WARP-G: WORKLOAD-AWARE REGIONAL PROVISIONING FOR GRAPHRAG

Anonymous authors
Paper under double-blind review

ABSTRACT

GraphRAG connects evidence across documents through entity relations and graph propagation, but constructing a full-corpus graph can incur substantial offline cost. Efficient GraphRAG methods typically simplify construction or select structurally important content. These approaches do not directly account for uneven query demand or regional differences in the benefit of graph retrieval over a passage baseline. We introduce **WARP-G** (Workload-Aware Regional Provisioning for GraphRAG), which uses an evidence-annotated design workload to select regional graph indexes. WARP-G partitions the corpus using query co-access and weak semantic links, prioritizes a limited set of graph probes by demand, baseline evidence gaps, and regional size, and measures their retrieval gains. It then selects graphs with positive conditional utility on a shared design sample and reuses their indexes at inference time, alongside full-corpus passage retrieval. We evaluate WARP-G with HippoRAG2 on four QA benchmarks. Retained indexes account for 0–14.0% of Full Graph construction tokens, and first-use graph-token cost falls by 38.7–67.2%, including design and online retrieval. On 2WikiMultihopQA, WARP-G retains one graph costing 8.2% of Full Graph construction tokens, while improving ER@10 by 1.00 percentage point and answer F1 by 1.74 points over Full Graph. WARP-G thus reduces graph-related cost while preserving or improving answer quality relative to the passage baseline on the evaluated workloads.

1 INTRODUCTION

Retrieval-augmented generation (RAG) gives language models access to external knowledge that can be updated and traced to its source (Guu et al., 2020; Lewis et al., 2020; Gao et al., 2023). Passage retrievers such as BM25 and Dense Passage Retrieval (DPR) provide a scalable foundation (Robertson & Zaragoza, 2009; Karpukhin et al., 2020), while graph-augmented retrieval connects evidence across documents and reasoning steps (Yang et al., 2018; Xiong et al., 2020). GraphRAG systems organize entities, relations, and passages into graphs, then retrieve through community summaries or graph propagation (Edge et al., 2024; Gutiérrez et al., 2024; 2025; Guo et al., 2025).

This additional structure requires knowledge extraction, embedding, indexing, and sometimes community summarization. Construction cost limits deployment even when passage retrieval remains competitive (Wang et al., 2026a). Graph value also varies across the corpus. Some regions receive few queries; others are frequently accessed but already well served by passage retrieval. Regional graphs target queries whose related evidence is missing or ranked too low by passage retrieval. The deployment objective is to build these indexes where they improve the expected workload. Each region forms a concrete deployment unit whose passages are indexed, evaluated, and retained together.

Existing approaches mainly reduce the cost per graph or select a smaller set of content to index. E^{2}GraphRAG, LinearRAG, and ZoomRAG use lighter representations or processing pipelines (Zhao et al., 2025; Zhuang et al., 2026; Hu et al., 2026); KET-RAG and G2ConS use PageRank or concept centrality to select content (Page et al., 1999; Huang et al., 2025; Liu et al., 2025). WARP-G selects regions by their contribution to the query workload. Figure 1 contrasts these choices: structural criteria select connected content, while demand and measured gain identify regions that improve retrieval for expected queries.

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.

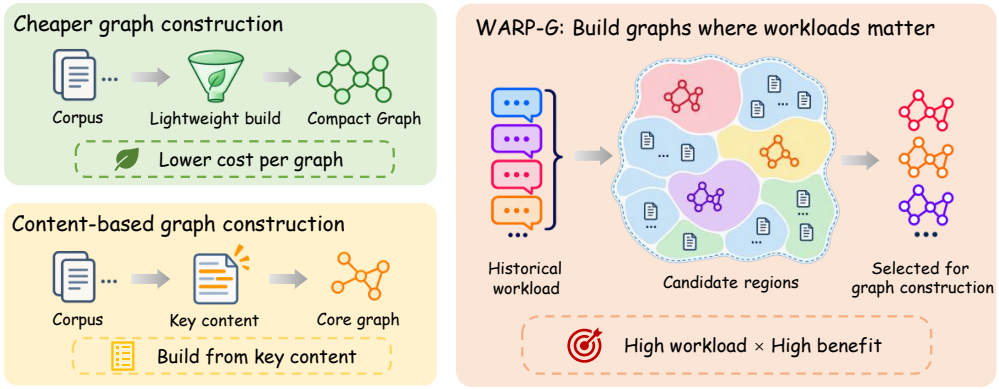


Figure 1: Three approaches to efficient GraphRAG. Lighter construction reduces cost per graph, content-based selection retains structurally important content, and WARP-G uses workload demand and measured retrieval gain to decide which regional graphs to retain.

Demand and regional size are known before construction; retrieval benefit is measured after an index exists. Building every candidate graph to assess its value would consume the resources selective deployment aims to save. We introduce **WARP-G: Workload-Aware Regional Provisioning for GraphRAG**, which separates limited exploration from deployment selection. WARP-G groups passages by historical query co-access and weak semantic links, then prioritizes regional probes using demand, local evidence gaps, and text volume. It evaluates graph combinations on a shared design sample and retains additions with positive conditional utility. This second stage accounts for overlap and competition among graph candidates in the final ranked list.

This formulation connects GraphRAG to workload-aware index and materialized-view selection, where reusable structures are chosen for their workload benefit and resource cost (Harinarayan et al., 1996; Chaudhuri & Narasayya, 1997; Agrawal et al., 2000; Aouiche & Darmont, 2017). It also complements query-time adaptation (Jiang et al., 2023; Asai et al., 2024; Jeong et al., 2024; Dong et al., 2026; Chen et al., 2026): WARP-G determines which indexes are available before serving begins. At inference time, full-corpus passage retrieval activates selected graphs in reached regions, and their candidates are fused and reranked. Selected probe indexes are reused without reconstruction.

With HippoRAG2 on four QA workloads, WARP-G retains zero, one, two, and zero regional graphs on HotpotQA, 2Wiki, MuSiQue, and NQ. It matches or improves Hybrid's answer F1 while reducing first-use graph tokens by 38.7–67.2% relative to Full Graph. Our contributions are:

- We formulate graph construction as workload-aware regional provisioning, using a design workload to choose where to augment passage retrieval.
- We develop limited regional probing followed by conditional selection, separating probe allocation from the joint evaluation of retained indexes.
- We characterize quality and cost across four workloads, linking different deployment outcomes to baseline evidence coverage, regional access, and measured graph gains.

2 Related Work

Graph-augmented retrieval. Graph-augmented retrieval organizes evidence through relations among entities and passages (Peng et al., 2025; Han et al., 2024). Early systems such as GRAFT-Net and PullNet retrieve question-specific subgraphs for multi-hop QA (Sun et al., 2018; 2019). More recent systems use entity communities, graph propagation, or joint graph–text representations, including Microsoft GraphRAG, HippoRAG, LightRAG, and GRAG (Edge et al., 2024; Jeh & Widom, 2003; Gutiérrez et al., 2024; 2025; Guo et al., 2025; Hu et al., 2025). Hierarchical methods such as RAPTOR and SF-RAG further organize evidence across multiple levels (Sarthi et al., 2024; Yu et al., 2026). WARP-G determines where to build graph indexes by grouping passages through query co-access and evaluating regional retrieval gains.

Efficient and selective GraphRAG. Recent work reduces GraphRAG cost through cheaper processing or selective construction. E^{2}GraphRAG, dependency-based enterprise pipelines, LinearRAG, ZoomRAG, and core-based hierarchies reduce construction or retrieval cost with simpler representations, processing, or sampling (Zhao et al., 2025; Min et al., 2025; Zhuang et al., 2026; Hu et al., 2026; Hossain & Sarıyüce, 2026). Other methods reduce graph coverage: KET-RAG and G2ConS select content using PageRank or concept centrality (Page et al., 1999; Huang et al., 2025; Liu et al., 2025). Scaling studies evaluate construction cost alongside retrieval quality (Wang et al., 2026a). WARP-G selects corpus regions using query demand and measured gain over passage retrieval. Limited probes measure individual gains, and conditional selection evaluates graph combinations.

Adaptive retrieval and query routing. Adaptive retrieval changes the retrieval procedure at query time. FLARE, Self-RAG, and Adaptive-RAG adjust retrieval using generation confidence, reflection, or question complexity (Jiang et al., 2023; Asai et al., 2024; Jeong et al., 2024), while other work studies routing strategies and quality–cost trade-offs (Dong et al., 2026; Wang et al., 2026b; Qian et al., 2026). VDGR-RAG combines routing, multiple retrieval channels, and reflection over a hierarchical index (Chen et al., 2026). Query-time adaptation operates over available indexes. WARP-G chooses which regional indexes to build offline, then uses baseline retrieval to activate them at serving time.

Workload-aware physical design. WARP-G is also related to database physical design, where workloads, expected benefit, and resource constraints guide the selection of indexes and materialized views (Harinarayan et al., 1996; Chaudhuri & Narasayya, 1997; Agrawal et al., 2000; Aouiche & Darmont, 2017). Extensions cover learned spatial indexes, view management, and graph partitioning (Sheng et al., 2023; Han et al., 2023; Ciaperoni et al., 2021; Pang et al., 2024; Priyadarshi & Kochut, 2021; 2022). For text-derived regional graphs, retrieval benefit is unknown before construction. WARP-G therefore uses limited probes to measure utility and reuses selected indexes.

3 METHOD

WARP-G treats regional graphs as optional indexes over a full-corpus passage baseline (Figure 2). It partitions passages into regions, probes a limited set of regional graphs, and selects indexes by conditional retrieval gain. The same retrieval pipeline evaluates candidate deployments during design and answers queries at inference time.

3.1 PROBLEM SETUP AND RETRIEVAL UTILITY

Let D be the passage corpus, Q_{des} the evidence-annotated design workload, and $E_q \subseteq D$ the nonempty annotated evidence set for query q. A partition $\mathcal{R} = \{r_1, \ldots, r_m\}$ divides D into disjoint regions, each supporting an independent graph index. Let $S \subseteq \mathcal{R}$ denote the deployed graph set. The passage baseline searches all of D; selecting no graph ($S = \varnothing$) therefore recovers Hybrid retrieval with the same reranker.

For the final top-10 passage set $L_q(S)$, define

$$M(q, S) = \frac{1}{2}\frac{|E_q \cap L_q(S)|}{|E_q|} + \frac{1}{2}\mathbb{I}[E_q \subseteq L_q(S)]. \tag{1}$$

The recall term rewards partial evidence coverage; the completeness term rewards recovering the full annotated evidence set. Selection weights these objectives equally. For a question with two annotated evidence passages, retrieving one gives $M = 0.25$, while retrieving both gives $M = 1$. The completeness term thus rewards finishing an evidence set, beyond the gain from recovering one more passage. Answer EM and token F1 measure downstream quality separately. Evidence labels are used for design queries, with test queries held out until selection is complete.

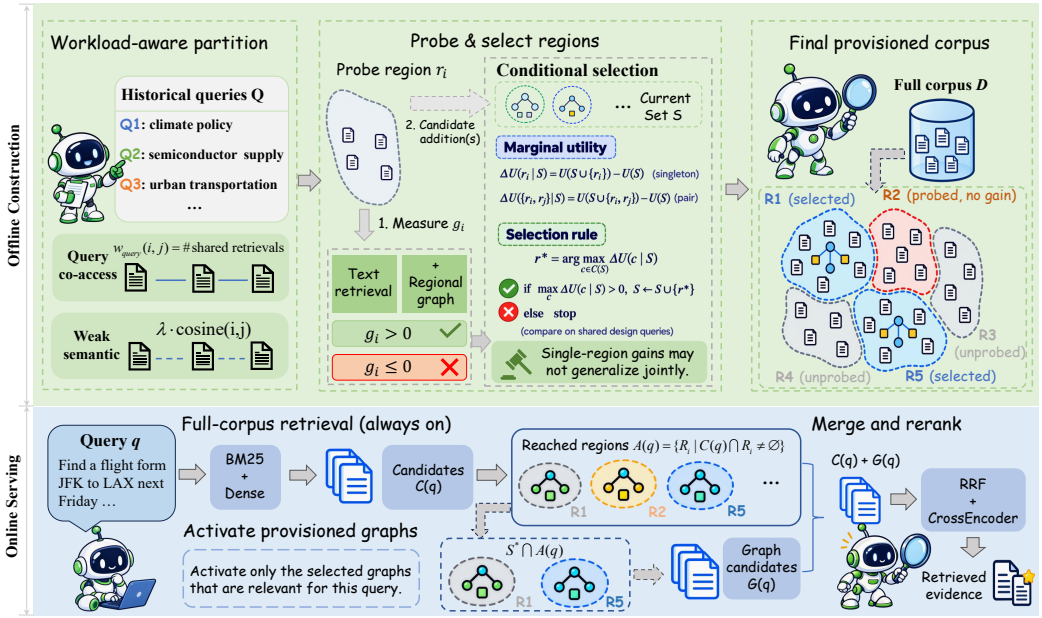


Figure 2: WARP-G design and retrieval. (a) Query co-access and semantic links define regions; workload signals guide probing, and measured gains guide selection. (b) Full-corpus passage retrieval activates selected graphs in reached regions; candidates are fused and reranked. Serving reuses probe indexes, and selecting no graph recovers the baseline.

3.2 Partitioning by Query Co-Access

Let $H_k(q)$ denote the top-k candidates from the BM25–dense baseline after reciprocal-rank fusion (RRF) and before reranking. For distinct passages i and j, define

$$w(i,j) = \sum_{q \in Q_{\text{des}}} \mathbb{I}[i,j \in H_{20}(q)] + \lambda w_{\text{sem}}(i,j), \tag{2}$$

where the first term counts query co-access and $w_{\text{sem}}(i,j)$ is cosine similarity on a symmetrized five-nearest-neighbor graph, with zero weight for absent edges. We use $\lambda = 0.05$.

We apply Leiden clustering and merge communities with fewer than five passages (Traag et al., 2019). The semantic term weakly connects related passages that rarely co-occur in the design workload. Co-access groups passages retrieved for the same questions; semantic edges connect nearby passages that those questions leave sparsely observed. The two signals therefore contribute workload structure and local text similarity, respectively. Partitioning uses neither knowledge extraction nor evidence labels and is fixed before probing. Graph propagation remains regional, while the passage baseline and candidate fusion retain cross-region access.

3.3 Choosing and Measuring Graph Probes

For probe selection and online routing, use the first 20 candidates of a depth-50 baseline run: $L_q^{\text{route}} = H_{50}(q)[1{:}20]$. For each region r_i, define

$$Q_i = \{q \in Q_{\text{des}} : L_q^{\text{route}} \cap r_i \neq \varnothing\}, \qquad F_i = |Q_i|, \qquad a_i = \frac{F_i A_i (1 - R_i)}{T_i}, \tag{3}$$

where A_i is the fraction of Q_i with annotated evidence in r_i, R_i is the mean local-evidence recall among those queries, and T_i is regional text volume. Recall is measured in the routing list and set to zero when no query has local evidence. The product $F_i A_i$ counts reached queries with evidence in the region, while $1 - R_i$ measures their average local evidence gap. Dividing by T_i favors opportunities that require less text to index. This acquisition score directs exploration; paired retrieval then measures actual graph benefit.

We probe approximately 20% of reached regions. Probe allocation interleaves acquisition-ranked choices with seeded random exploration. For each probed region, we build its graph index and sample $n_i = \min(F_i, 64)$ associated design queries $\widehat{Q}_i$. Paired retrieval runs estimate the graph's gain over the baseline:

$$g_i = \frac{1}{n_i} \sum_{q \in \widehat{Q}_i} \left[M(q, \{r_i\}) - M(q, \varnothing) \right], \qquad \widetilde{g}_i = \frac{n_i}{n_i + \alpha} g_i. \tag{4}$$

We set $\alpha = 16$ to downweight estimates from smaller query samples while preserving their sign. Measuring the final ranked list captures both recovered evidence and useful baseline passages displaced by graph candidates. These measured gains determine the selection shortlist. Appendix A.1 gives sampling and implementation details.

3.4 Selecting graphs by conditional gain

Independent probes use different local query samples, and graph candidates overlap and compete for positions in the final ranked list. We therefore shortlist at most six probed graphs by $s_i = F_i \max(\widetilde{g}_i, 0)$ and compare their combinations on a common sample. Zero-score candidates remain eligible for reconsideration.

Starting from $S = \varnothing$, we evaluate candidate additions on a shared sample $Q_c \subseteq Q_{\text{des}}$ of at most 32 queries:

$$U(S) = \frac{1}{|Q_c|} \sum_{q \in Q_c} M(q, S), \qquad \Delta(B \mid S) = U(S \cup B) - U(S), \tag{5}$$

where B is one remaining graph or a graph pair. We shrink conditional gains as $\widetilde{\Delta}(B \mid S) = \frac{|Q_c|}{|Q_c| + \alpha} \Delta(B \mid S)$. All proposals use the same Q_c, so shrinkage preserves their sign and ordering.

At each round, we accept the evaluated proposal with the largest strictly positive conditional gain and update $S \leftarrow S \cup B$. Evaluating additions against the current deployment accounts for both complementary evidence and redundant candidates. Pair evaluation captures complementary graphs, while proposals with non-positive conditional gain are discarded.

We evaluate remaining single-graph additions and at most six pairs per round, for up to three rounds and 16 distinct graph sets. Search stops when no evaluated proposal has positive gain. These limits bound exploration over existing probe indexes. Deployment size follows conditional utility, without an explicit retained-token budget; implementation details appear in Appendix A.2.

3.5 Routing, ranking, and index reuse

For query x, let $C(x) = H_{50}(x)$ be the baseline candidate list and $G_i(x)$ contain up to 50 candidates from graph i. The reached regions are $A(x) = \{r_i : r_i \cap C(x)[1{:}20] \neq \varnothing\}$: each contains at least one of the first 20 baseline candidates.

Only selected graphs in reached regions are invoked. The fused candidates are reranked as follows:

$$Z(x, S) = \text{Rank}_{x,50} \left(\text{RRF}_{50} \left[C(x), \{ G_i(x) : r_i \in S \cap A(x) \} \right] \right). \tag{6}$$

RRF assigns equal weight to each list, scoring passage d by $\sum_{L: d \in L} (60 + \text{rank}_L(d))^{-1}$ following Cormack et al. (2009). Here, $\text{Rank}_{x,50}$ applies the shared cross-encoder reranker to the top 50 fused candidates. The first 10 reranked passages form $L_x(S)$.

Routing uses baseline candidates without evidence labels. Serving reuses selected probe indexes directly. Appendix A.2 provides additional implementation details.

4 Experiments

4.1 Experimental setup

We evaluate three multi-hop QA workloads, HotpotQA, 2WikiMultihopQA (2Wiki), and MuSiQue, together with predominantly single-hop Natural Questions (NQ). Their corpus subsets contain 9,811,

6,119, 11,655, and 9,633 passages. Each uses 800 design queries and 200 held-out test queries with seed 42. Design queries determine partitions, probes, and selected graphs; test queries evaluate the resulting deployment. Appendix B.1 gives subset statistics, and Appendix B.2 reports query-bootstrap intervals.

Compared systems. BM25, Dense, and their reciprocal-rank fusion (Hybrid) are passage baselines. HippoRAG2 retrieves from a full-corpus graph; Full Graph additionally fuses Hybrid candidates. WARP-G uses HippoRAG2 for regional indexes. Our KET-RAG and G2ConS adaptations apply PageRank- and concept-based selection to the same backend, targeting 80% of passages. All systems share the final reranker. Full Graph has global connectivity, while WARP-G combines regional propagation with full-corpus passage access.

Metrics and accounting. We report evidence recall (ER@k), complete evidence recovery (CE@k), answer exact match (EM), and token F1. The model answers each question using the top five reranked passages. Graph-token accounting separates design, retained-index construction (deployment), and online retrieval. First use includes design or construction plus logical online tokens, counting reused indexes once. Shared passage indexing, reranking, and answer generation are excluded. Table 1 reports quality; Appendix B.2 provides additional retrieval depths.

4.2 Retrieval and answer quality

WARP-G matches or improves Hybrid's answer F1 across all four workloads while retaining at most two regional graphs (Table 1). The passage retriever still searches the entire corpus; regional selection determines where graph retrieval adds candidates. Small graph coverage therefore retains access to every passage.

HotpotQA: high baseline coverage, no retained graph. Hybrid reaches 97.75 ER@10 and 95.50 CE@10, recovering complete top-10 evidence for 191 of 200 test questions. All four design probes have non-positive mean gains, and WARP-G selects no graph. It therefore preserves Hybrid's 74.53 F1 and eliminates online graph retrieval. The high baseline coverage and non-positive probe gains support passage-only serving for this multi-hop workload.

2Wiki: one regional graph improves ranking and answers. WARP-G retains one 570-passage region costing 8.2% of Full Graph construction tokens. ER@10 reaches 80.00, the highest score among the compared systems, and F1 reaches 64.21, improving over Hybrid and Full Graph by 1.15 and 1.74 percentage points (pp). The evidence used to answer questions also improves: ER@5 rises from 73.50 to 74.38 and CE@5 from 43.50 to 45.00 relative to Hybrid. CE@10 remains 53.00. The top five passages contain complete evidence for 90 questions, up from 87 with Hybrid. More questions therefore have all annotated evidence available for answer generation.

MuSiQue: higher answer F1 with mixed retrieval gains. Hybrid starts at 73.51 ER@10 and 46.00 CE@10. WARP-G retains two regions costing 14.0% of Full Graph construction tokens, raising ER@5 from 65.92 to 66.38 and F1 from 43.07 to 44.17. CE@10 falls from 46.00 to 45.50, while BM25 attains the highest F1 (45.79) despite lower recall. These results separate two effects: recovering more annotated evidence and providing evidence that helps the model answer correctly.

NQ: conditional selection filters localized gains. Several regional probes improve their local design samples, but no evaluated addition improves utility on the shared selection sample. WARP-G consequently retains no graph and matches Hybrid at 87.06 ER@10 and 60.39 F1. The two passage-only deployments follow different design outcomes: non-positive local gains on HotpotQA and no positive conditional gain on NQ.

4.3 Design, deployment, and online cost

WARP-G reduces graph cost at both deployment and first use (Figure 3). Retained indexes cost 0%, 8.2%, 14.0%, and 0% of Full Graph construction on HotpotQA, 2Wiki, MuSiQue, and NQ. By comparison, the content-selected baselines consume 84.4–93.2% across these workloads (Ta-

Table 1: Retrieval and answer quality (%). Bold and underlined values mark the best and next distinct score within each dataset, including ties at the displayed precision. Rules separate method groups.

Dataset	Type	Method	ER@5	CE@5	ER@10	CE@10	EM	F1
HotpotQA	Passage	BM25	93.75	87.50	96.75	93.50	63.00	74.24
		Dense	93.75	88.00	97.50	95.00	63.00	73.85
		Hybrid	**94.50**	**89.50**	**97.75**	**95.50**	**63.50**	**74.53**
	Full-corpus	HippoRAG2	93.50	87.50	97.25	94.50	62.50	73.32
		Full Graph	93.75	88.00	**97.75**	**95.50**	63.00	73.53
	Content-selected	KET-RAG	94.25	89.00	97.50	95.00	63.00	74.03
		G2ConS	94.25	89.00	97.25	94.50	**63.50**	**74.53**
	Regional	**WARP-G**	**94.50**	**89.50**	**97.75**	**95.50**	**63.50**	**74.53**
2Wiki	Passage	BM25	73.00	43.00	78.88	52.00	57.50	60.69
		Dense	73.38	43.50	79.38	**53.50**	59.50	62.54
		Hybrid	73.50	43.50	79.25	53.00	60.00	63.06
	Full-corpus	HippoRAG2	73.38	43.50	78.25	51.00	**61.00**	63.79
		Full Graph	73.13	43.00	79.00	52.50	59.50	62.47
	Content-selected	KET-RAG	72.88	42.50	78.00	51.00	60.00	63.20
		G2ConS	73.00	42.50	78.25	51.50	59.00	62.13
	Regional	**WARP-G**	**74.38**	**45.00**	**80.00**	53.00	**61.00**	**64.21**
MuSiQue	Passage	BM25	63.58	30.50	70.25	42.00	**33.50**	**45.79**
		Dense	65.88	33.50	73.25	**46.00**	33.00	44.28
		Hybrid	65.92	33.00	73.51	**46.00**	32.00	43.07
	Full-corpus	HippoRAG2	65.75	33.50	72.96	45.50	32.50	44.00
		Full Graph	66.04	33.50	72.96	44.50	33.00	43.28
	Content-selected	KET-RAG	**66.42**	**34.00**	73.33	45.00	33.00	43.80
		G2ConS	66.17	33.50	**73.96**	**46.00**	32.50	43.57
	Regional	**WARP-G**	66.38	**34.00**	73.62	45.50	32.00	44.17
NQ	Passage	BM25	68.47	39.00	87.41	**70.50**	47.50	60.19
		Dense	**68.95**	**39.50**	87.53	69.50	**48.00**	**60.80**
		Hybrid	**68.95**	**39.50**	87.06	69.50	**48.00**	60.39
	Full-corpus	HippoRAG2	**68.95**	**39.50**	87.62	69.50	**48.00**	**60.80**
		Full Graph	**68.95**	**39.50**	**87.81**	**70.50**	**48.00**	60.62
	Content-selected	KET-RAG	**68.95**	**39.50**	86.99	69.50	**48.00**	60.19
		G2ConS	68.45	39.00	87.22	69.50	**48.00**	60.62
	Regional	**WARP-G**	**68.95**	**39.50**	87.06	69.50	**48.00**	60.39

ble 7). Workload-guided selection retains substantially smaller indexes than these content-selection baselines.

Design, including probe construction and conditional evaluation, uses 40.4%, 62.4%, 40.5%, and 35.3% of Full Graph construction tokens. Selected indexes are reused, so their construction is already covered by design. Including this upfront work and online retrieval, first-use graph tokens fall by 62.3%, 38.7%, 57.5%, and 67.2%, respectively. For example, 2Wiki uses 6.713 million first-use tokens versus 10.943 million for Full Graph (Table 6). This total includes all probed graphs and selection runs, including graphs discarded before serving. HotpotQA and NQ spend 6.849 and 6.066 million tokens on design, then serve all test queries without graph calls.

Design overhead explains the differences in first-use savings. 2Wiki has the largest design-to-full-build ratio and therefore the smallest first-use saving, despite retaining only one graph. Online

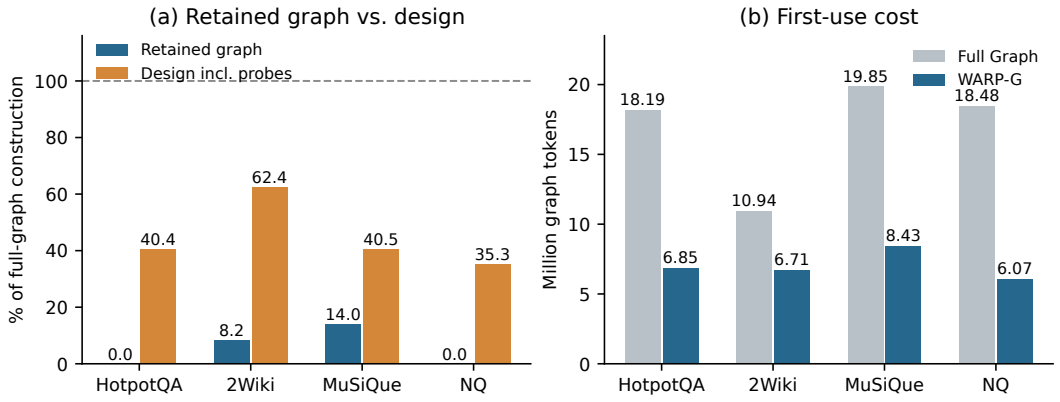


Figure 3: Graph costs across the four workloads. (a) Retained-index construction and total design tokens, normalized to Full Graph construction; retained indexes are part of design. (b) First-use tokens include design or construction and logical online retrieval for 200 test queries, counting reused indexes once.

retrieval uses 213 backend calls on 2Wiki and 291 on MuSiQue, versus 400 for Full Graph, and none on HotpotQA or NQ. Figure 4 places these savings alongside answer F1. Reusing a deployment amortizes design over subsequent queries; Appendix B.3 gives the full accounting.

4.4 Ablation Study

Table 2: Ablation results (%). Parentheses show changes from WARP-G in percentage points; deployment cost is in millions of tokens.

Dataset	Variant	ER@10	CE@10	Deploy (M)
2Wiki	**WARP-G**	80.00	53.00	0.798
	Frequency only	79.13 (-0.87)	52.50 (-0.50)	1.946
	Gain only	78.88 (-1.12)	52.50 (-0.50)	0.849
	w/o semantic edges	79.25 (-0.75)	53.00 (+0.00)	0.000
	w/o query co-access	79.25 (-0.75)	53.00 (+0.00)	0.000
MuSiQue	**WARP-G**	73.62	45.50	2.598
	Frequency only	73.58 (-0.04)	46.50 (+1.00)	2.925
	Gain only	74.54 (+0.92)	48.50 (+3.00)	2.087
	w/o semantic edges	73.50 (-0.12)	45.50 (+0.00)	1.444
	w/o query co-access	73.71 (+0.09)	46.00 (+0.50)	0.971

Combining demand and measured gain. Table 2 compares selection rules at WARP-G's retained-region count. On 2Wiki, WARP-G improves ER@10 over frequency-only and gain-only selection by 0.87 and 1.12 pp, respectively, using fewer deployment tokens than either rule. Combining demand and conditional utility improves the quality–cost balance on 2Wiki. Both rules retain one region, yet construction costs 0.798 million tokens for WARP-G and 1.946 million for frequency-only selection.

On MuSiQue, gain-only selection achieves 74.54 ER@10 and 48.50 CE@10, versus 73.62 and 45.50 for WARP-G, with lower deployment cost. Selection-rule performance differs across workloads: conditional selection leads on 2Wiki, and local-gain ranking leads on MuSiQue. Appendix B.4 reports additional metrics, costs, and the paired comparison.

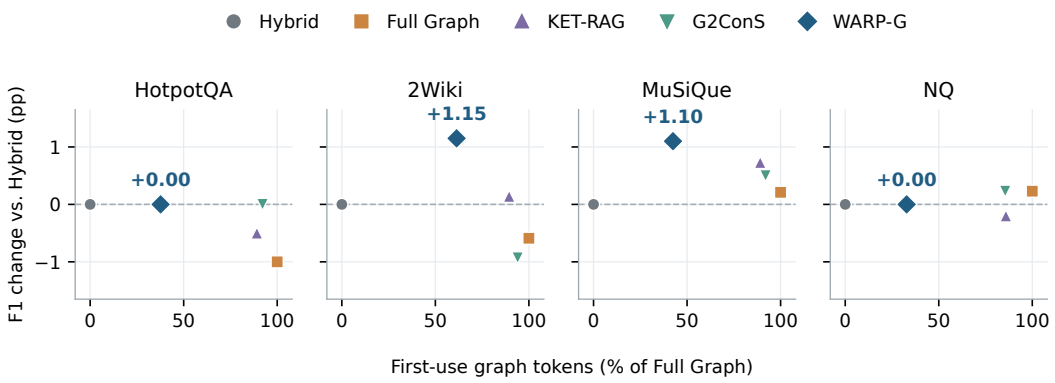


Figure 4: Answer F1 versus first-use graph tokens, normalized to Full Graph. WARP-G matches or improves Hybrid's F1 with less than 62% of Full Graph's first-use tokens on every dataset.

Partitioning shapes quality and online work. On 2Wiki, removing semantic edges or query co-access yields no retained graph and returns Hybrid's scores. Random partitioning reaches 80.13 ER@10 and 54.00 CE@10 at lower deployment cost. On MuSiQue, random partitioning also reduces deployment cost but requires 651 backend calls, compared with 291 for WARP-G and 400 for Full Graph. A query invokes each selected region it reaches, so several small graphs can generate more calls than one large graph. Partitioning must therefore be assessed through both retained-index cost and online work.

4.5 Dataset differences and deployment choices

The 2Wiki and MuSiQue results distinguish reaching an evidence region from placing its passages among the five used to answer the question.

Baseline routing reaches every gold-evidence region for 94.0% of 2Wiki queries and 87.5% of MuSiQue queries (Table 4). Thus, 12 versus 25 test queries miss at least one such region, reducing opportunities for regional augmentation on MuSiQue. Full-corpus passage retrieval remains available in both cases.

Selection rewards top-10 utility, while answer generation uses the top five passages. Evidence ranked sixth to tenth contributes to the selection score but is not used to answer the question. We therefore report retrieval at both depths alongside answer F1.

Frequency-only provisioning removes gain measurement from deployment: rank regions by query access and build the highest-ranked regions under a chosen capacity. At the matched region counts, frequency-only selection is within 0.87 pp of WARP-G in ER@10 on 2Wiki and within 0.04 pp on MuSiQue. It retains larger graphs on both datasets, but skipping gain probing and conditional evaluation lowers first-use tokens from 6.713 to 2.786 million on 2Wiki and from 8.429 to 3.826 million on MuSiQue (Table 9). The two choices address different deployment needs. Frequency-only provisioning fills a given capacity using access logs. WARP-G spends design effort to decide which graph additions improve retrieval and how many graphs to retain, including selecting none.

5 Conclusion

WARP-G allocates graph construction according to query demand and retrieval benefit. Limited regional probes and conditional selection produce reusable indexes costing 0–14.0% of Full Graph construction tokens. Across four QA workloads, WARP-G reduces first-use graph tokens by 38.7–67.2% and matches or improves Hybrid's answer F1. The dataset differences identify the roles of baseline evidence coverage, regional access, and evidence ranking in graph deployment. Frequency-only provisioning provides a simpler deployment path: at a chosen capacity, it retains competitive evidence recall without gain measurement or conditional evaluation. Workload-aware regional provisioning therefore supports both benefit-driven index selection and lightweight deployment based on query access alone.

AI USE STATEMENT

Generative AI tools were used to improve manuscript clarity and presentation, provide feedback on experimental design, and assist with code development and debugging. All methodological choices, experimental protocols, implementations, analyses, and reported results were reviewed and verified by the authors. The authors take full responsibility for the final manuscript and associated artifacts.

ETHICS STATEMENT

This work uses established question-answering benchmarks and does not involve new human-subject data collection. As with other retrieval systems, deployed implementations may reproduce biases, private information, or factual errors present in the source corpus. In addition, WARP-G allocates graph capacity using a historical design workload, so query types that are rare or absent from that workload may receive less graph support under distribution shift. Selective materialization improves efficiency but does not eliminate these risks. Deployments should therefore maintain appropriate corpus governance, access control, workload monitoring, and output auditing.

REPRODUCIBILITY STATEMENT

Section 4 describes the datasets, splits, baselines, metrics, and evaluation protocol. Appendix A provides the partitioning, probing, conditional-selection, routing, backend configuration, and cost-accounting details. The accompanying code and configurations implement these procedures with fixed random seeds and the settings used in the reported experiments.

REFERENCES

Sanjay Agrawal, Surajit Chaudhuri, and Vivek R Narasayya. Automated selection of materialized views and indexes in sql databases. In *VLDB*, volume 2000, pp. 496–505, 2000.

Kamel Aouiche and Jérôme Darmont. Index and materialized view selection in data warehouses. *arXiv preprint arXiv:1701.08029*, 2017.

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*, volume 2024, pp. 9112–9141, 2024.

Surajit Chaudhuri and Vivek R Narasayya. An efficient, cost-driven index selection tool for microsoft sql server. In *VLDB*, volume 97, pp. 146–155. San Francisco, 1997.

Wenqi Chen, Haofei Yang, Rui Yang, and Fangming Li. Vdgr-rag: Vectors, directories, graphs, and reflection are all you need for unified reasoning over hierarchical enterprise knowledge. *arXiv preprint arXiv:2608.07994*, 2026.

Martino Ciaperoni, Cigdem Aslay, Aristides Gionis, and Michael Mathioudakis. Workload-aware materialization of junction trees. *arXiv preprint arXiv:2110.03475*, 2021.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 758–759, 2009.

Su Dong, Qinggang Zhang, Yilin Xiao, Shengyuan Chen, Chuang Zhou, and Xiao Huang. Use graph when it needs: Efficiently and adaptively integrating retrieval-augmented generation with graphs. *arXiv preprint arXiv:2602.03578*, 2026.

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

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

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

Bernal J Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569, 2024.

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

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

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*, 2024.

Yue Han, Chengliang Chai, Jiabin Liu, Guoliang Li, Chuangxian Wei, and Chaoqun Zhan. Dynamic materialized view management using graph neural network. In *39th IEEE International Conference on Data Engineering, ICDE*, pp. 3–7, 2023.

Venky Harinarayan, Anand Rajaraman, and Jeffrey D Ullman. Implementing data cubes efficiently. *Acm Sigmod Record*, 25(2):205–216, 1996.

Jakir Hossain and Ahmet Erdem Sarıyüce. Core-based hierarchies for efficient graphrag. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 1722–1733, 2026.

Xianming Hu, Jingyang Chen, Bin Tang, Yihe Liu, Yihong Huang, Hongbo Zhao, Nuoyi Chen, Jie Zhang, Ping Li, and Kai Zhang. Zoomrag: Hierarchical random-walk zooming across multi-scale information graphs for fast and accurate rag. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 32840–32859, 2026.

Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. Grag: Graph retrieval-augmented generation. In *Findings of the association for computational linguistics: NAACL 2025*, pp. 4145–4157, 2025.

Yiqian Huang, Shiqi Zhang, and Xiaokui Xiao. Ket-rag: A cost-efficient multi-granular indexing framework for graph-rag. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 1003–1012, 2025.

Glen Jeh and Jennifer Widom. Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*, pp. 271–279, 2003.

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 the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)*, 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 the 2023 conference on empirical methods in natural language processing*, pp. 7969–7992, 2023.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, 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. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Ziyu Liu, Yijing Liu, Jianfei Yuan, Minzhi Yan, Le Yue, Honghui Xiong, and Yi Yang. Graph-guided concept selection for efficient retrieval-augmented generation. *arXiv preprint arXiv:2510.24120*, 2025.

Congmin Min, Rhea Mathew, Joyce Pan, Sahil Bansal, Abbas Keshavarzi, and Amar Viswanathan Kannan. Efficient knowledge graph construction and retrieval from unstructured text for large-scale rag systems. *arXiv e-prints*, pp. arXiv–2507, 2025.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, 1999.

Yue Pang, Lei Zou, Jeffrey Xu Yu, and Linglin Yang. Materialized view selection & view-based query planning for regular path queries. *Proceedings of the ACM on Management of Data*, 2(3): 1–26, 2024.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. Graph retrieval-augmented generation: A survey. *ACM Transactions on Information Systems*, 44(2):1–52, 2025.

Amitabh Priyadarshi and Krzysztof J Kochut. Wawpart: workload-aware partitioning of knowledge graphs. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 383–395. Springer, 2021.

Amitabh Priyadarshi and Krzysztof J Kochut. Awapart: adaptive workload-aware partitioning of knowledge graphs. *arXiv preprint arXiv:2203.14884*, 2022.

Pin Qian, Su Wang, Chong Peng, Junxian You, Lifei Liu, Haoran Yu, Yihang Chen, and Xiaochong Jiang. When should active rag retrieve? a budget-aware evaluation of utility, calibration, and cost. *arXiv preprint arXiv:2607.24010*, 2026.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and trends® in information retrieval*, 4(1-2):1–174, 2009.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, volume 2024, pp. 32628–32649, 2024.

Yufan Sheng, Xin Cao, Yixiang Fang, Kaiqi Zhao, Jianzhong Qi, Gao Cong, and Wenjie Zhang. Wisk: a workload-aware learned index for spatial keyword queries. *Proceedings of the ACM on Management of Data*, 1(2):1–27, 2023.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 4231–4242, 2018.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 2380–2390, 2019.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019.

Pengyu Wang, Benfeng Xu, Shaohan Wang, Xin Zeng, Huarui Wu, Lei Zhang, and Licheng Zhang. Bm25 wins at scale: A scaling study of retrieval-augmented generation paradigms. *arXiv e-prints*, pp. arXiv–2607, 2026a.

Ziqi Wang, Xi Zhu, Shuhang Lin, Haochen Xue, Minghao Guo, and Yongfeng Zhang. Ragrouter-bench: A dataset and benchmark for adaptive rag routing. *arXiv preprint arXiv:2602.00296*, 2026b.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, et al. Answering complex open-domain questions with multi-hop dense retrieval. *arXiv preprint arXiv:2009.12756*, 2020.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Rui Yu, Tianyi Wang, Ruixia Liu, and Yinglong Wang. Sf-rag: Structure-fidelity retrieval-augmented generation for academic question answering. *arXiv preprint arXiv:2602.13647*, 2026.

Yibo Zhao, Jiapeng Zhu, Ye Guo, Kangkang He, and Xiang Li. Eˆ 2graphrag: Streamlining graph-based rag for high efficiency and effectiveness. *arXiv preprint arXiv:2505.24226*, 2025.

Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. Linearrag: Linear graph retrieval augmented generation on large-scale corpora. In *International Conference on Learning Representations*, volume 2026, pp. 147053–147075, 2026.

A Implementation and Evaluation Details

A.1 Partitioning, features, and probing

Leiden uses `RBConfigurationVertexPartition`, resolution 1.0, and seed 42. Communities with fewer than five passages are merged into the larger community with the greatest connecting edge weight; disconnected small communities use the lowest-indexed nonempty alternative. Symmetric semantic edges retain the maximum similarity between the two directions. All passages remain assigned to exactly one region.

Partitioning uses $H_{20}(q)$ from a top-20 baseline retrieval run, whereas probing and routing use the first 20 passages of a separate $H_{50}(q)$ run. For either depth k, BM25 and Dense each retrieve $2k$ candidates before RRF. Because the component retrievers are rerun at different depths and fused separately, $H_{20}(q)$ need not equal $H_{50}(q)[1{:}20]$. We therefore keep the two definitions distinct throughout the implementation.

Probe acquisition uses the quantities in Equation (3); other recorded regional features are used only for diagnostics and are not used to predict graph gains. Text volume counts contiguous Unicode word characters after lowercasing. Local recall is averaged over queries containing annotated evidence in the region; a mean with no valid observations is set to zero. Priority ties are broken by text volume and then region identifier.

For n_+ reached regions, the probe target is

$$\min\{n_+, \max(1, \mathrm{round}(0.2n_+))\}.$$

Most probes follow acquisition-score order. To provide limited exploration, one choice in every five, starting at position 2 (2, 7, 12, and so on), follows a seeded random ordering of region identifiers. HotpotQA, 2Wiki, MuSiQue, and NQ therefore probe four, three, six, and six regions, respectively.

Associated query identifiers are sorted and, when necessary, sampled without replacement to at most 64 queries using a region-specific seed. Independent graph gains use the shrinkage pseudocount $\alpha = 16$. The shrinkage factor regularizes small local samples but is not a confidence interval or statistical significance test.

Unprobed regional gains remain unknown. We do not fit a gain predictor: each dataset provides only three to six probes. WARP-G selects only graphs whose gains have been measured directly.

A.2 Conditional selection and online retrieval

Conditional selection uses at most six probed graphs. Shortlist-score ties are broken by region identifier, and zero-score candidates remain eligible for evaluation on the shared query sample. At each round, all remaining single-graph additions are evaluated before graph pairs. Pairs are enumerated in shortlist order and truncated to the first six. If positive proposal scores tie, the first proposal is retained.

The implementation evaluates at most three rounds and caches at most 16 distinct graph sets, including the empty set. Each set is identified by its sorted region identifiers. Once the cache is full, previously unseen configurations are skipped and only cached sets are reused. This restricted search controls design cost but is not guaranteed to recover the globally optimal graph set. It also imposes no explicit deployment-token budget.

Only probed graphs enter WARP-G's conditional selection, and selected probe indexes are reused without reconstruction. By contrast, the random, frequency-only, and gain-only controls may select previously unprobed regions and therefore build additional indexes. Gain-only assigns unknown gains a score of zero, so an unprobed region may rank above a region with a measured negative gain. These controls match WARP-G's number of retained regions rather than its construction-token budget.

Regional indexes use the official HippoRAG2 backend. RRF gives equal weight to all candidate lists and breaks score ties by passage identifier. Graph propagation remains within each regional index, while the shared passage baseline always searches the full corpus. A selected graph is invoked only when the first 20 passages of the baseline H_{50} run reach its region.

Table 3 summarizes the principal backend and design settings.

Table 3: Backend and design settings used in the reported experiments.

Component	Configuration
Co-access depth / semantic neighbors / weight	20 / 5 / .05
Leiden resolution / minimum community	1.0 / 5 passages
Probe fraction / per-region query cap / seed	.2 / 64 / 42
Gain shrinkage pseudo-count	16
Conditional candidates / pairs / rounds / sets	6 / 6 / 3 / 16
Shared conditional query sample	At most 32 design queries
Route / fusion / output depth	20 / 50 / 10
Baseline and graph embedding model	`nvidia/NV-Embed-v2`
Graph extraction and answer generation	`deepseek-v4-flash`, thinking disabled
Graph retrieval depth / linking depth	200 / 5
Propagation damping / passage-node weight	.5 / .05
Synonym neighbors / similarity threshold	2,047 / .8
Reranker	`BAAI/bge-reranker-v2-m3`
Reranking batch / maximum length	32 / 512

A.3 Evaluation protocol and cost accounting

The reported experiments use seed 42 with 800 design and 200 held-out test queries per dataset. Only design queries affect region construction, probe allocation, graph-gain measurement, and conditional selection. Test evidence labels are used only for final evaluation.

Retrieval metrics are ER and CE at depths 2, 3, 5, and 10. A fixed model answers each question using the cached top five passages; answers are evaluated by EM and token F1. BM25, Dense, and Hybrid use the same retrieval depths and final reranker.

Full Graph constructs a single graph over the entire corpus and retains fact and synonym edges that may cross WARP-G region boundaries. It is therefore an end-to-end full-corpus GraphRAG baseline rather than the union of all independent regional graphs. Comparisons with Full Graph consequently reflect both selective deployment and the change from global to regional graph topology.

Partition ablations use query-co-access edges only, semantic edges only, or random passage assignment. Because changing the partition also changes associated query sets, probes, selected graphs, and routing coverage, these ablations measure the resulting end-to-end design rather than the isolated effect of one edge type.

Routing diagnostics report the fraction of questions for which baseline routing reaches at least one or all gold-evidence regions. Reaching a region does not imply that a graph is selected there or that its evidence appears in the final reranked list.

We distinguish three graph-cost categories:

- **Design:** graph tokens used to construct probes and evaluate gains during probing and conditional selection.
- **Deployment:** construction tokens associated with the graph indexes retained for serving.
- **Online:** graph-retrieval tokens consumed by test queries; calls and wall time are reported separately.

For WARP-G, first-use graph tokens equal design plus logical online tokens. Deployment is a subset of design because selected probe indexes are reused and is therefore not added again. For Full Graph, first use equals construction plus logical online tokens. Shared passage indexing, reranking compute, and answer generation are excluded from this graph-token accounting. Logical tokens measure requested work before cache reuse, whereas physical tokens record actual usage after cache reuse. Regional text volume is used only for probe prioritization and is not itself a billed token count.

B Additional Experiments and Results

B.1 Dataset and design protocol

Table 4 summarizes the four corpus subsets and the fixed design/test protocol. WARP-G retains zero, one, two, and zero regions on HotpotQA, 2Wiki, MuSiQue, and NQ, respectively. Only design queries determine partitions, probes, and graph selection.

Table 4: Corpus and design statistics. Routing coverage is the percentage of test questions reaching any or all gold-evidence regions.

Dataset	Passages	Design/test	Regions/probes	Any	All
HotpotQA	9,811	800/200	22/4	100.0	99.5
2Wiki	6,119	800/200	14/3	100.0	94.0
MuSiQue	11,655	800/200	29/6	100.0	87.5
NQ	9,633	800/200	33/6	99.5	99.5

The conditional-selection objective equally weights ER@10 and CE@10. Independent probe means are shrunk toward zero using a pseudo-count of 16. Conditional selection evaluates at most six measured graphs on a shared sample of at most 32 design questions, for at most three rounds and 16 distinct graph-set evaluations. Search stops when no evaluated addition has positive marginal utility. These limits bound design effort without imposing a deployment-token budget.

Each dataset has one completed design/test evaluation under seed 42. The reported experiments therefore quantify uncertainty over held-out test questions but do not measure variation across alternative design workloads or random seeds.

B.2 Additional quality metrics and uncertainty

Table 5 reports shallower retrieval results. Query-bootstrap 95% intervals for WARP-G ER@10 are [96.25, 99.00], [76.88, 83.13], [70.00, 77.29], and [83.47, 90.36] on HotpotQA, 2Wiki, MuSiQue, and NQ, respectively. Corresponding CE@10 intervals are [92.50, 98.00], [46.00, 60.00], [38.50, 52.50], and [63.00, 76.00]. These intervals describe uncertainty from the held-out test sample; they do not measure variation in the design procedure, and interval overlap is not a substitute for paired testing.

We use the saved tests and their original multiple-comparison groups; rounded table values are not used to recompute significance. Answer F1 and partition-ablation differences are reported as point estimates.

B.3 Design, deployment, and online costs

Table 6 reports first-use graph-token accounting, counting reused indexes once and including logical online retrieval for 200 test queries. WARP-G uses 6,848,991; 6,713,373; 8,429,116; and 6,065,845 tokens on HotpotQA, 2Wiki, MuSiQue, and NQ, respectively. Full Graph uses 18,186,102; 10,942,996; 19,845,492; and 18,476,372 tokens.

BM25, Dense, and Hybrid incur no graph tokens under this accounting; shared passage indexing and reranking are treated as common infrastructure.

Table 7 reports deployment construction tokens at one-token precision, whereas Table 6 also includes design cost. KET-RAG and G2ConS consume 84.4–93.2% of Full Graph construction tokens despite targeting 80% of passages, showing that passage coverage is only a rough proxy for graph-construction cost.

Table 8 compares logical retrieval tokens with physical usage after cache reuse. WARP-G's 213 calls on 2Wiki contain 114 dense fallbacks and 99 graph-seeded calls; its 291 calls on MuSiQue contain 162 fallbacks and 129 graph-seeded calls. Invoking the graph backend therefore does not necessarily produce graph-seeded retrieval. Calls may also exceed the number of test queries because one

Table 5: Retrieval at smaller cutoffs (percent).

Dataset	Method	ER@2	CE@2	ER@3	CE@3
HotpotQA	BM25	84.75	71.50	89.75	80.50
	Dense	84.50	71.00	89.50	80.00
	Hybrid	84.50	71.00	89.75	80.50
	HippoRAG2	84.75	71.50	89.75	80.50
	Full Graph	84.50	71.00	90.00	81.00
	KET-RAG	84.50	71.00	89.75	80.50
	G2ConS	84.50	71.00	89.75	80.50
	WARP-G	84.50	71.00	89.75	80.50
2Wiki	BM25	67.38	35.00	69.88	37.00
	Dense	67.63	35.50	70.50	38.50
	Hybrid	67.38	35.00	69.75	37.00
	HippoRAG2	67.63	35.50	70.50	38.50
	Full Graph	67.38	35.00	69.75	37.00
	KET-RAG	67.38	35.00	69.50	36.50
	G2ConS	67.38	35.00	69.75	37.00
	WARP-G	67.63	35.50	70.13	37.50
MuSiQue	BM25	49.17	10.50	55.79	20.00
	Dense	49.83	10.00	57.29	20.00
	Hybrid	50.25	10.50	57.92	21.50
	HippoRAG2	49.83	10.00	57.54	20.50
	Full Graph	50.00	10.00	57.13	20.00
	KET-RAG	50.25	10.50	57.92	21.50
	G2ConS	50.25	10.50	57.92	21.50
	WARP-G	50.17	10.00	57.38	20.50
NQ	BM25	42.55	19.00	54.59	28.00
	Dense	42.38	19.00	55.02	28.50
	Hybrid	42.55	19.00	55.19	28.50
	HippoRAG2	42.38	19.00	55.02	28.50
	Full Graph	42.38	19.00	55.06	28.50
	KET-RAG	43.05	19.50	55.06	28.50
	G2ConS	42.38	19.00	54.56	28.00
	WARP-G	42.55	19.00	55.19	28.50

Table 6: Design-aware graph cost (millions of tokens). WARP-G first use equals design plus logical online usage; deployed probe graphs are reused.

Dataset	Full build	Retained	Design	Online	First use	Full first use
HotpotQA	16.961	0.000	6.849	0.000	6.849	18.186
2Wiki	9.717	0.798	6.068	0.646	6.713	10.943
MuSiQue	18.618	2.598	7.538	0.891	8.429	19.845
NQ	17.168	0.000	6.066	0.000	6.066	18.476

Table 7: Graph construction and retained-index cost. Tokens are in millions with six decimals.

Dataset	Method	Input	Output	Embed.	Total
HotpotQA	Full / Hippo	9.245766	3.578159	4.137110	16.961035
	KET-RAG	7.586526	2.991866	4.406124	14.984516
	G2ConS	7.767126	3.161326	4.609327	15.537779
	WARP-G	0.000000	0.000000	0.000000	0.000000
2Wiki	Full / Hippo	5.476080	1.968852	2.272184	9.717116
	KET-RAG	4.511241	1.663475	2.391903	8.566619
	G2ConS	4.648706	1.829987	2.582360	9.061053
	WARP-G	0.486575	0.145780	0.165215	0.797570
MuSiQue	Full / Hippo	10.693617	3.706059	4.218442	18.618118
	KET-RAG	8.786972	3.087789	4.504035	16.378796
	G2ConS	8.970624	3.261045	4.719589	16.951258
	WARP-G	1.539723	0.493851	0.564517	2.598091
NQ	Full / Hippo	9.648302	3.525687	3.994013	17.168002
	KET-RAG	7.553825	2.818851	4.203380	14.576056
	G2ConS	7.542107	2.786224	4.169796	14.498127
	WARP-G	0.000000	0.000000	0.000000	0.000000

query can reach multiple selected regions or trigger repeated backend calls. Reported wall times are observational measurements rather than controlled latency comparisons.

Table 8: Online graph retrieval on 200 test questions. Logical and physical counts sum LLM input and output tokens.

Dataset	Method	Calls	Logical	Physical	Time (s)
HotpotQA	Full Graph	400	1,225,067	1,225,067	2,511.6
	HippoRAG2	400	1,225,086	6,223	885.2
	WARP-G	0	0	0	0.0
2Wiki	Full Graph	400	1,225,880	1,225,880	3,293.3
	HippoRAG2	400	1,226,076	53,474	2,505.7
	KET-RAG	400	1,221,040	1,221,040	9,336.6
	G2ConS	400	1,224,567	427,728	2,172.1
	WARP-G	213	645,869	545,104	889.8
MuSiQue	Full Graph	400	1,227,374	61,245	2,489.4
	HippoRAG2	400	1,227,325	3,103	2,000.7
	WARP-G	291	891,369	516,684	1,149.2
NQ	Full Graph	400	1,308,370	55,591	784.1
	HippoRAG2	400	1,308,355	0	607.4
	WARP-G	0	0	0	0.0

B.4 Selection and partition details

Table 9 reports the costs of frequency-only and gain-only controls; Table 10 includes random selection in the quality comparison. Each retains the same number of regions as WARP-G. Frequency-only selection records zero design tokens because it excludes the cost of determining that retained-region count; the same convention applies to the random control. Gain-only includes probe cost and may additionally construct previously unprobed regions. Unknown gains receive a score of zero and may therefore rank above measured negative gains. These controls compare alternative ranking

Table 9: Selection-control costs in millions of tokens. First use counts reused indexes once; controls inherit WARP-G's retained-region count.

Dataset	Variant	Deployment	Design	First use
HotpotQA	WARP-G	0.000	6.849	6.849
	Frequency only	0.000	0.000	0.000
	Gain only	0.000	6.108	6.108
2Wiki	WARP-G	0.798	6.068	6.713
	Frequency only	1.946	0.000	2.786
	Gain only	0.849	4.606	5.927
MuSiQue	WARP-G	2.598	7.538	8.429
	Frequency only	2.925	0.000	3.826
	Gain only	2.087	6.730	9.037
NQ	WARP-G	0.000	6.066	6.066
	Frequency only	0.000	0.000	0.000
	Gain only	0.000	5.604	5.604

Table 10: Selection rules at smaller retrieval cutoffs (percent). Each rule selects the same number of regions as WARP-G; token costs can differ.

Dataset	Variant	ER@2	CE@2	ER@3	CE@3	ER@5	CE@5
HotpotQA	WARP-G	84.50	71.00	89.75	80.50	94.50	89.50
	Random selection	84.50	71.00	89.75	80.50	94.50	89.50
	Frequency only	84.50	71.00	89.75	80.50	94.50	89.50
	Gain only	84.50	71.00	89.75	80.50	94.50	89.50
2Wiki	WARP-G	67.63	35.50	70.13	37.50	74.38	45.00
	Random selection	67.38	35.00	69.75	37.00	73.50	43.50
	Frequency only	67.63	35.50	70.25	38.00	73.38	43.50
	Gain only	67.38	35.00	69.75	37.00	73.50	43.50
MuSiQue	WARP-G	50.17	10.00	57.38	20.50	66.38	34.00
	Random selection	50.75	11.50	58.46	22.50	66.46	33.50
	Frequency only	49.75	10.00	58.54	22.00	66.08	33.00
	Gain only	50.33	11.50	58.96	23.50	66.79	34.50
NQ	WARP-G	42.55	19.00	55.19	28.50	68.95	39.50
	Random selection	42.55	19.00	55.19	28.50	68.95	39.50
	Frequency only	42.55	19.00	55.19	28.50	68.95	39.50
	Gain only	42.55	19.00	55.19	28.50	68.95	39.50

Table 11: Partition ablations. Deployment and design costs are in millions of tokens.

Dataset	Variant	ER@10	CE@10	Deploy (M)	Design (M)
HotpotQA	WARP-G	97.75	95.50	0.000	6.849
	w/o semantic edges	97.75	95.50	0.000	10.843
	w/o query co-access	97.75	95.50	0.000	6.489
2Wiki	WARP-G	80.00	53.00	0.798	6.068
	w/o semantic edges	79.25	53.00	0.000	6.354
	w/o query co-access	79.25	53.00	0.000	6.173
MuSiQue	WARP-G	73.62	45.50	2.598	7.538
	w/o semantic edges	73.50	45.50	1.444	5.654
	w/o query co-access	73.71	46.00	0.971	8.923
NQ	WARP-G	87.06	69.50	0.000	6.066
	w/o semantic edges	87.06	69.50	0.000	3.968
	w/o query co-access	87.06	69.50	0.000	12.629

rules at a fixed region count. An independent frequency-only deployment would set that capacity externally; WARP-G determines it through conditional selection.

For gain-only selection versus WARP-G on MuSiQue, the CE@10 difference has Holm-adjusted $p = 0.2970$ and is not statistically significant. The selection sample contains at most 32 design queries; repeated design splits would help assess how the ranking rules transfer across workloads.

Frequency-only deployment. A deployment with a chosen region capacity can use the same partition and routing rules without measuring graph gains. First, run passage retrieval on historical queries and count how often each region is reached. Rank regions by this access count and build indexes for the highest-ranked regions up to the chosen capacity. At serving time, invoke a retained index only when baseline retrieval reaches its region, then fuse and rerank its candidates with the full-corpus passage results. This procedure uses query text and retrieval results without evidence annotations. Graph construction is spent on the retained indexes, with no additional probe graphs or conditional-selection runs. The capacity specifies a region count; the resulting construction tokens depend on the sizes and extraction costs of those regions. Our frequency-only controls use one region on 2Wiki and two on MuSiQue, matching the counts selected by WARP-G.

B.5 Dataset differences and applicability

Complete routing coverage is 99.5%, 94.0%, 87.5%, and 99.5% on HotpotQA, 2Wiki, MuSiQue, and NQ, respectively. This metric measures whether baseline routing reaches every region containing annotated evidence; it does not bound final retrieval quality because the passage baseline continues to search the full corpus.

HotpotQA and NQ illustrate two routes to passage-only serving: non-positive local probe means and localized gains that do not improve shared utility, respectively. On 2Wiki, one regional graph improves both ER@5 and CE@5. MuSiQue combines lower complete routing coverage with mixed evidence-recovery gains; its gain-only control also achieves higher test retrieval scores than conditional selection. Together, these diagnostics connect deployment outcomes to baseline coverage, regional access, and the query sample used to assess graph combinations.

MuSiQue random partitioning further illustrates that routing coverage, final ranking quality, and graph cost are distinct quantities: it reduces complete routing coverage to 76.0%, yet improves CE@10 over the default partition while invoking 651 graph calls. WARP-G is therefore most applicable when the design workload is reasonably representative, multi-passage relations matter, and baseline routing reaches regions from which graph propagation can recover useful evidence. Evaluation under larger and shifting design workloads remains an important direction for future work.