reader budget grows. Port heads + dense fill slightly improves fractional Recall@5, whereas round-robin retains more complete evidence. The last-port and MMR controls remain far below policies that reserve capacity across several demands. The single MMR setting does not establish that every parameterization must fail; it shows that generic embedding diversity does not replace port coverage in this configuration. These results motivate a completeness-oriented round-robin default while leaving learned or adaptive set allocation open.

Table 9: Frozen selection policies under the same five-passage budget. Dashes indicate unavailable prefix measurements.

Policy	R@1	R@2	R@3	R@4	R@5	Full@5
EviPort round-robin	35.32	64.80	75.56	79.57	81.74	60.20
Within-port dense order	35.54	63.74	73.40	78.12	80.76	57.80
Plan-order truncation	35.32	41.83	44.19	46.64	47.74	11.80
Port heads + dense fill	35.32	65.38	76.24	79.92	81.86	59.10
Last port + dense fill	32.17	44.75	49.65	53.64	56.00	27.50
MMR ($\lambda = 0.5$)	–	–	–	–	57.11	23.90

F.3 Mechanism interventions by hop count

Table 10 shows that removing grounded binding or using plan-order truncation causes the largest losses in every hop group. The latter comparison should not be read as proof that round-robin is the optimal allocator; the strong port-head control slightly improves QA. Four-hop FullEvidence remains low even for EviPort (7.23 percent), despite a 9.64-point Recall advantage over HippoRAG 2 + NV. Long-chain completion, rather than first-hit discovery, is therefore the dominant remaining retrieval bottleneck.

Table 10: Mechanism interventions by hop count (recall/full-evidence, percent).

Method	2R	2F	3R	3F	4R	4F
EviPort	90.83	82.82	80.59	50.95	55.57	7.23
No grounded binding	70.85	43.05	62.66	20.57	40.51	1.81
Bound, Dense-100 pool	87.45	76.06	77.74	46.20	52.86	7.83
No ranking judge	90.35	82.05	78.27	44.30	55.57	7.83
Plan-order truncation	60.23	22.39	37.55	0.63	28.16	0.00

F.4 Planner and binding failure decomposition

The frozen planner produces 1, 2, 3, and 4 ports for 7, 577, 300, and 116 questions, respectively (0.7%, 57.7%, 30.0%, and 11.6%). Execution follows explicit predecessor references rather than the planner's categorical topology label. Those references yield 875 strict chains, 118 non-chain plans, and 7 single-port plans. Across the full set, 92 plans have multiple roots, 33 contain fan-out, and 54 contain a multi-parent merge; these categories overlap.

Table 11 relates plan length and binding support to candidate availability and final complete-evidence recovery. Its port pool is the union of the $L = 5$ candidates retrieved for every port; the diagnostic pool additionally includes the Top-100 original-query dense fallback. More than half of four-hop questions are under-decomposed, only 27.71% place all gold passages in the port pool, and unsupported bindings are more than twice as frequent as in the aggregate. Across all questions, the broad diagnostic pool contains every annotated passage for 92.40%, yet the five-slot output retains the complete set for only 60.20%. This 32.20-point gap indicates substantial room for improved fixed-budget selection, but it is not a pure allocator-error rate because the diagnostic pool may exceed the dense prefix traversed online.

Table 11: Post-hoc failure indicators on MuSiQue-1K. Indicators overlap and are not oracle interventions.

Indicator (rate, %)	All	2-hop	3-hop	4-hop
FullEvidence success	60.20	82.82	50.95	7.23
Plan hop-count match	75.50	91.89	63.61	46.99
Plan under-decomposed	17.40	1.35	25.00	53.01
All gold in port pool	73.60	88.22	73.73	27.71
All gold in diagnostic pool	92.40	97.88	95.89	68.67
Unsupported binding present	8.40	5.60	7.91	18.07

G Oracle Component Diagnostics

Table 12 isolates headroom in planning, binding, and reading; these annotation-assisted conditions are diagnostics rather than retrieval baselines. Replacing the predicted plan with the annotated decomposition raises Recall@5 from 81.74 to 85.68 and FullEvidence@5 from 60.20 to 67.70. Using annotated predecessor answers as well raises the scores to 90.00 and 74.20. Both conditions otherwise retain the same corpus, NV-Embed-v2 index, passage judge, allocator, and 1,000 questions as the main evaluation.

Supplying all annotated supporting passages directly to the unchanged reader gives 54.60 EM and 64.19 F1, 9.30 EM and 9.17 F1 points above EviPort but still far below perfect answering. FullEvidence is therefore a useful annotated-evidence completeness proxy, but completeness is neither necessary nor sufficient for exact answer generation.

Table 12: Oracle component diagnostics on MuSiQue-1K. These rows use annotation-derived information and are excluded from the unsupervised comparison.

Condition	R@5	Full@5	Any@5	EM	F1
Predicted plan and binding	81.74	60.20	98.50	45.30	55.02
Gold plan, predicted binding	85.68	67.70	99.10	–	–
Gold plan and binding	90.00	74.20	99.60	–	–
Gold evidence to reader	100.00	100.00	100.00	54.60	64.19

H Runtime Measurement Boundaries

Table 13 reports three fresh-cache repetitions for the retrieval-to-context stage used in Section 6.5. The iterative control and EviPort use the same NV index, Qwen3.5-9B service, temperature, candidate depth, concurrency, 250 full-corpus searches, and 350 LLM calls per repetition. EviPort is 1.53× slower in the current implementation while improving Recall@5 and FullEvidence@5 by 4.09 and 10.00 points on this prospective subset. Equal invocation counts therefore do not imply equal latency. Shared-reader time is excluded because asynchronous backend serialization and retries did not permit a controlled end-to-end timing comparison.

Table 13: Fresh-cache retrieval-to-context runtime on one H20 GPU for 100 prospective questions. Each repetition uses 350 LLM calls and 250 full-corpus searches; shared-reader time is excluded.

Method	Rep. 1	Rep. 2	Rep. 3	Mean ± SD	sec/q	R@5 / Full@5
Iterative control	490.6	493.9	492.1	492.2 ± 1.7	4.92	75.83 / 46.00
EviPort	757.2	750.3	752.4	753.3 ± 3.5	7.53	79.92 / 56.00