WHEN DOES GRAPH STRUCTURE HELP? A CITATION-GROUNDED BENCHMARK FOR SCIENTIFIC LITERATURE RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge graph structure is widely expected to improve scientific literature retrieval, but existing evaluations either presuppose the graph or score a generated answer rather than the retrieved set, so the retrieval step itself is rarely the quantity measured. KGRAF-BENCH exposes one corpus through both a flat and a typed-graph view, retrieves with one encoder at one matched budget, and scores the retrieved paper sets against the citation decisions of survey authors. Its construction is automated: given a list of source-paper arXiv identifiers, the pipeline assembles a corpus, extracts a typed graph over it, and generates questions whose ground truth is the citations of the source papers. We release two ML/AI corpora of 8,219 and 89,476 papers with a graph over each, 647 questions, the per-question features, along with 2 additional instantiations in the biomedical and physics domains.

Adding entity embeddings to the index improves recall over flat retrieval, $+.026$ $[+.010, +.043]$ pooled over 647 questions with no LLM call on either side. The advantage is consistent in direction and small in magnitude: over 20 configurations varying corpus scale, canonicalization threshold, encoder and construction method, entity-aware traversal is significantly ahead of flat retrieval in 52 of the 80 configurations tested. Selecting a strategy per question does not improve retrieval: a router trained on pre-retrieval features under cross-validation performs worse than the best fixed strategy. Changing the canonicalization threshold barely moves traversal's advantage, while the entity-similarity channel's swings widely and turns negative at the most aggressive setting.

1 INTRODUCTION

Systems such as PaperQA2 (Skarlinski et al., 2024), Ai2 ScholarQA (Singh et al., 2025b) and OpenAI (2025) synthesize scientific knowledge at scale, and Karpathy (2026)'s Auto-research extends this to autonomous experimentation; all are bounded by the quality of the literature search underneath. A natural question is whether augmenting that retrieval step with Knowledge Graph (KG) structure, organizing papers, methods, tasks and topics into a typed graph and traversing it during search, improves paper retrieval over flat similarity search.

This premise is widely adopted but has not been tested where it is made. Graph Retrieval-Augmented Generation (GraphRAG) covers a spectrum: Edge et al. (2024) build a graph index of text chunks and community summaries with no typed entity schema, while Think-on-Graph (Sun et al., 2024), G-Retriever (He et al., 2024) and LightRAG (Guo et al., 2025) traverse typed entity relationships to retrieve paths and subgraphs. All three retrieval channels we evaluate are visible in this literature. Direct paper similarity (channel a) is what flat retrieval uses and what every system above falls back on when the graph yields too little; the community-based approaches add entity-mediated discovery (channel b), in which entities matching the query bridge to papers that do not share its vocabulary; the traversal-based systems additionally rely on graph neighborhood expansion (channel c), following edges out from seed nodes. §3 defines the three. All are proposed on the premise that graph structure captures what cosine similarity misses (Zhang et al., 2025). Yet they evaluate on benchmarks that assume graph structure (Sun et al., 2024), on summarization tasks where community detection is a natural fit (Edge et al., 2024), or with LLM-as-judge metrics that conflate

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.

retrieval with generation (Guo et al., 2025), and none report conditions under which graph-based retrieval *degrades* performance.

Recent works have begun to close this gap, reporting end-to-end generation quality as their headline metric: Han et al. (2025) compare RAG and GraphRAG under a unified protocol, GraphRAG-Bench (Xiang et al., 2025) grades tasks by difficulty taking frequent underperformance as its motivation, and Zeng et al. (2025) find gains more moderate than reported once evaluation biases are removed.A parallel line compares retrieval mechanisms directly (Lee et al., 2025; Delile et al., 2024; Lei et al., 2025), but each proposes one method on its own corpus with its own questions, so the results do not compose into a statement about which modality to prefer. Established scientific retrieval benchmarks (Mysore et al., 2021; Wang et al., 2023; Ajith et al., 2024; Wang et al., 2025) expose no typed graph structure at all, so the two modalities cannot be compared on them. The benchmarks that do expose both modalities do not contrast them. Every retrieval method evaluated on STaRK (Wu et al., 2024) fuses textual and relational information rather than isolating either, and its baselines do not share an encoder; CRAG's (Yang et al., 2024) web-search and knowledge-graph interfaces are separately constructed stores queried by separate question sets, so the two are never scored on the same questions, and it scores generated answers. To the best of our knowledge no existing benchmark makes a *modality-matched* retrieval contrast its primary axis, applying the same encoder and budget to similarity-based and graph-structural retrieval over one corpus against shared ground truth and scoring the retrieved sets. KGRAF-BENCH is built for that contrast; Table 4 in Appendix A positions it against both lines. *Extended related work is in Appendix A.*

Per-query strategy selection is itself established. Query performance prediction estimates retrieval quality before relevance judgments (Amati et al., 2004; Yom-Tov et al., 2005; Zhou & Croft, 2007; Carmel & Yom-Tov, 2010) and has been used to pick between strategies per query (Arabzadeh et al., 2021; Dong et al., 2026). Those scorers read the query string alone, with no access to the corpus or the index; the entity-paper gap is computed entirely from the index and never reads the query's syntax. The two draw on disjoint information and could be combined. Ours needs no labels, no run of either pipeline, and no tuned thresholds.

KGRAF-BENCH comprises two corpora of machine learning papers with a typed knowledge graph over each, 8,219 papers in the primary corpus and 89,476 in an extension that widens the venue and year range of the same construction, and 647 questions in two sets: 274 standard questions generated with the cited papers' abstracts in view, biased toward flat similarity, and 373 relational questions generated from paragraph pairs and whole sections with no cited text in view at all, requiring cross-paper connections. Ground truth for each question is the set of papers the source authors cited in the extract it was generated from. §2 gives construction in full.

The evaluation answers four questions in turn. **RQ1**: does graph structure improve retrieval at all, measured against both a lexical baseline and a purely structural one (§4.1)? **RQ2**: how much does graph construction affect each channel (§4.2)? **RQ3**: which of these findings survive a change of corpus scale, encoder or domain (§4.3)? **RQ4**: can the choice of strategy be made per question (§4.4)?

KGRAF-BENCH scores retrieved paper sets rather than generated answers, with similarity-based and graph-structural retrieval applicable to one corpus at a matched budget and ground truth taken from the source authors' citation decisions (§2). It is a construction method as much as a dataset: the pipeline takes a list of source-paper arXiv identifiers and returns a corpus, a typed graph over it and citation-grounded questions, so the same contrast can be rebuilt on another corpus. We use it for a two more domains, biomedical and physics. Measuring four retrieval strategies on it, we find:

- Graph structure helps, consistently in direction and by a small margin. Across 20 configurations varying corpus scale, canonicalization threshold, encoder and construction method, entity-aware traversal is significantly ahead of flat retrieval in 52 of 80 paired tests and significantly behind in none. Stronger embeddings do not absorb the advantage: on Relational questions traversal separates from flat retrieval under BGE-large and Qwen3-Embedding, and it holds on a third, physics corpus
- We measure the benchmark's own noise floor and its cause. Almost every question has papers tied at identical similarity inside its top-50, so rebuilding the index shifts a strategy by up to .013 and changing device by up to .009. We set the threshold at .015 for any comparison between configurations, and hold our own results to it.

- Construction choices affect the two channels by different amounts. Across five canonicalization thresholds the traversal advantage varies by .014 and the entity-similarity advantage by .051, the latter falling below flat retrieval at the most aggressive value.
- Per-question strategy selection does not work. A cheap pre-retrieval signal, the entity-paper gap, does not separate the questions traversal rescues from those flat retrieval already handles, and a router trained on it under cross-validation does not beat the best fixed strategy.

We release both corpora, the constructed graph over each, all 647 questions, the per-question features, and the construction code with all prompts, so the benchmark can be run as shipped without repeating the extraction, or rebuilt from public APIs where licensing prevents us from redistributing paper text.[1]

2 KGRAF-BENCH

2.1 CORPUS AND KNOWLEDGE GRAPH

KGRAF-BENCH is built so that both retrieval modalities apply to the same corpus. Every paper and every extracted entity is embedded with one model, which lets cosine similarity rank papers directly, and typed edges connect each paper to the entities extracted from it, which lets traversal reach a paper through entities it shares with others. The two question sets then vary in how much they require those shared entities, and ground truth reflects expert curation instead of crowd annotation or synthetic generation. The corpus is derived from 7 source papers (L0), 6 surveys and 1 seminal methods paper, selected for citation count, recency and topical coverage at the intersection of knowledge graphs, retrieval and language models. The source papers are not part of the retrieval corpus: their citation decisions provide ground-truth labels and their LaTeX sources provide material for question generation. Ground-truth papers are resolved exclusively through the Semantic Scholar API with verified metadata (title, abstract $\geq$50 characters, unique paper ID), so every label maps to one verified corpus entry (Appendix C).

The 8,219 papers are organized in three layers, sized in Table 6 of Appendix C. L1 (724 papers) are those explicitly cited by the source papers. L2 (3,514) are papers that cite an L1 paper, collected over 538 L1 seeds with no venue filtering; these are *hard distractors*, sharing citation context with L1 but cited by no source paper. L3 (3,981) comes from keyword-venue search over top ML/AI venues 2021–2024 sorted by citation count, and supplies scale and topical diversity.

We construct a typed KG over the corpus using the Extract-Define-Canonicalize (EDC) pipeline (Zhang & Soh, 2024): for each paper an LLM extracts entities of five types (Method, Task, Topic, Dataset, Claim) from the abstract, and entities are canonicalized by embedding similarity at threshold 0.85. The primary graph has 8,218 paper nodes, 79,109 entity nodes and 166,734 edges over seven edge types. Of the entity nodes, 4,185 (5.3%) are shared by two or more papers, which are the cross-paper connections graph retrieval can exploit. No benchmark in the literature exposes one corpus through both a flat and a typed-entity view, so we construct the graph ourselves. Table 4 in Appendix A positions KGRAF-BENCH against every candidate we found, and Appendix C.5 gives the full schema.

We build the graph with a parallelized batch pipeline: 50 extraction workers, with canonicalization against a growing cache in batches of 100 papers. It takes 1.4 hours for 8,219 papers and 45 hours for the 89,476-paper extension, against 70 hours for the sequential pipeline on the smaller corpus. Both graphs are released built, so reproducing our numbers requires no extraction. The extension widens L3 to five further venues (ACL, EMNLP, NAACL, CVPR, AAAI) and to 2016–2025. It leaves the questions and their ground truth untouched, so it is the same benchmark against an eleven-fold larger pool of distractors rather than a larger benchmark, and absolute recall falls for every strategy as a result. We also retain an earlier sequential build, which canonicalizes each paper's entities against all previously extracted entities. The two differ in more than speed: the sequential run emits four entity types and five edge types, with no direct Paper$\rightarrow$Task edge, while the batch run emits five and seven. §4.2 measures how far this difference affects retrieval, and §4.3 varies the threshold and the encoder. Appendix I.1 gives the worker layout and the cache and batching scheme for both pipelines; the prompts are in the released construction code.

[1] Code and data: `https://anonymous.4open.science/r/kgraf-bench-6CB7`

Standard question (Peng et al. GraphRAG survey (Peng et al., 2024), §2.3 KBQA)

Source paragraph: "KBQA is a significant task in natural language processing, aiming to respond to user queries based on external knowledge bases [...] Previous surveys typically categorize existing KBQA approaches into two main types: Information Retrieval (IR)-based methods and Semantic Parsing (SP)-based methods. Specifically, IR-based methods retrieve information related to the query from the knowledge graph (KG) and use it to enhance the generation process. While SP-based methods generate a logical form (LF) for each query and execute it against knowledge bases to obtain the answer."

Generated question: *What are the two main methodological approaches used in Knowledge Base Question Answering systems and how do they differ in their mechanisms for answering questions?*

Ground truth (4 papers): UniKGQA (Jiang et al., 2023); RNG-KBQA (Ye et al., 2022); ReTraCk (Chen et al., 2021); ArcaneQA (Gu & Su, 2022)

Figure 1: One Standard question with the source paragraph it was generated from. The generator sees that paragraph together with the abstracts of the papers it cites, and the ground truth is those papers. Citations inside the quoted source text are replaced with [...].

2.2 Question Sets

The 274 Standard questions come from citation-dense paragraphs (≥ 2 cited corpus papers) in the source papers' LaTeX, with the cited abstracts in view during generation. Because the generator sees those abstracts, question vocabulary overlaps with abstract vocabulary, which biases this set toward flat similarity; the bias is intended, since Standard questions are the set on which similarity search should already be sufficient. An LLM also classifies each citation as *essential* or *background* and ground truth keeps the essential ones, which narrows it on 59 of the 274 and leaves the rest unchanged. This step was applied to the Standard set only; Relational ground truth is the union of all papers cited by the two source paragraphs, unfiltered.

The 373 Relational questions come from source text spanning more than one paragraph: 200 from pairs of paragraphs that share cited papers, 173 from whole sections citing many papers. Generation sees that text and *neither* titles nor abstracts of the cited papers, so this set tests where graph structure should help as Standard tests where similarity should suffice. Ground truth is everything the source text cites. Appendix C.7 gives the generation passes, the three relational question types and the local/global split of the Standard set, and the source-paper selection criteria are in Appendix C.1. Figure 1 shows a Standard question with its source paragraph; Relational examples and further cases are in Appendix C.2. All 647 questions were validated by the authors for relevance and consistency. A random sample of 50 was additionally rated against four quality criteria (Appendix K): 49 pass all four, and in all 50 the cited papers were judged relevant to the question, which is the assumption the ground truth rests on.

2.3 Ground Truth

Ground-truth papers are those the source paragraph cites, less any the LLM classified as background. The labels therefore record the citation decisions of the source paper authors, experts writing a survey in their own field, rather than crowd judgements or LLM-generated relevance. The LLM's role is subtractive only: it can drop a paper the authors cited but never add one, and it does so on 59 of the 647 questions. This is what distinguishes the labels from STaRK's largely synthesized queries and DORIS-MAE's LLM-annotated pairs. CRAG's answers, human-annotated with a manual validation pass, are better sourced than ours.

These labels are incomplete. A citation shows that the source authors judged a paper relevant, but its absence does not show that a paper is irrelevant, and a relevant yet uncited paper stays in L2 or L3 where it is scored as a distractor. Recall@K is therefore recall against the authors' citation decisions and not against all relevant literature, so absolute values understate what every strategy finds, the same incompleteness pooled IR test collections carry (Buckley & Voorhees, 2004; Sakai & Kando, 2008). All four strategies are scored against the same labels, so that understatement applies equally to each; it would distort the comparison only if one strategy were differentially good at surfacing relevant papers the authors chose not to cite, which we have not measured. Appendix C reports the 21 ground-truth papers our reference data initially mislabelled and how we corrected them.

A closed-book LLM baseline (Claude Sonnet 4.5, no retrieval) reaches 12.7% F1 across the 274 Standard questions, with 1 question (0.4%) exceeding F1 > 0.8, so the questions are not answerable from parametric memory alone.

We report Recall@K ($K \in \{10, 20, 50\}$), MRR and per-question strategy win rates at matched retrieval budgets. Each question is released as a JSON record holding the source paragraph, ground-truth paper IDs and titles, the question, and provenance metadata; release and reproduction detail is in the reproducibility statement at the end of the main text.

3 Retrieval Strategies

3.1 Retrieval Graph and Channels

Let $G = (V_P \cup V_E, T)$ be a knowledge graph with paper nodes V_P, extracted entity nodes V_E (Methods, Tasks, Topics, Claims, Datasets) and typed directed edges T. Every node v has an embedding $\mathbf{v} \in \mathbb{R}^d$, and for a question q we write $\mathrm{sim}(q, v) = \cos(\mathbf{q}, \mathbf{v})$ and $\mathrm{sim}(q, V_P) = (\mathrm{sim}(q, p))_{p \in V_P}$. For an entity e, its paper neighborhood is $\mathcal{P}(e) := \{p \in V_P : (p, e) \in T\}$, extended to sets by $\mathcal{P}(F) := \bigcup_{e \in F} \mathcal{P}(e)$. Traversal discovers nodes by following edges from a seed set $S \subseteq V_P \cup V_E$, and we write $N_k(S)$ for the paper nodes reachable from S within k edges.

All strategies use the same embedding model and corpus and return a ranked list of papers, differing in which of three channels they employ. Channel (a), *direct paper similarity*, scores papers by $f_a(q) := \mathrm{sim}(q, V_P)$. Channel (b), *entity-mediated discovery*, collects $f_b(q) := \mathcal{P}(E_q)$ for a set $E_q \subseteq V_E$ of entities selected by similarity, so entities matching the question bridge to papers that share concepts with q without sharing vocabulary. S_2 and S_3 select E_q by different rules, a rank cut and a similarity threshold, given below. Channel (c), *graph neighborhood expansion*, collects $f_c(q, S) := N_k(S)$, reaching papers structurally connected but invisible to both (a) and (b).

3.2 Strategy Definitions

S_1, flat RAG, uses channel (a) only: return the top-K papers by $\mathrm{sim}(q, V_P)$. No graph or LLM calls.

S_2, hybrid similarity, uses channels (a)+(b). Rank all nodes by $\mathrm{sim}(q, v)$ over $V_P \cup V_E$, collect $\mathcal{P}(e)$ for each entity e in the top results, merge the discovered papers, re-rank by $\mathrm{sim}(q, p)$ and return top-K. No LLM calls.

S_3, entity-aware graph retrieval, uses channels (a)+(b)+(c) in five steps: (1) select top-n seed papers S_P by $\mathrm{sim}(q, V_P)$; (2) collect $N_2(S_P)$; (3) an LLM extracts search phrases from q; each phrase ϕ is embedded and compared against the ten nearest nodes, and the entities among those scoring at least $\tau_e = 0.7$ form $\mathcal{E}_q = \bigcup_\phi \{e \in V_E : \mathrm{sim}(\phi, e) \geq \tau_e\}$, so the match is on the phrases and not on the question vector; (4) collect $\mathcal{P}(\mathcal{E}_q)$; (5) merge candidates, rank by $\mathrm{sim}(q, p)$, pad with a channel (a) fallback if fewer than K. One LLM call.

S_4, hybrid-scored entity-aware retrieval, takes the same candidate pool as S_3 and ranks it instead by

$$\mathrm{score}(p \mid q) = \mathrm{sim}(q, p) + \lambda \cdot \frac{|\{e \in \mathcal{E}_q : p \in \mathcal{P}(e)\}|}{|\mathcal{E}_q|} + \lambda \cdot \max_{e \in \mathcal{E}_q \,:\, p \in \mathcal{P}(e)} \mathrm{sim}(q, e) \tag{1}$$

with a single weight $\lambda = 0.1$ shared by both bonus terms, the second rewarding entity coverage and the third entity match strength. We did not tune the two separately, and both bonuses are bounded by λ because coverage and match strength each lie in $[0, 1]$, so the entity terms together can move a candidate by at most 2λ against a cosine similarity of comparable magnitude. Taking the third term's maximum over the empty set as zero, any candidate connected to no matched entity scores exactly $\mathrm{sim}(q, p)$, which includes every paper added by the fallback, so fallback papers are never promoted by the entity terms (Appendix B). S_4 and S_3 differ in the ranking function only. One LLM call.

The four strategies add channels cumulatively, S_1 using (a), S_2 (a)+(b), and S_3 and S_4 all three. Not every resulting contrast isolates a single channel. S_2 against S_1 does: the two share the corpus, the encoder, the budget and the ranking function, neither makes an LLM call, and they differ only in whether entity embeddings are present in the index. Channel (b) is therefore isolated exactly, which is why we report it first in §4.1. S_4 against S_3 isolates the ranking function, since the two draw

Table 1: ΔR@50 against S_1 on the primary configuration, paired bootstrap over questions (10,000 resamples); † marks an interval excluding zero. *Pooled* is a stratified estimate over all 647 questions, and is not narrower than the Relational column because Standard questions average 2.2 ground-truth papers and so carry roughly twice the per-question variance.

	Standard (274)	Relational (373)	Pooled (647)
S_2 Hybrid	+.017 [−.016, +.052]	+.032 [+.019, +.046]†	+.026 [+.010, +.043]†
S_3 Entity-aware	+.041 [+.006, +.075]†	+.005 [−.008, +.017]	+.020 [+.004, +.037]†
S_4 Hybrid-scored	+.042 [+.006, +.079]†	+.007 [−.005, +.019]	+.022 [+.005, +.039]†

from an identical candidate pool. S_3 against S_2 is not: it adds neighborhood expansion, an LLM phrase-extraction step and a different entity-matching rule at once, so we read it as a cumulative addition and make no claim that it isolates channel (c). S_3 and S_4 against S_1, is not an ablation, as S_3 and S_4 call an LLM once per question to enter the graph. Think-on-Graph (Sun et al., 2024) and LightRAG (Guo et al., 2025) enter the same way. The same model performs this extraction and the entity extraction that built the graph, so phrases and entity names share a vocabulary. Retrieving flat with those phrases separates the two: experimentally se confirm that the phrases do not retrieve on their own (Appendix I.4). Finally, at every K, a strategy returning fewer than K papers is padded with the next-best by $\mathrm{sim}(q, p)$ and one returning more is truncated after ranking, so the retrieval budget is identical across strategies by construction.

4 Results

Throughout, the *primary configuration* is the 8,219-paper corpus with its batch-constructed graph at canonicalization threshold $\tau_c = 0.85$ and MiniLM embeddings, scored on all 647 questions at a budget of $K = 50$. §4.3 varies one factor of it at a time.

4.1 Does Graph Structure Improve Recall over Flat Retrieval? (RQ1)

Recall and MRR for every strategy on the primary configuration are in Appendix D. Three controls isolate one component each, and none reaches flat retrieval on both question sets (Appendix I.4). The lexical control, BM25, reaches R@50 .423 on Standard questions and .101 on Relational against flat retrieval's .531 and .199; adding the extracted entity text to its index gives .447 and .104, so its shortfall is not a matter of how much text is indexed. The structural control, Personalized PageRank over the typed graph following Gutiérrez et al. (2024), propagates from query-matched entities without ranking papers by $\mathrm{sim}(q, p)$ at all, and reaches .460 on Standard, below flat retrieval, and .202 on Relational, marginally above its .199.

Table 1 gives paired bootstrap differences in R@50 against S_1, by question set and pooled over all 647 questions in KGRAF-Bench. Pooled, S_2 gains +.026, S_3 +.020 and S_4 +.022, and all three intervals exclude zero. Combining embeddings with typed structure therefore improves recall over flat retrieval when the two question sets are taken together.

S_2 differs from flat retrieval only in whether entity embeddings are present in the index. It gains +.032 [+.019, +.046] on Relational questions, and +.017 on Standard, where the interval includes zero. S_3 and S_4 gain +.041 [+.006, +.075] and +.042 [+.006, +.079] on Standard, with S_3 significant on all four metrics there, and +.005 and +.007 on Relational, where both are positive but not separable from flat retrieval. §4.2 examines how far this depends on how the graph was built.

4.2 How Sensitive Are the Channels to Graph Construction? (RQ2)

Both graphs in Table 2 cover the same 8,219 papers at the same canonicalization threshold ($\tau_c = 0.85$), encoder and retrieval budget, and differ in what the LLM extractor emits (Appendix C.5). On the batch graph, S_3 and S_4 separate from flat retrieval on Standard questions and S_2 on Relational (Table 1). On the sequential graph, no strategy separates on Standard (+.026, +.025 and +.031) and all three do on Relational (+.016, +.014 and +.013). Corpus, questions, encoder, budget and threshold are fixed across the two rows, so the rows differ only in how the graph was built. The sequential Standard

Table 2: Two graphs over the same 8,219 papers at the same canonicalization threshold $\tau_c = 0.85$, retrieved over with the same encoder and the same budget, differing in extraction schema and in merge order. Upper block: properties of each graph, with ADDR_TASK the Paper→Task edge and *shared ent.* entities appearing in two or more papers. Lower block: ΔR@50 against S_1 for the three strategies that use the graph, paired bootstrap, † marking an interval excluding zero.

	Nodes / edges		Types		Paper-to-entity edges	
	entities	edges	entity	edge	ADDR_TASK	shared ent.
Sequential	46,304	116,978	4	5	absent	6,399 (13.8%)
Batch (primary)	79,109	166,734	5	7	14,492	4,185 (5.3%)
	Standard (274 questions)			Relational (373 questions)		
	S_2	S_3	S_4	S_2	S_3	S_4
Sequential	+.026	+.025	+.031	+.016†	+.014†	+.013†
Batch (primary)	+.017	+.041†	+.042†	+.032†	+.005	+.007

estimates are positive and of the same order as the batch ones, so what differs there is which intervals exclude zero. We therefore test the row-to-row differences directly, pairing each question across builds: five of six are null, the exception being S_2 on Relational at +.017 [+.007, +.026], against +.015 [−.017, +.049] and +.011 [−.021, +.044] for traversal on Standard. Per-question R@50 has a standard deviation near 0.30 on Standard, so 274 questions cannot resolve a difference as small as our 0.015 floor; about 1,500 would be needed. These two builds therefore show the entity-similarity channel to be sensitive to construction, and the traversal channels not measurably so.

The sequential schema has no direct Paper→Task edge, so 5,358 of its 5,535 Task nodes carry no edge to any paper and can connect no two papers. The batch schema adds that edge and the Dataset type, giving traversal 23,814 paper-to-entity edges it does not have on the sequential graph.

The builds also differ in merge aggressiveness, how readily the canonicalizer collapses near-duplicate entity mentions into one node. Across five canonicalization thresholds on the batch graph, traversal never falls below flat retrieval, while S_2 does at $\tau_c = 0.70$ (Appendix I.2). The batch graph also holds fewer entities shared by two or more papers than the sequential one, 4,185 (5.3%) against 6,399 (13.8%), and shared entities are what bridge papers with no vocabulary in common. The bridging traversal exploits on the batch graph therefore does not come from more aggressive merging. The same threshold applied to different entity inventories still merges differently, so the builds differ in schema and in canonicalization at once. A third build, the batch schema with sequential-style merging, would separate the two factors, and we did not construct one (§5).

4.3 Does Scale, Encoder or Domain Reverse the Advantage? (RQ3)

Rebuilding the retrieval index leaves flat retrieval unchanged, every question returning the same ranked list on both of the backends we ran, the NebulaGraph store described in Appendix B and the file-based index the release ships. The strategies that read entities out of the graph are not invariant: they move by .003 to .013 in R@50 between those two backends, and by a further .002 to .009 when one configuration is run on different hardware (Appendix I.6). We set the threshold at 0.015; Appendix I.5 reports bootstrap intervals for every contrast and metric. We do not read a difference between two configurations as an effect unless it exceeds that floor. That bound is on differences across configurations, where the strategies are scored on the same questions and the paired intervals reported above apply.

Table 3 tallies the 20 configurations we built, each rebuilding KGRAF-BENCH with one factor changed from the primary configuration; Appendix I reports every block in full. Traversal (S_3) is positive in 52 of its 80 paired tests against S_1 and negative in none, and the hybrid-scored variant (S_4) in 46 with none negative. The entity-embedding channel (S_2) is positive in 14, and no test in the grid is significantly negative for any strategy. We read the absence of negatives rather than the count of positives, which would need a multiple-comparison correction.

Two interventions hold graph construction fixed. At 89,476 papers, 799,737 entities and 1,825,762 edges, a graph the batch pipeline builds in 45 hours, traversal is positive in 4 of 8 tests, and the entity-

Table 3: Paired tests against S_1 across every configuration we built. Each configuration changes one factor from the primary configuration and is scored on one question set, contributing four tests per strategy, one per metric; cells count the tests whose bootstrap interval lies entirely above zero. No test is significantly negative for any strategy: S_3's estimate is positive in all 80, and S_2's is negative in 32 without reaching significance in any of them.

Intervention	Configs	Tests	S_2	S_3	S_4
Canonicalization threshold, 5 τ_c	10	40	8	**27**	**24**
Extraction schema, sequential	2	8	1	2	4
Encoder, 2 stronger models	4	16	4	**14**	**10**
Corpus scale, 89K papers	2	8	1	4	4
Scale × encoder, 89K + Qwen3	2	8	0	5	4
Total, ML/AI question sets	20	80	14	**52**	**46**
of which significantly negative			0	**0**	**0**
Further domains, own question sets, not pooled with the above					
Biomedical, 781 papers (L1 only)	1	4	0	0	0
Biomedical, 8,387 papers (+distractors)	1	4	0	0	0
Physics, 11,714 papers, 1,683 Qs	2	8	3	**7**	6

embedding channel falls below flat retrieval there on Standard questions ($-.012$). Re-embedding the primary graph with two stronger encoders leaves traversal positive in 14 of 16 tests and negative in none, which is the strongest block in the grid for traversal; on Relational questions traversal separates from flat retrieval under both stronger encoders and does not under MiniLM. The entity-embedding channel does not track encoder strength, gaining $+.033$ on Relational questions under BGE-large against $+.006$ under Qwen3 and $+.032$ under MiniLM. Hardware variation is reported in Appendix I.6, where it contributes to the reproducibility floor rather than to this grid.

Instantiating the benchmark on biomedical literature reproduces the ordering, with traversal ahead of flat retrieval by $+.024$ on the 781-paper L1-only corpus and $+.021$ once distractors are added, both above the reproducibility floor. But on its own 111 questions not one of the 24 tests, 8 for each of the three strategies, reaches significance in either direction, and unlike in ML/AI the advantage does not grow with the distractor pool. The extension establishes that the construction pipeline runs on another literature; it does not establish the retrieval finding there (Appendix J).

A third instantiation on physics uses 1,683 questions over 11,714 papers. Traversal is significantly ahead of flat retrieval in 7 of its 8 paired tests and behind in none, the exception being R@50 on Standard questions; it gains $+.025$ [$+.012$, $+.039$] on Relational R@50 where the entity-embedding channel gains $+.028$. So the ordering reproduces in a third literature, and traversal no longer leads the cheaper channel. That corpus shares 1.8% of its entities between two or more papers, against 5.3% on the primary graph and 13.8% on the sequential one, and shared entities are the only route by which traversal reaches a paper flat similarity misses. Physics entity names are specific where ML method vocabulary is reused, so canonicalization merges far less. The questions are LLM-generated and have not been author-validated (Appendix F for construction details).

4.4 Can the Retrieval Strategy Be Chosen Per Question? (RQ4)

The four strategies do not all win on the same questions, so choosing between them per question could beat any fixed choice. Doing that needs a signal available before retrieval runs. We test the *entity-paper gap*, $\text{epg}(q) = \max_{e \in V_E} \text{sim}(q, e) - \max_{p \in V_P} \text{sim}(q, p)$, positive when some entity matches the question more closely than any paper abstract, costing two cosine lookups against cached embeddings with no LLM call and no relevance labels.

It does not separate. Among the Relational questions flat retrieval fails on, the 22 that S_4 rescues have a larger gap than the 64 it does not, but the interval spans zero, $d = +0.23$ [-0.25, $+0.83$]. On Standard questions the contrast is null as well ($|d| \leq 0.14$). Figure 2 shows why: the marked groups overlap on both question sets.

Acting on it does not help either. A router trained on seven features available before retrieval, under five-fold cross-validation, returns $-.002$ [$-.016$, $+.012$] on Standard and $-.029$ [$-.042$, $-.017$] on

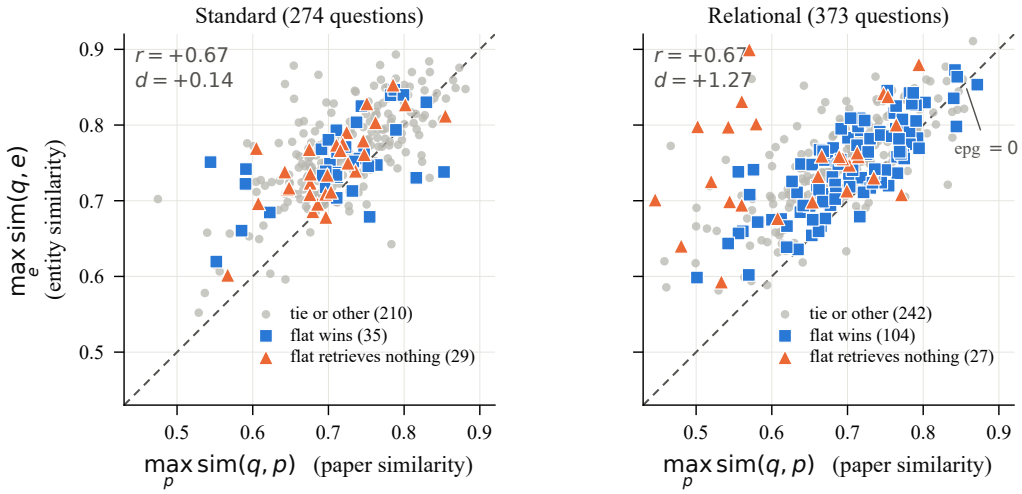


Figure 2: Entity similarity against paper similarity for every question on the primary configuration, Standard left and Relational right. The dashed diagonal is epg $= 0$. Triangles mark questions where flat retrieval retrieves no ground-truth paper while S_4 retrieves at least one; squares mark questions where flat retrieval wins outright. The two groups overlap on both sets.

Relational against the best fixed strategy, the latter significantly worse. An oracle given the realised outcome would gain $+.090$ and $+.042$, so the headroom exists and these features do not reach it. Appendix E gives the router, and two looser contrasts that produce larger effects by comparing groups defined on retrieval outcome. The gap is cheap to compute and appears negative, it does not help locating the questions where the choice of strategy matters.

5 Discussion and Limitations

For a system choosing among these strategies, the claim that transfers is the ordering rather than the size of any gap. Entity-aware traversal was never significantly behind flat retrieval in 80 paired tests across 20 configurations, while the cheaper entity-embedding channel was ahead in only 14 and behind in none. A system that cannot re-measure on its own corpus should prefer traversal and pay the LLM call; one that can should check the two conditions under which the cheap channel inverts (§4.3, with deployment guidance in Appendix H). Two extensions follow directly from the released pipeline: instantiating the benchmark on further domains, and generating question sets that vary the dependence on graph structure independently of ground-truth set size, which *(iii)* below identifies as an explanation for the split that we cannot rule out.

Several limitations bound what we have shown. *(i) Domain scope.* The biomedical extension (Appendix J) reproduces the direction of the graph advantage but separates nothing significantly on its 111 questions, and separating it would take roughly three times as many. *(ii) KG construction.* One canonicalization pipeline and one entity schema, and the two builds we compare differ in schema and in merge order at once; the third build that would separate them was not constructed. *(iii) The two regimes are partly a property of the benchmark.* The question sets differ by design in how much they depend on graph structure, and also differ nine-fold in ground-truth set size, 2.2 papers against 20.7, which is a simpler mechanism for the split than abstract visibility and one we cannot separate from it with the question sets we have. The split should be read as established on KGRAF-Bench, not for scientific search in general. *(iv) Effect sizes near the reproducibility floor.* The within-condition effects we report are significant but small, and the same size as the 0.015 cross-configuration floor measured in §4.3, which is why our cross-condition claims are about sign and ordering. Appendix G takes these further and covers question quality, the absence of a trained query-performance baseline, and the clustering of questions within source papers.

AI USE STATEMENT

Generative AI tools are a core component of the benchmark construction pipeline itself, and this use is described in the main text rather than only disclosed here: LLMs perform entity and relation extraction, canonicalization, question generation, and the essential/background classification that narrows the ground-truth set on 59 of the 647 questions (Section 2). The models and versions are reported in Appendix B and every prompt ships with the construction code. Ground truth is derived from the citation decisions of the source papers' authors; the classification step can only remove papers those authors cited, never add one, and no LLM scores, ranks or judges a retrieved set at any point in the evaluation.

Generative AI tools were also used to assist with reproducibility work on this submission: re-running experiments, recomputing reported quantities from the released artifacts and reconciling the two, diagnosing a defect in how per-question similarity features were computed, and auditing the paper's numbers against the runs behind them. Where that work changed a reported value, the authors verified the change against the artifacts before adopting it. AI assistance was also used for code in the experiment and analysis scripts.

We used AI tools for literature search and discovery, to surface candidate related work across the GraphRAG, query-performance-prediction and scientific-retrieval lines this paper positions itself against. Candidates were a starting point rather than a result: the authors read each work before citing it, and are responsible for every characterization of prior work in the paper and in Appendix A.

Generative AI tools were used in the writing beyond copy-editing. Alongside tightening and copy-editing prose throughout, they drafted passages that the authors then revised, principally in the appendices, and adapted the paper's interpretation to corrected numbers. The research questions, the experimental design, the analysis plan and the interpretation of the results are the authors' own, and generative AI tools were not used to generate research ideas, to design the study or its experiments, or to produce the reported experimental results. Every AI-assisted change to the text, the code and the reported numbers was reviewed by the authors, and we take responsibility for the final content of this work, including all text, claims and artifacts.

REPRODUCIBILITY STATEMENT

Every number in this paper is recomputable from the released artifacts, and the release is built so that the expensive step does not have to be repeated. We ship the four constructed knowledge graphs directly: the sequential 8K graph (46,304 entity nodes, 116,978 edges), the batch 8K graph that is the primary configuration (79,109 and 166,734), the 89K extension (799,737 and 1,825,762) and the biomedical instantiation (84,316 and 174,186). A hydration script reassembles any of them into the form the evaluation harness loads, in roughly fifteen minutes with a Semantic Scholar API key. The alternative, rebuilding a graph from the corpus with LLM extraction, takes about 70 hours sequentially or 83 minutes with the batch pipeline, and no reader has to do it. The construction code, all prompts and the corpus specification ship as well, for anyone who wants to.

Alongside the graphs we release all 647 questions with their ground truth and source paragraphs, the per-question features, the per-strategy retrieval results for every configuration, and the full evaluation harness. Section 2 specifies corpus selection, graph construction and ground-truth derivation; Section 3 gives the four strategies and the matched-budget protocol, including the hybrid scoring function. Implementation details, model versions and retrieval hyperparameters are in Appendix B; complete per-strategy results are in Appendix D. We also release the toolkit that instantiates a benchmark from nothing but a set of source-paper arXiv IDs, automating corpus assembly, graph construction and question generation; that is how the biomedical extension was built, and it is the route by which the two-regime distinction can be tested beyond ML/AI.

Two things are deliberately not redistributed. Paper abstracts are third-party text, so the graphs ship with titles, venues and years and the hydration script refills abstracts from the Semantic Scholar API. The 89K graph ships without its embedding matrices, which would add roughly 650 MB; the hydration script recomputes them with the same model and the same text construction used to build it.

We are explicit about what cannot be reproduced exactly, because the effects we report are small enough for it to matter. Flat retrieval is bit-exact: rebuilding the index returns the same ranked list for every question. The strategies that read entities out of the graph are not, moving .003 to .013 in

R@50 across index rebuilds and a further .002 to .009 across hardware, which is why we treat 0.015 as a floor and rest cross-condition claims on sign and ordering rather than magnitude (Appendix I.6). Batch construction is itself only reproducible to within a build, because canonicalization matches against a cache that fills in a different order across parallel workers (Appendix I.1). Per-question similarity features are computed against the paper and entity indices separately; a flag on the feature routine reproduces the earlier behaviour, which drew one ranking over both node types together, for anyone comparing against previously published values.

Robustness is documented rather than asserted: canonicalization threshold sensitivity (Appendix I.2), batch versus sequential construction (Appendix I.1), embedding-model substitution (Appendix I.3), non-neural baselines (Appendix I.4), bootstrap confidence intervals (Appendix I.5), hardware non-determinism (Appendix I.6), the cross-domain biomedical extension (Appendix J) and the question quality audit (Appendix K).

REFERENCES

Anirudh Ajith, Mengzhou Xia, Alexis Chevalier, Tanya Goyal, Danqi Chen, and Tianyu Gao. LitSearch: A retrieval benchmark for scientific literature search. In *Proceedings of EMNLP*, 2024.

Gianni Amati, Claudio Carpineto, and Giovanni Romano. Query difficulty, robustness, and selective application of query expansion. In *Proceedings of the 26th European Conference on Information Retrieval (ECIR)*, pp. 127–137, 2004.

Negar Arabzadeh, Xinyi Yan, and Charles L. A. Clarke. Predicting efficiency/effectiveness trade-offs for dense vs. sparse retrieval strategy selection. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM)*, 2021. arXiv:2109.10739.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *Proceedings of ICLR*, 2024.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. MS MARCO: A human generated MAchine reading COmprehension dataset. In *Proceedings of the Workshop on Cognitive Computation*, 2016.

Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 25–32, 2004.

David Carmel and Elad Yom-Tov. *Estimating the Query Difficulty for Information Retrieval*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan and Claypool, 2010.

Shuang Chen, Qian Liu, Zhiwei Yu, Chin-Yew Lin, Jian-Guang Lou, and Feng Jiang. ReTraCk: A flexible and efficient framework for knowledge base question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 325–336, 2021.

Julien Delile, Srayanta Mukherjee, Anton Van Pamel, and Leonid Zhukov. Graph-based retriever captures the long tail of biomedical knowledge. *arXiv preprint arXiv:2402.12352*, 2024. Presented at the ICML 2024 Workshop on Machine Learning for Life and Material Science (ML4LMS), non-archival.

Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, Enrico Motta, and Harald Sack. AI-KG: An automatically generated knowledge graph of artificial intelligence. In *Proceedings of the 19th International Semantic Web Conference (ISWC)*, 2020.

Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. CS-KG: A large-scale knowledge graph of research entities and claims in computer science. In *Proceedings of the 21st International Semantic Web Conference (ISWC)*, 2022.

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, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Dongzhe Fan, Zheyi Xue, Siyuan Liu, and Qiaoyu Tan. Do we still need GraphRAG? benchmarking RAG and GraphRAG for agentic search systems. *arXiv preprint arXiv:2604.09666*, 2026.

Michael Färber and David Lamprecht. Linked papers with code: The latest in machine learning as an RDF knowledge graph. *arXiv preprint arXiv:2310.20475*, 2023.

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models. *arXiv preprint arXiv:2310.04560*, 2023.

Yu Gu and Yu Su. ArcaneQA: Dynamic program induction and contextualized encoding for knowledge base question answering. In *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 1718–1731, 2022.

Jiayan Guo, Lun Du, Hengyu Liu, Mengyu Zhou, Xinyi He, and Shi Han. GPT4Graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking. *arXiv preprint arXiv:2305.15066*, 2023.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. In *Proceedings of EMNLP*, 2025. arXiv:2410.05779.

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.

Safayat Bin Hakim, Muhammad Adil, Alvaro Velasquez, and Houbing Herbert Song. SymRAG: Efficient neuro-symbolic retrieval through adaptive query routing. *arXiv preprint arXiv:2506.12981*, 2025.

Haoyu Han, Li Ma, Yu Wang, Harry Shomer, Yongjia Lei, Zhisheng Qi, Kai Guo, Zhigang Hua, Bo Long, Hui Liu, Charu C. Aggarwal, and Jiliang Tang. RAG vs. GraphRAG: A systematic evaluation and key insights. *arXiv preprint arXiv:2502.11371*, 2025.

Taher H. Haveliwala. Topic-sensitive PageRank. In *Proceedings of the 11th International Conference on World Wide Web*, 2002.

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

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*, 2024.

Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. UniKGQA: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023.

Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 2023. arXiv:2312.02783.

Andrej Karpathy. autoresearch: AI agents running research on single-GPU nanochat training automatically, 2026. GitHub repository, `https://github.com/karpathy/autoresearch`.

Rodney Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, Miles Crawford, Doug Downey, Jason Dunkelberger, Oren Etzioni, Rob Evans, Sergey Feldman, Joseph Gorney, David Graham, Fangzhou Hu, Regan Huff, Daniel King, Sebastian Kohlmeier, Bailey Kuehl, Michael Langan, Daniel Lin, Haokun Liu, Kyle Lo, Jaron Lochner, Kelsey MacMillan, Tyler Murray, Chris Newell, Smita Rao, Shaurya Rohatgi, Paul Sayre, Zejiang Shen, Amanpreet

Singh, Luca Soldaini, Shivashankar Subramanian, Amber Tanaka, Alex D. Wade, Linda Wagner, Lucy Lu Wang, Chris Wilhelm, Caroline Wu, Jiangjiang Yang, Angele Zamarron, Madeleine Van Zuylen, and Daniel S. Weld. The semantic scholar open data platform. *arXiv preprint arXiv:2301.10140*, 2023.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Meng-Chieh Lee, Qi Zhu, Costas Mavromatis, Zhen Han, Soji Adeshina, Vassilis N. Ioannidis, Huzefa Rangwala, and Christos Faloutsos. HybGRAG: Hybrid retrieval-augmented generation on textual and relational knowledge bases. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2025. arXiv:2412.16311.

Yongjia Lei, Haoyu Han, Ryan A. Rossi, Franck Dernoncourt, Nedim Lipka, Mahantesh M Halappanavar, Jiliang Tang, and Yu Wang. Mixture of structural-and-textual retrieval over text-rich graph knowledge bases. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 18306–18321, 2025. doi: 10.18653/v1/2025.findings-acl.941. arXiv:2502.20317.

Chang Liu and Bo Wu. Evaluating large language models on graphs: Performance insights and comparative analysis. *arXiv preprint arXiv:2308.11224*, 2023.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3219–3232, 2018.

Chuan Meng, Negar Arabzadeh, Arian Askari, Mohammad Aliannejadi, and Maarten de Rijke. Query performance prediction using relevance judgments generated by large language models. *ACM Transactions on Information Systems*, 2024. arXiv:2404.01012.

Sheshera Mysore, Timothy J. O'Gorman, Andrew McCallum, and Hamed Zamani. CSFCube – a test collection of computer science research articles for faceted query by example. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

OpenAI. Deep research, 2025. ChatGPT feature, launched February 2025.

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*, 2024. arXiv:2408.08921.

Tetsuya Sakai and Noriko Kando. On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Information Retrieval*, 11(5):447–470, 2008.

Chenyang Shao, Fengli Xu, and Yong Li. SciNet: Evaluating AI agents in relation-aware scientific literature retrieval. *arXiv preprint arXiv:2601.03260*, 2025.

Aditi Singh, Abul Ehtesham, Saket Kumar, Tala Talaei Khoei, and Athanasios V. Vasilakos. Agentic retrieval-augmented generation: A survey on agentic RAG. *arXiv preprint arXiv:2501.09136*, 2025a.

Amanpreet Singh, Joseph Chee Chang, Chloe Anastasiades, Dany Haddad, Aakanksha Naik, Amber Tanaka, Angele Zamarron, Cecile Nguyen, Jena D. Hwang, Jason Dunkelberger, Matt Latzke, Smita Rao, Jaron Lochner, Rob Evans, Rodney Kinney, Daniel S. Weld, Doug Downey, and Sergey Feldman. Ai2 Scholar QA: Organized literature synthesis with attribution. *arXiv preprint arXiv:2504.10861*, 2025b.

Michael D. Skarlinski, Sam Cox, Jon M. Laurent, James D. Braza, Michaela M. Hinks, Michael J. Hammerling, Manvitha Ponnapati, Samuel G. Rodriques, and Andrew D. White. Language agents achieve superhuman synthesis of scientific knowledge. *arXiv preprint arXiv:2409.13740*, 2024.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *Proceedings of ICLR*, 2024. arXiv:2307.07697.

Zeinab Sadat Taghavi, Ali Modarressi, Hinrich Schütze, and Andreas Marfurt. With argus eyes: Assessing retrieval gaps via uncertainty scoring to detect and remedy retrieval blind spots. *arXiv preprint arXiv:2602.09616*, 2026.

Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Proceedings of the Sixth International Conference on Data Mining (ICDM)*, 2006.

Jianyou Wang, Kaicheng Wang, Xiaoyue Wang, Prudhviraj Naidu, Leon Bergen, and Ramamohan Paturi. DORIS-MAE: Scientific document retrieval using multi-level aspect-based queries. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 36, pp. 38404–38419, 2023.

Jianyou Wang, Weili Cao, Kaicheng Wang, Xiaoyue Wang, Ashish Dalvi, Gino Prasad, Qishan Liang, Hsuan-lin Her, Ming Wang, Qin Yang, Gene W. Yeo, David E. Neal, Maxim Khan, Christopher D. Rosin, Ramamohan Paturi, and Leon Bergen. EvidenceBench: A benchmark for extracting evidence from biomedical papers. In *Conference on Language Modeling (COLM)*, 2025. arXiv:2504.18736.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, 2020.

Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022.

Shirley Wu, Shiyu Zhao, Michihiro Yasunaga, Kexin Huang, Kaidi Cao, Qian Huang, Vassilis N. Ioannidis, Karthik Subbian, James Zou, and Jure Leskovec. STaRK: Benchmarking LLM retrieval on textual and relational knowledge bases. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2024.

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

Yilin Xiao, Junnan Dong, Chuang Zhou, Su Dong, Qian-wen Zhang, Di Yin, Xing Sun, and Xiao Huang. GraphRAG-Bench: Challenging domain-specific reasoning for evaluating graph retrieval-augmented generation. *arXiv preprint arXiv:2506.02404*, 2025.

Chuan Xu, Qiaosheng Chen, Yutong Feng, and Gong Cheng. mmRAG: A modular benchmark for retrieval-augmented generation over text, tables, and knowledge graphs. In *International Workshop on the Semantic Web*, 2025. arXiv:2505.11180.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas Scheffer, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen-tau Yih, and Xin Luna Dong. CRAG – comprehensive RAG benchmark. *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2024.

Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. RNG-KBQA: Generation augmented iterative ranking for knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of ACL*, 2016.

Elad Yom-Tov, Shai Fine, David Carmel, and Adam Darlow. Learning to estimate query difficulty: Including applications to missing content detection and distributed information retrieval. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 512–519, 2005.

Qiming Zeng, Hao Luo, Yuhao Lin, Yicheng Jin, Yuxiang Wang, Fangcheng Fu, Xiao Yan, and Jiawei Jiang. How significant are the real performance gains? an unbiased evaluation framework for GraphRAG. *arXiv preprint arXiv:2506.06331*, 2025.

Bowen Zhang and Harold Soh. Extract, define, canonicalize: An LLM-based framework for knowledge graph construction. In *Proceedings of EMNLP*, 2024.

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

Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Yijian Qin, Simin Wu, and Wenwu Zhu. LLM4DyG: Can large language models solve spatial-temporal problems on dynamic graphs? *arXiv preprint arXiv:2310.17110*, 2023.

Junhao Zheng, Shengjie Qiu, Chengming Shi, and Qianli Ma. Towards lifelong learning of large language models: A survey. *ACM Computing Surveys*, 2024. arXiv:2406.06391.

Yun Zhou and W. Bruce Croft. Query performance prediction in web search environments. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 543–550, 2007.

Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. LLMs for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *arXiv preprint arXiv:2305.13168*, 2023.

Table 4: Positioning of KGRAF-BENCH. *Graph*: the corpus exposes typed structure, so graph retrieval is applicable at all. *Retrieval only*: retrieval is scored independently of generation. *Matched*: modalities compared on one corpus, embeddings and budget against shared ground truth. *Predicts*: whether a pre-retrieval signal can be tested against which modality wins, which needs per-question features released alongside the retrieved sets. The last two columns are what KGRAF-BENCH adds; the signal we test on it is a null (§4.4). STaRK's 34K queries are largely synthesized (13,323 against 84 human-written on its MAG split), so they are not a scale advantage over our 647.

Benchmark / study	Corpus	Qs	Graph	Retrieval only	Matched cond.	Predicts when
General-purpose retrieval and KGQA benchmarks						
CRAG (Yang et al., 2024)	mock APIs	4.4K	Yes	No	No	No
STaRK (Wu et al., 2024)	1.4M nodes	34K	Yes	Yes	No	No
Scientific literature retrieval benchmarks						
CSFCube (Mysore et al., 2021)	CS abstracts	50	No	Yes	n/a	No
DORIS-MAE (Wang et al., 2023)	CS abstracts	100	No	Yes	n/a	No
LitSearch (Ajith et al., 2024)	S2 corpus	597	No	Yes	n/a	No
EvidenceBench (Wang et al., 2025)	biomed papers	107K‡	No	Yes	n/a	No
Graph-versus-flat retrieval comparisons						
HybGRAG (Lee et al., 2025)	STaRK	n/r	Yes	Yes	No	No
Delile et al. (2024)	biomed KG	n/r	Yes	Yes	No	Partial
Lei et al. (2025)	text-rich KBs	n/r	Yes	Yes	No	Partial
Han et al. (2025)	existing	varies	Yes	No	No	Partial
GraphRAG-Bench (Xiang et al., 2025)	2 domains	4.1K	Yes	No	No	Partial
KGRAF-BENCH	8.2K papers	647	Yes	Yes	**Yes**	**Yes**

A EXTENDED RELATED WORK

Recent surveys provide comprehensive coverage of GraphRAG systems, organized around pipeline stages and system taxonomies (Peng et al., 2024; Zhang et al., 2025). We focus here on the lines of work most relevant to our contribution: graph-augmented retrieval systems, comparative evaluations of RAG and GraphRAG, adaptive retrieval routing, and benchmarks for scientific literature retrieval.

Graph-augmented retrieval. GraphRAG (Edge et al., 2024) builds a static graph index with community summaries for query-focused summarization. LightRAG (Guo et al., 2025) extracts entities and relationships into a knowledge graph and applies dual-level retrieval (low-level entity-specific and high-level topic-based) on every query, using graph structure unconditionally regardless of question type. Think-on-Graph (Sun et al., 2024) performs LLM beam search over a KG, evaluating on KGQA benchmarks. G-Retriever (He et al., 2024) formulates subgraph extraction as a Prize-Collecting Steiner Tree problem. These systems evaluate on benchmarks pre-selected to favor graph structure or use LLM-as-judge metrics that conflate retrieval with generation quality, and do not report conditions under which graph retrieval degrades performance or identify when simpler similarity search suffices.

RAG vs GraphRAG comparisons. We position our work relative to concurrent efforts along three axes: what is evaluated (end-to-end vs retrieval), how ground truth is obtained, and whether predictive signals are identified.

Han et al. (2025) present a comprehensive comparison, with a unified protocol across HotpotQA, MuSiQue, and NarrativeQA. They find distinct strengths for RAG and GraphRAG and explore integration strategies. However, they evaluate end-to-end generation on existing benchmarks with crowd-sourced ground truth, making it difficult to attribute improvements to retrieval specifically.

GraphRAG-Bench (Xiang et al., 2025) evaluates GraphRAG frameworks on two corpora, Project Gutenberg novels and NCCN clinical guidelines, across 4,076 questions at four ascending difficulty levels, and reports retrieval-stage metrics (Evidence Recall, Context Relevance) alongside generation accuracy. They find that vanilla RAG wins on simple fact retrieval while graph structure helps as questions grow more complex. Retrieval budgets are not held constant across conditions, however: GraphRAG(global)'s prompt grows from 7,800 to 40,000 tokens across difficulty levels, so retrieval

quality and context budget move together and the comparison does not isolate either. A separate benchmark also named GraphRAG-Bench (Xiao et al., 2025) derives 1,018 college-level questions from 20 computer science textbooks spanning 16 disciplines; its retrieval evaluation reports indexing and retrieval *time* rather than retrieval quality, scoring quality at the answer and rationale level instead.

Zeng et al. (2025) identify that existing GraphRAG evaluations suffer from unrelated questions and LLM-based assessment biases. They propose an unbiased framework and find that performance gains are much more moderate than reported. Their contribution is methodological rather than a new benchmark.

We also distinguish mmRAG (Xu et al., 2025), which modularizes RAG evaluation across text, tables and knowledge graphs as three separate sources. Our corpus instead exposes one document set through both a flat and a graph view, which is what makes the matched comparison possible: in mmRAG a difference between sources confounds modality with content, whereas here the content is held fixed by construction.

Fan et al. (2026) introduce RAGSearch, evaluating under agentic search with dynamic multi-round retrieval. They find that agentic search narrows the gap, but GraphRAG remains advantageous for complex multi-hop reasoning. Similarly to Han et al. (2025), they use existing benchmarks and evaluate end-to-end.

Our work complements these efforts: we isolate the retrieval step (four strategies sharing the same embeddings and corpus), use expert-grounded ground truth derived from source paper citations, and report the entity-paper gap as a cheap pre-retrieval signal that does not locate the questions on which the choice of retrieval modality matters.

Adaptive retrieval routing. Adaptive-RAG (Jeong et al., 2024) routes between retrieval depths based on query complexity. Self-RAG (Asai et al., 2024) uses reflection tokens to decide when to retrieve. CRAG (Yan et al., 2024) triggers corrective actions when retrieval confidence is low. SymRAG (Hakim et al., 2025) routes between symbolic and neural pathways. These systems route between retrieval depths or sources, not between retrieval *mechanisms*. Our question is which of the two mechanisms, similarity or graph-structural retrieval, suits a given question, and we test the entity-paper gap as a candidate signal for that, with a null result.

Query performance prediction. Predicting retrieval quality per query, without relevance judgments, is a long-standing line in information retrieval. Pre-retrieval and post-retrieval predictors were developed for query difficulty and for deciding when to apply a technique such as query expansion (Amati et al., 2004; Yom-Tov et al., 2005; Zhou & Croft, 2007), and the area has a standard reference treatment (Carmel & Yom-Tov, 2010) and a recent LLM-based line (Meng et al., 2024). The idea of using such a prediction to pick a retrieval strategy per query is also established: Arabzadeh et al. (2021) select between dense and sparse retrieval per query on an efficiency and effectiveness trade-off. Closest to our routing framing, Dong et al. (2026) route between vanilla RAG and GraphRAG per query using what they call a complexity score $s(q) \in (0, 1)$. A featurizer extracts syntactic, dependency, lexical and surface-semantic features from the query string alone: constituency-parse unit counts and ratios, dependency-tree statistics, and named-entity, lexical-diversity and question-type indicators. An MLP maps these to $s(q)$, trained as a binary classifier on the subset of training queries where the two pipelines disagreed in answer correctness and labelled by which of the two answered correctly, and two thresholds $\tau_L < \tau_H$ turn $s(q)$ into a hard three-way choice among dense retrieval, graph retrieval, and a fusion of the two.

Their signal and ours draw on disjoint information. Theirs is computed from the query string alone, with no access to the corpus, the index, or retrieval results; the entity-paper gap is computed entirely from the retrieval index and does not read the query's syntactic structure at all. Their features do count entities, but these are query-side named-entity counts with no reference to any entity in the index, whereas the entity-paper gap is a similarity against the KG's entity embeddings. Neither signal is computable from the other's features, so the two are complementary rather than competing and could in principle be combined. The practical difference is supervision: their scorer requires ground-truth answers for the training queries, a run of both pipelines over them to obtain the disagreement labels, and two tuned decision thresholds, whereas the entity-paper gap is closed-form over the index. We evaluate our signal against retrieval outcomes directly and report that it neither locates where the choice of modality matters nor beats the best fixed strategy (§4.4).

We position the entity-paper gap inside this line rather than outside it. Three things distinguish it. Its target is which *modality* to use rather than how hard a query is, so it selects between retrieval mechanisms over one corpus. It is closed-form over the retrieval index, one maximum over paper embeddings and one over entity embeddings, so it requires no training data, no supervision, and no tuned decision thresholds. And it is computable before the expensive strategies run, which is what would have made it usable as a gate on them. We do not compare against a trained query performance predictor, so we cannot say whether a standard predictor would locate the same questions as well or better. Such a comparison would inherit the difficulty we report in §4.4: on this benchmark the strategies return the same R@50 on most questions, so any predictor, ours or a trained one, is being asked to rank questions by the size of a difference that is mostly inside the noise floor.

Typed knowledge graphs over scientific literature. Typed entity graphs over research papers are an established resource, and we built one rather than using one. Linked Papers With Code (Färber & Lamprecht, 2023) exposes roughly 400,000 machine learning publications as an RDF graph with tasks, datasets, methods and evaluation results. CS-KG (Dessì et al., 2022) and AI-KG (Dessì et al., 2020) extract research entities and claims across computer science at comparable scale, and SciERC (Luan et al., 2018) provides the entity and relation schema that much of this line builds on.

For part of our graph an existing resource would have served. Counting on the sequential build, whose per-type breakdown is in Appendix C.5, Method and Task entities are 13,331 and 5,535 nodes, 18,866 together, which is 40.7% of its 46,304 entities, and those are exactly the types Linked Papers With Code already provides. Its largest entity type is Claim at 20,510 nodes, 44.3% of entities, and claims participate in 62,644 of 116,978 edges, 53.6%, through CLAIM_ABOUT_TOPIC and PAPER_MAKES_CLAIM. The primary batch build is larger, 79,109 entities over 166,734 edges, and adds a Dataset type, so its proportions differ, but the overlap with what already exists is of the same kind. We make no novelty claim for that schema either. CS-KG and AI-KG extract research entities and claims from computer science text, from the same group and the same lineage, so a claim-bearing typed graph over papers is established work. What this paper contributes is the retrieval-evaluation apparatus built over such a schema, a matched comparison of retrieval modalities against shared ground truth, rather than the schema itself. The reason we built a corpus rather than adopting one is the ground truth: every paper cited by a source survey has to be present, so the corpus is fixed by the ground-truth requirement rather than chosen from an existing graph. Linked Papers With Code is derived from Papers With Code and inherits its coverage, which is organised around published implementations and results rather than around any particular survey bibliography.

That decision costs 8,219 papers against roughly 400,000, and an LLM extraction pass over every one of them, 1.4 hours with the batch pipeline and roughly 70 with the sequential one. Validating our findings on Linked Papers With Code as an external graph, with its own typed entities and no involvement from our extraction pipeline, would test generalization without our construction in the loop. We have not run it.

Retrieval blind spots. Taghavi et al. (2026) define blind spots as entities relevant to a query but with low embedding similarity, and propose document augmentation as a remedy. We observe the same phenomenon but propose a different solution: using existing graph structure to bridge the gap when the entity-paper gap is high.

Retrieval benchmarks. NQ (Kwiatkowski et al., 2019) and MS MARCO (Bajaj et al., 2016) provide no graph structure. WebQSP (Yih et al., 2016) pre-assumes graph structure. CRAG (Yang et al., 2024) and STaRK (Wu et al., 2024) provide both modalities but do not isolate the retrieval mechanism. KGRAF-BENCH addresses these gaps with a corpus where both similarity and graph based retrieval are applicable and question sets designed to vary in graph dependence.

Scientific literature retrieval. Scientific paper discovery is a core task in research workflows, and recent systems demonstrate growing demand for retrieval over scientific corpora. PaperQA2 (Skarlinski et al., 2024) achieves superhuman performance on scientific literature synthesis using RAG over full-text articles. Ai2 ScholarQA (Singh et al., 2025b) provides organized literature synthesis with attribution over the Semantic Scholar corpus (Kinney et al., 2023). LitSearch (Ajith et al., 2024) introduces a retrieval benchmark for scientific literature search using inline citations and author-written queries. SciNet (Shao et al., 2025) evaluates relation-aware retrieval agents over a meta-database of 269M papers spanning 7 disciplines, with 8,940 tasks. OpenAI Deep Research (OpenAI, 2025) and Karpathy (2026) further demonstrate the trend toward autonomous research

agents that depend on high-quality literature retrieval. These systems use flat similarity search; none evaluate whether graph-structured retrieval over a scientific knowledge graph improves paper discovery. KGRAF-BENCH provides a testbed for this question.

B IMPLEMENTATION DETAILS

Embedding model. All node embeddings (papers and entities) and question embeddings are computed using `sentence-transformers/all-MiniLM-L6-v2` (Wang et al., 2020), a 384-dimensional model trained on over 1 billion sentence pairs. Embeddings are L2-normalized, so cosine similarity reduces to a dot product. We use a singleton instance with an in-memory cache to avoid recomputation.

LLM. All LLM calls, entity extraction, phrase extraction, canonicalization decisions, and question generation, use Claude Sonnet 4.5 via Amazon Bedrock (`us.anthropic.claude-sonnet-4-5-20250929-v1:0`), with temperature 0 and a maximum of 4,096 output tokens. Retries use adaptive backoff (up to 3 attempts).

Graph database. The knowledge graph is stored in NebulaGraph, an open-source distributed graph database. We use a single space with tagged vertices (`paper`, `entity`) and typed edges. The connection pool is configured with up to 10 concurrent sessions. The graph traversal queries in S_3 and S_4 use nGQL, NebulaGraph's native query language.

Retrieval parameters. S_1 and S_2 draw candidates from a nearest-neighbour search over the node index at depth $10K$, which is 500 at the reported budget of $K = 50$. S_3 and S_4 take $n = 5$ seed papers from a depth-200 search, expand to the 2-neighbourhood of those seeds, and match question phrases to entities at $\tau_e = 0.7$. S_4 applies the two entity bonuses of Equation (1) with the shared weight $\lambda = 0.1$; S_3 ranks the identical candidate pool by $\mathrm{sim}(q, p)$ alone. The channel (a) fallback that pads a short candidate list carries each padded paper's own cosine similarity and is never given an entity bonus, so a padded paper cannot outrank a traversal candidate of equal similarity. Canonicalization uses $\tau_c = 0.85$ except where the threshold ablation varies it (Appendix I.2).

Compute. Both construction pipelines require approximately 2 LLM calls per paper (extraction plus cross-entity edge validation) and canonicalize by embedding similarity against an index of existing nodes. The batch pipeline that builds the primary graph takes 1.4 hours on the 8,219-paper corpus and 45 hours on the 89,476-paper extension; the sequential pipeline used as the schema contrast processes one paper at a time against all previously extracted entities and takes approximately 70 hours on the same 8,219 papers. Worker layout, batching and per-stage timings for both are in Appendix I.1. Retrieval evaluation (all 4 strategies on 647 questions) takes 10.7 hours of summed per-question wall time, 4.4 hours on the 274 Standard questions and 6.3 hours on the 373 Relational questions. S_3 and S_4 dominate that total: each makes exactly one LLM call per question. On the sequential graph they take 26 to 29 seconds against 4.0 for S_2 and 0.01 for S_1; on the primary configuration the same strategies take 8 to 11 seconds against 0.02 for both S_1 and S_2. Per-question wall times are recorded per configuration in the released results.

C BENCHMARK CONSTRUCTION DETAILS

C.1 SOURCE PAPER SELECTION CRITERIA

The 7 source papers were selected based on three criteria: (i) high citation count relative to publication date, indicating community recognition; (ii) recency (2022–2025), ensuring the cited papers reflect the current state of the field; and (iii) topical coverage of the intersection between knowledge graphs, retrieval-augmented generation, and large language models, the domain where graph-augmented retrieval is most actively studied. Six are surveys providing comprehensive coverage of their respective fields; one (Edge et al.) is a seminal methods paper whose related work section provides expert citations for the GraphRAG domain. The source papers were identified by the authors based on familiarity with the field and confirmed via Semantic Scholar citation counts. We deliberately included papers from adjacent sub-areas (transformers for time series, lifelong learning) to ensure the

corpus is not limited to a single research community and to provide topical diversity in the question sets.

C.2 Question Examples

Figure 1 in the main text shows a Standard question with its source paragraph. Figure 3 is the Relational counterpart, generated from a pair of paragraphs, and Figure 4 collects further cases of both types across source papers.

Relational question (Jin et al. LLMs on Graphs survey (Jin et al., 2023), §4 Pure Graphs + §4.1 Direct Answering)

Source paragraph 1 (§4): "[...] input graphs are serialized as part of the input sequence, either by verbalizing the graph structure or by encoding the graph structure into implicit feature sequences. The studied reasoning problems range from simpler ones like connectivity, shortest paths, and cycle detection to harder ones like maximum flow and Hamiltonian pathfinding (an NP-complete problem)."

Source paragraph 2 (§4.1): "Although graph-based reasoning problems usually involve complex computation, researchers still attempt to let language models directly generate answers from the serialized input graphs [...] most studies find that LLMs possess preliminary graph understanding ability, but the performance is less satisfactory on more complex problems or larger graphs where reasoning is necessary."

Generated question: *What types of graph reasoning problems have been used to benchmark the limitations of direct answer generation approaches with LLMs, and what complexity characteristics make these problems challenging?*

Ground truth (4 papers): Liu & Wu (2023); Talk like a Graph (Fatemi et al., 2023); GPT4Graph (Guo et al., 2023); LLM4DyG (Zhang et al., 2023)

Figure 3: A Relational question with the two source paragraphs it was generated from. The generator sees both paragraphs and the survey's title and summary, and neither the titles nor the abstracts of the cited papers; the ground truth is the union of those citations. Citations inside the quoted source text are replaced with [...].

C.3 Source Papers

KGRAF-Bench is grounded in 7 ML/AI source papers (6 surveys + 1 methods paper) that collectively cite 724 unique papers (Table 5).

Table 5: Source papers for KGRAF-Bench. Each source paper provides ground-truth relevance judgments: papers cited by the source are relevant to the topics it discusses.

Source paper	Year	Papers cited
Graph Retrieval-Augmented Generation (Peng et al., 2024)	2024	150
Large Language Models on Graphs (Jin et al., 2023)	2023	198
KG Construction with LLMs (Zhu et al., 2023)	2023	69
Agentic RAG (Singh et al., 2025a)	2025	43
Lifelong Learning of LLMs (Zheng et al., 2024)	2024	225
Transformers in Time Series (Wen et al., 2022)	2022	59
GraphRAG (Edge et al., 2024)	2024	46
Total citation incidences		790
Total unique cited papers		724

C.4 Corpus Layer Construction

Layer 1. For each source paper, we retrieve all cited papers via the Semantic Scholar API, collecting paper ID, title, abstract, year, venue, and citation count. Papers cited by multiple source papers appear once. This yields 724 unique papers.

Standard question (Peng et al. GraphRAG survey (Peng et al., 2024), §2.2 LLMs on Graphs)

Source paragraph: "[LLMs on Graphs] explores the synergy between LLMs and graph-structured data, covering how LLMs can be applied to graph-related tasks and how graph structure can enhance LLM capabilities [...]"

Generated question: *What are the fundamental limitations of large language models when processing graph-structured or non-Euclidean data compared to their text processing capabilities?*

Ground truth (3 papers): Large Language Models and Knowledge Graphs: Opportunities and Challenges; Unifying Large Language Models and Knowledge Graphs: A Roadmap; Can Language Models Solve Graph Problems in Natural Language

Standard question (Peng et al. GraphRAG survey (Peng et al., 2024), §2.2 LLMs on Graphs)

Generated question: *What approaches have been developed to combine large language models with graph neural networks for improving performance on graph-based tasks like node classification and link prediction?*

Ground truth (2 papers): Efficient Tuning and Inference for Large Language Models on Textual Graphs (ENGINE); Bridging Local Details and Global Context in Text-Attributed Graphs

Relational question (Zheng et al. Lifelong Learning survey (Zheng et al., 2024), cross-section pair)

Generated question: *How do architecture-based approaches for continual instruction tuning compare to dynamic resource allocation methods used in multilingual neural machine translation, and what common principles do they share for preventing catastrophic forgetting?*

Ground truth: 39 papers across two source paper paragraphs

Relational question (Zheng et al. Lifelong Learning survey (Zheng et al., 2024), cross-section pair)

Generated question: *How do the common techniques used for continual relation extraction compare to those used for continual machine translation?*

Ground truth: 35 papers across two source paper paragraphs

Relational question (Zheng et al. Lifelong Learning survey (Zheng et al., 2024), within-section pair)

Generated question: *How do different replay-based approaches for continual learning in language models differ in their architectural requirements and memory mechanisms?*

Ground truth (9 papers): Including LAMOL, LFPT5, and hippocampal memory-based approaches

Figure 4: Additional question examples from KGRAF-BENCH across different source papers and question types.

Table 6: Corpus layers in KGRAF-BENCH. L0 source papers provide ground-truth labels; L1–L3 form the retrieval corpus.

Layer	Source	Papers	Role
L0	Source papers (6 surveys + 1 methods)	7	Ground-truth labels
L1	Cited by L0 source papers	724	Ground-truth relevant
L2	Papers citing an L1 paper	3,514	Hard distractors
L3	Keyword-venue search	3,981	Corpus scale
	Total (L1–L3)	8,219	

Layer 2. For each of 538 L1 papers used as seeds, we retrieve papers that cite that seed via the Semantic Scholar API, filtering out surveys and papers with abstracts shorter than 50 characters, and excluding anything already in L1. No venue filtering is applied. This yields 3,514

papers, a median of 4 per seed and at most 38. Every L2 paper records which seed it came from, as `citation_neighbor_of` in `corpus_annotations.json`. These are *hard distractors*: they cite the same work a source paper cited, so they share its citation context without being cited by any source paper themselves. The released `construction/build_corpus.py` samples 10 seeds per source paper and keeps at most 20 citers of each, which is a narrower sweep than the one that produced this corpus; a rebuild from the shipped code will therefore give a smaller L2.

Layer 3. We search the Semantic Scholar API using topic keywords extracted from survey titles combined with venue and year filters. Venues are restricted to NeurIPS, ICML, KDD, and ICLR; years to 2021–2024. We also include fixed queries for core topics ("transformer," "knowledge graph," "retrieval augmented," "graph neural," "attention mechanism," "language model"). Results are sorted by citation count and surveys are excluded. This yields 3,981 papers providing corpus scale and ensuring retrieval systems must discriminate among a large pool of topically plausible candidates.

C.5 Knowledge Graph Schema and Statistics

Table 7 summarizes the node and edge types, and Table 8 reports the statistics of both graphs. One naming note: the entity type we call CLAIM throughout the paper is stored as `Thesis` in the released graphs and in the extraction prompts, and is reached by the PAPER_MAKES_CLAIM edge. The two names denote the same type; we use CLAIM in the text because it is the more transparent label.

Table 7: Knowledge graph schema. Paper→entity edges are deterministic; cross-entity edges are LLM-validated. The batch-constructed graph used for the primary evaluation emits all types listed; the sequential graph used as the schema contrast in §4.2 omits the DATASET type and the PAPER_ADDRESSES_TASK and PAPER_USES_DATASET edges, marked $*$ below.

Node types		
PAPER		Source document
METHOD	(0–2 per paper)	Techniques, algorithms
TASK	(0–2 per paper)	Problems, objectives
TOPIC	(1–3 per paper)	Research sub-areas
CLAIM	(1–3 per paper)	Findings, contributions (`Thesis` in the data)
DATASET*	(0–2 per paper)	Evaluation corpora
Edge types		
PAPER_USES_METHOD	Paper → Method	Deterministic
PAPER_ABOUT_TOPIC	Paper → Topic	Deterministic
PAPER_MAKES_CLAIM	Paper → Claim	Deterministic
PAPER_ADDRESSES_TASK*	Paper → Task	Deterministic
PAPER_USES_DATASET*	Paper → Dataset	Deterministic
METHOD_ADDRESSES_TASK	Method → Task	LLM-validated
CLAIM_ABOUT_TOPIC	Claim → Topic	LLM-validated

C.6 Knowledge Graph Construction Pipeline

Algorithm 1 summarizes the sequential EDC pipeline, which canonicalizes each paper's entities against every entity extracted before it and emits the four-type schema. It is the schema contrast of §4.2; the primary configuration uses the batch pipeline of Appendix I.1, which emits five types.

Papers are processed sequentially. A checkpoint file saves progress every 50 papers, enabling resumption after interruption. One paper out of 8,219 failed during extraction (empty abstract after preprocessing), yielding a KG with 8,218 paper nodes.

Example. For the paper "UniKGQA: Unified Retrieval and Reasoning for Solving Multi-hop Question Answering Over Knowledge Graph" (Jiang et al., 2023), the extraction step produces:

- METHOD: "UniKGQA" (unified retrieval and reasoning framework)
- TASK: "Multi-hop KBQA" (answering questions requiring multiple reasoning steps over a KG)

Table 8: Statistics of both 8,219-paper graphs. Shared entities (appearing in 2 or more papers) are the cross-paper connections graph retrieval can exploit. The batch graph is the primary evaluation; the sequential graph is the schema contrast of §4.2. Note that the batch graph has more entities but proportionally fewer shared ones, so its advantage for traversal does not come from extra bridging (§4.2). Both graphs cover 8,218 papers, so per-paper rates follow by division.

	Batch (primary)	Sequential
Paper nodes	8,218	8,218
Entity nodes	79,109	46,304
of which shared by $\geq$2 papers	4,185 (5.3%)	6,399 (13.8%)
of which linked to no paper	3	5,566 (12.0%)
Entity types	5	4
Total edges	166,734	116,978
Edge types	7	5
Build time	1.4 h	70 h
Entity nodes by type		
Claim	23,003	20,510
Method	20,074	13,331
Topic	16,704	6,928
Task	12,935	5,535
Dataset	6,393	absent
Edges by type		
CLAIM_ABOUT_TOPIC	44,237	39,010
METHOD_ADDRESSES_TASK	33,526	18,471
PAPER_MAKES_CLAIM	23,153	23,634
PAPER_ABOUT_TOPIC	21,612	20,618
PAPER_USES_METHOD	20,392	15,245
PAPER_ADDRESSES_TASK	14,492	absent
PAPER_USES_DATASET	9,322	absent

Algorithm 1 Passive EDC Knowledge Graph Construction

Require: Corpus $\mathcal{C} = \{p_1, \ldots, p_n\}$, LLM $\mathcal{L}$, embedding model ϕ, similarity threshold $\tau_c = 0.85$
Ensure: Knowledge graph $G = (V_P \cup V_E, T)$

```
Initialize V_E ← ∅, T ← ∅
for each paper p_i ∈ C do
    Extract: E_i ← L(title(p_i), abstract(p_i)) {Methods, Tasks, Topics, Claims}
    Remove intra-paper duplicates: if cos(φ(e_j), φ(e_k)) > τ_c for e_j, e_k ∈ E_i, keep one
    for each entity e ∈ E_i do
        Canonicalize: Find top-5 candidates C_e = {e' ∈ V_E : type(e') = type(e), cos(φ(e), φ(e')) > τ_c}
        if C_e ≠ ∅ and L("match?", e, C_e) = yes then
            Reuse existing node: e ← e'_matched, enrich description
        else
            Create new node: V_E ← V_E ∪ {e}
        end if
        Define: Add typed edge (p_i, e) to T
    end for
    Cross-entity edges: L validates Method→Task and Claim→Topic relations; add to T
end for
```

- TOPIC: "Knowledge Base Question Answering"
- CLAIM: "Unifying retrieval and reasoning improves multi-hop QA performance"

During canonicalization, "Knowledge Base Question Answering" matches an existing Topic node (shared by 12 other papers), creating a cross-paper connection. The Method "UniKGQA" is new and creates a fresh node. The edge METHOD_ADDRESSES_TASK(UniKGQA, Multi-hop KBQA) is validated by the LLM in the cross-entity edge step.

C.7 QUESTION SETS

§2.2 defines the two sets and the bias each carries; Table 9 restates their sizes for reference. Two further splits are recorded here rather than in the main text. Within the Standard set, partitioning on the number of corpus papers cited by the source paragraph, not on ground-truth set size, gives 95 *local* questions from paragraphs citing at most three and 179 *global* ones from paragraphs citing four or more. Within the Relational set, 200 questions come from paragraph pairs and 173 from whole sections, and the released manifest records the split as `n_pair` and `n_section`. The two routes differ in ground-truth size, since a section cites more papers than a pair of paragraphs does, and the Relational set's range reflects that: 3 to 67 ground-truth papers, mean 20.7, median 15. Neither split is used as a reporting axis in the main text; we record them because the released question records carry the labels.

Table 9: Question sets in KGRAF-BENCH.

Set	Questions	LLM sees	Bias
Standard	274	Paragraph + cited abstracts	Toward flat RAG
Relational	373	Paragraph pair or section, no cited text	Toward graph
Total	647		

C.8 QUESTION GENERATION

Standard question generation. We parse each source paper's LaTeX source into a structured representation of sections and paragraphs. An LLM first summarizes the source paper's overall goal and each section's role (Pass 1). Then, for each citation-dense paragraph (≥ 2 cited corpus papers), the LLM generates questions given the full context: source paper summary, section description, preceding and following paragraph summaries, the paragraph text, and cited paper abstracts truncated to 500 characters (Pass 2). The number of questions per paragraph scales with the citation count. Each question records its ground-truth paper IDs (the papers cited in the source paragraph) and distinguishes essential from background citations.

Relational question generation. Two routes feed this set. The *pair* route identifies pairs of paragraphs in the same source paper that share cited papers, which is the survey author having connected them thematically, and prompts for 1–2 questions about the relationship between the work the two discuss; it yields 200 questions in two templates, **R1 (Comparison)** where the paragraphs sit in one section and discuss different methods for the same task, and **R2 (Transfer)** where they sit in different sections and share a method, so the question asks how that method carries across tasks. The *section* route takes a whole section citing many papers and prompts for 1–3 **R3 (Synthesis)** questions about the landscape of work in it, the categories of method, the trends across papers, or the gaps; it yields the remaining 173.

Both prompts supply the survey title and goal, the section names and the paragraph or section text, and *neither* supplies the titles or the abstracts of the cited papers, so question vocabulary is not drawn from the text the retrieval index is built over. Both also instruct the LLM not to name any paper in the question and to phrase it as a researcher would before reading the survey. Ground truth is every corpus paper the source text cites: the union over the two paragraphs on the pair route, and the section's citations on the section route. Neither route runs the essential/background classification, so no Relational ground-truth set is narrowed.

C.9 GROUND TRUTH VERIFICATION

We verify ground truth quality through several checks:

Reference chain validation. All ground-truth paper IDs are confirmed present in the corpus (0 missing references). For the GraphRAG survey, we additionally cross-checked against the Semantic Scholar API's reference endpoint: 81.4% of ground-truth papers were confirmed, with the 18.6% gap attributable to known S2 limitations with recent preprints.

Essential vs background classification, and how far it actually reaches. The question generation LLM is asked, as a third step of the same prompt that produces the question, to classify each cited paper as essential (needed to answer the question) or background (cited for context). Where the classification returns a non-empty essential list, ground truth is the essential papers; where it returns nothing, the generator falls back to every resolved citation in the source paragraph. The intent is to prevent foundational papers cited ubiquitously (e.g., "Attention is All You Need") from inflating ground-truth sets.

Measured against the released question sets, the reach of this step is limited, and we report it because it bounds how much of the ground truth is LLM-mediated:

- **Standard (274 questions).** An essential list is present for 271, but it is a strict subset of the cited papers on only **59**, removing **77** papers in total. On 212 of the remaining 215 the LLM marked every cited paper essential, and on the last 3 it returned no list at all and the generator fell back to every resolved citation; in both cases ground truth is the citation list unchanged. Three questions fell through to the fallback.
- **Relational (373 questions).** The essential and background fields are **empty for all 373**. The classification never ran on this set, and every ground-truth set is the union of all papers cited by the two source paragraphs.

Across the benchmark, the LLM therefore narrows ground truth on **59 of 647 questions (9.1%)**; for the remaining 588 it is the source authors' citation list verbatim. Two consequences follow. First, the *expert-grounded* characterization is accurate: an LLM can only drop papers the authors chose to cite, never introduce one they did not, and on 90.9% of questions it drops none. No LLM scores or ranks retrieved results anywhere in the evaluation. Second, the absence of filtering on the Relational set is one mechanism behind the nine-fold difference in ground-truth set size between the two question sets (20.7 papers against 2.2), which we discuss as an alternative explanation for the regime split in §5.

Layer promotion. During verification, 21 papers used as ground truth were found in L2/L3 rather than L1. They were genuinely cited by source papers (confirmed in LaTeX `.bib` files) but missed by the S2 API during corpus construction. These were promoted to L1 to ensure all ground-truth papers are in the correct layer.

Manual spot-check. The citation-as-relevance assumption was checked by hand on a sample of 50 questions; Appendix K reports that audit in full.

Contamination analysis. The closed-book result of §2.3 has an open-book counterpart. Presented with the ground-truth papers mixed into a candidate list of random distractors, the same model reaches 69.1% F1, against 12.7% with no retrieval at all. It can therefore recognise the relevant papers when shown them and cannot recall them from parametric memory.

D Full Per-Question Results

D.1 All Metrics, All Strategies

Table 10 gives every metric for every strategy at 8K scale, including R@20, which the main text does not report for space.

Table 10: Complete retrieval metrics on the 8,219-paper corpus, batch-constructed graph at $\tau_c = 0.85$ ($K = 50$). Same runs as Table 1, with R@20 added.

	Standard (274 Qs)				Relational (373 Qs)			
Strategy	R@10	R@20	R@50	MRR	R@10	R@20	R@50	MRR
S_1 Flat RAG	.341	.432	.531	.271	.074	.119	.199	.294
S_2 Hybrid	.354	.426	.548	.289	**.081**	**.137**	**.231**	.295
S_3 Entity-aware	.376	**.464**	.571	.276	.078	.123	.204	.295
S_4 Hybrid-scored	**.383**	.457	**.573**	**.296**	.077	.123	.206	**.299**

D.2 Win-Rate Analysis

Table 11 reports per-question win rates at R@50 on the 8K corpus.

Table 11: Strategy win rates at R@50 (8K corpus). A strategy wins a question when it attains the maximum R@50 on it, so a tied strategy counts as a winner and the rates sum to >100%.

Strategy	Standard	Relational
S_1 Flat RAG	78%	50%
S_2 Hybrid	80%	**57%**
S_3 Entity-aware	82%	55%
S_4 Hybrid-scored	**83%**	53%

On Standard questions, win rates range from 78% to 83%, a narrow spread driven by high tie rates (215 of 274 questions are ties between S_4 and S_1). On Relational questions, the spread is wider (50%–57%) and S_2 leads. The strategies diverge only on a subset of questions, namely those where the four do not all return the same R@50: 115 of 274 Standard questions (42.0%) and 309 of 373 Relational questions (82.8%). Of those, exactly one strategy is strictly best on 34 Standard and 181 Relational questions, and the rest are partial ties.

D.3 Scale Comparison: 8K vs 89K

Table 12 compares strategy performance across corpus scales on both question sets. All strategies achieve lower recall at 89K, because an 11× larger corpus puts more distractors inside the top-50 budget, but the degradation is not uniform across strategies.

Table 12: Full scale comparison (R@50), with the drop from 8K to 89K, discussed below. Both columns are batch-constructed at $\tau_c = 0.85$, so corpus size is the only factor that changes.

	Standard (274 Qs)			Relational (373 Qs)		
Strategy	8K	89K	Δ	8K	89K	Δ
S_1 Flat RAG	.531	.457	−.073	.199	.148	−.051
S_2 Hybrid	.548	.445	−.103	.231	.168	−.063
S_3 Entity-aware	.571	.481	−.090	.204	.155	−.049
S_4 Hybrid-scored	.573	.484	−.089	.206	.155	−.051
$S_4 - S_1$	+.042	+.026		+.007	+.007	
$S_3 - S_1$	+.041	+.024		+.005	+.006	
$S_2 - S_1$	+.017	−.012		+.032	+.020	

Scale narrows every gap on Standard questions and leaves the Relational ones where they are. Three of the six movements exceed the 0.015 floor of Appendix I.6, all on Standard: $S_2 - S_1$ inverts from +.017 to −.012, and the two traversal gaps narrow by .017 and .016. On Relational the three move by .012, .001 and .000, all inside the floor. So an 11× larger corpus costs every strategy recall, and it costs the graph strategies more of their margin than flat retrieval on the question set where flat retrieval was already sufficient. No gap grows with scale.

E Regime Threshold Sensitivity

The trained router of §4.4. The router is a ridge regression predicting the $S_4 - S_1$ difference in R@50 from seven features available before retrieval: the two similarity maxima, their difference, the number of entities above cosine 0.7, two question-keyword flags and question length. Features are standardised, the model is fit under five-fold cross-validation with tied questions dropped from the training folds but retained when scoring, routing goes to S_4 where the predicted difference is positive and to S_1 otherwise, and the comparison against the best fixed strategy is bootstrapped over questions with 10,000 resamples. The two similarity maxima correlate at $r = +0.67$ on both question sets, so the gap is a thin signed residual and not an independent quantity.

Contrasts that inflate the separation. §4.4 reports the gap between the 22 Relational questions S_4 rescues from flat retrieval's failures and the 64 it does not, $d = +0.23$ [-0.25, $+0.82$], which spans zero. Two looser contrasts do reach significance, but each compares groups themselves defined by which strategy won: $d = +0.89$ [$+0.40$, $+1.37$] for the rescued questions against the 48 where flat retrieval wins outright, and $d = +0.44$ [$+0.24$, $+0.60$] across the 309 Relational questions on which the strategies diverge. Selecting a comparison group on retrieval outcome selects on $\max_p \mathrm{sim}(q, p)$, which the gap subtracts, so those effects are definitional. The only contrast that does not select on outcome is the null one. On Standard questions all three are null.

Single-feature thresholds. We evaluate how well a single feature discriminates between S_1 and S_4 on Relational questions (8K corpus), by thresholding on it and selecting the better strategy on each side. For each numeric feature, we sweep thresholds at the 25th, 50th, and 75th percentiles and select the configuration maximizing R@50.

The best single-feature router uses $\max_p \mathrm{sim}(q, p)$ with threshold 0.752: route to S_1 when above, S_4 when below. This achieves R@50 = 0.215 with routing accuracy 79%, matching S_2 but not exceeding it. On Standard questions, the best router uses sim spread with threshold 0.038, achieving R@50 = 0.564 (+0.002 over S_4).

This single-feature analysis was run on the sequential graph, before the batch graph became the primary configuration, so its numbers are not directly comparable with §4.4; we report it as the predecessor of the trained router there. On that graph the oracle achieves R@50 = 0.267 on Relational and 0.652 on Standard, against .273 and .663 on the primary configuration. The gap between the best single-feature threshold and the oracle is $+0.052$ on Relational and $+0.088$ on Standard. That gap is a ceiling computed from the realised outcome rather than a reachable target: §4.4 trains a router on pre-retrieval features and it does not beat the best fixed strategy.

F PHYSICS EXTENSION

A third instantiation, built with the released toolkit from 7 physics surveys spanning 7 arXiv primary archives. It is the only extension large enough to resolve effects at the size the ML sets report.

Corpus and questions. 11,714 papers in the same three layers: 3,354 cited by the surveys, 4,338 co-citation distractors and 4,022 from physics-venue keyword sampling across 14 venues validated against the Semantic Scholar venue filter. All abstracts present, median 1,267 characters, median year 2024, no paper before 1990. The source surveys are excluded by resolved paper identifier, verified as 0 of 7 present and 0 ground-truth papers across all questions. Question generation yields 1,001 Standard and 682 Relational after removing 15 duplicates and 21 questions whose ground truth exceeded the ML maximum of 31 Standard and 67 Relational; over-cap questions are dropped rather than truncated, since truncating would invent a subset and misstate what the question asks. A per-survey cap of 300 keeps any one survey from dominating. Every ground-truth identifier resolves into the corpus.

Graph. 89,225 entities over 163,866 edges at $\tau_c = 0.85$ with MiniLM, the same settings as the primary configuration. No feature row is corrupted on either question set, on a graph where entities outnumber papers 7.6:1, because S_1 here queries the paper index directly.

Table 13: Physics extension, R@50 and paired differences against S_1 (1,000 resamples, seed 42). † marks an interval excluding zero.

	S_1	S_2	S_3	S_4	$S_2 - S_1$	$S_3 - S_1$	$S_4 - S_1$
Standard (1,001 Qs)	.493	.512	.493	.496	$+.019^{\dagger}$	$-.000$	$+.003$
Relational (682 Qs)	.298	.327	.323	.325	$+.028^{\dagger}$	$+.025^{\dagger}$	$+.026^{\dagger}$

Entity sharing. Only 1.82% of physics entities are shared by two or more papers, 1,622 of 89,225, against 4.19% for the biomedical graph, 5.29% for the primary batch graph and 13.82% for the sequential one. Shared entities are the only mechanism by which traversal reaches a paper flat

similarity misses, so a corpus with the largest entity index and the least sharing offers the least to traverse. Mean entity-paper gap is positive on both sets, $+.061$ and $+.068$, so the corpus does have the hard-distractor composition the domain argument predicts; what it lacks is cross-paper structure. We read this as evidence that entity sharing rather than distractor presence is the operative property, though with four corpora that is an inference and not a measured relationship.

The entity-paper gap does not discriminate here either. Cohen's d between S_4-wins and S_1-wins on divergent questions is $+0.118$ on Standard and -0.056 on Relational, the second with the sign inverted. This is an independent replication of the null in §4.4.

Caveats. The 1,683 questions are LLM-generated and have not been author-validated, unlike the 647 ML questions (§2.2); they should be treated as unaudited. Wall-clock times are not comparable to the other configurations because the evaluation ran at 16 workers, so per-strategy timings include thread contention; recall and MRR are unaffected. One paper's entities were lost to a mid-write crash and skipped on resume, and 3,231 duplicate entity lines across the resume boundary inflate the file without affecting results, since the store deduplicates by identifier on load.

G Limitations and Future Work

This section expands the four limitations of §5 and states three more that the main text does not have room for.

Domain scope. The biomedical instantiation is a second literature, not a disjoint one: 45 of its 430 ground-truth papers are also in the ML corpus. Its 111 questions are all Standard-type, so the extension tests the half of the benchmark where similarity is expected to suffice and never tests the two-regime finding at all. Reaching significance at the effect size we observe needs roughly 350 questions, which the pipeline can generate from further source reviews.

KG construction. The traversal advantage holds across five canonicalization thresholds, three encoders and two construction pipelines. What remains untested is a fundamentally different entity schema, and the two builds compared in §4.2 differ in schema and in merge order simultaneously, so neither factor is attributed. A third build, the batch schema canonicalized sequentially, would separate them; we did not construct one.

Question quality. Both question sets are LLM-generated. An expert audit by the first author passes 49 of 50 sampled questions on all four criteria and judges the ground-truth papers relevant in all 50, but with one annotator there is no agreement statistic; two further annotators will rate the same sample for the camera-ready version. Every question ships with the source paragraph it was generated from, so the audit can be extended or contested.

Shared extraction. One model performs entity extraction, canonicalization, phrase extraction and question generation (Appendix B), so the graph's entity names and a question's extracted phrases come from the same source. Retrieving flat with S_3's own extracted phrases separates that sharing from the traversal, and the phrases do not retrieve on their own: $-.045$ [$-.082$, $-.010$] against S_1 on Standard questions and $-.004$ [$-.017$, $+.010$] on Relational (Appendix I.4). That rules out query expansion as the explanation, but not the sharing itself, since phrase-to-entity matching still draws on one model's naming on both sides. The Relational set is generated without the cited abstracts in view and S_2 makes no query-time call, so the Relational S_2 result is the least exposed to this.

Clustered questions. All 647 questions derive from 7 source papers, and questions from one source share its bibliography and phrasing, so the intervals of Appendix I.5 treat clustered observations as independent. Resampling source papers instead leaves every significant result significant, and with 7 clusters that check is itself underpowered.

No predictive baseline. We compare the entity-paper gap against $\max_p \mathrm{sim}(q, p)$ but not against a trained query performance predictor. The per-question features are released so that comparison can be run against our numbers.

Future work. Two extensions follow from the released pipeline. Instantiating the benchmark on further domains tests whether the two-regime split survives a different literature, which the biomedical extension does not settle. More decisively, generating question sets whose dependence on graph structure varies independently of ground-truth set size would separate the two explanations in limitation *(iii)*: a Standard and a Relational set matched on ground-truth size would show whether the split is structural or an artifact of how many papers each question has to retrieve. Two further items remain: evaluating a citation-trained encoder such as SPECTER2 or SciNCL, and releasing the retrieved set for each strategy.

H Practical Guidance

Based on the findings in §4, we draw the following practical conclusions for practitioners building scientific retrieval systems:

- **Prefer S_3 when the deployment conditions are not known in advance.** S_3 and S_4 are the two strategies whose direction held in every configuration we evaluated, and S_3 is positive in the most tests of any strategy, 52 of its 80. S_2 is the cheaper choice and the strongest of the three on Relational questions at 8K, at 0.02 seconds per question and no LLM call, but its sign depends on the condition: it falls below S_1 on Standard questions at 89K papers ($-.012$) and at canonicalization threshold 0.70 (.510 against .531). A system adopting S_2 should re-measure it at its own corpus size and with its own encoder.
- **Entity-aware strategies are competitive, not harmful.** S_3 and S_4 are significantly behind S_1 in none of their 80 paired tests, at both corpus scales and in the biomedical instantiation, contrary to the underperformance reports that motivate Xiang et al. (2025). S_2 does not share that record: it falls below S_1 on Standard questions at 89K and at canonicalization threshold 0.70, and its estimate is negative in 32 of its 80 tests.
- **The entity-paper gap is not a usable signal.** It does not separate the Relational questions S_4 rescues from those it does not, $d = +0.23$ [-0.25, $+0.83$], and it is null on Standard questions ($|d| \leq 0.14$). A router trained on it under cross-validation is worse than the best fixed strategy (§4.4), and a single-feature threshold sweep on the sequential graph reaches the same conclusion (Appendix E).
- **Do not attribute the graph's value to a single channel.** Which strategy separates from flat retrieval is a property of the graph rather than of the channel: on the batch graph S_3 and S_4 separate on Standard questions and S_2 on Relational, and on the sequential graph all three separate on Relational and none on Standard (§4.2). We report no pairwise test among the three graph strategies. Entity extraction quality and traversal are both load-bearing, and no contrast in this paper isolates either one.

I Ablation Studies

Each subsection below is one block of the grid summarised in Table 3, reported in full. All of them use the four retrieval strategies and the matched-budget protocol of §3 unchanged, and vary one factor of the primary configuration.

I.1 Batch vs Sequential EDC Construction

Worker layout and batching. The sequential pipeline processes one paper at a time and canonicalizes each extracted entity against every entity already in the graph, so its canonicalization set grows monotonically and every decision sees all prior ones. The batch pipeline replaces that with four phases per batch of 100 papers: extraction, run as 50 concurrent LLM calls over a thread pool with one call per paper; embedding and within-batch deduplication of near-identical mentions, embedded 256 at a time; canonicalization of the surviving entities against a cache of the graph as it stood when the batch began; and accumulation into the entity and edge records. Progress is checkpointed every 5 batches so a run can resume. The consequence is the one visible in the entity counts: within a batch, two papers extracting the same entity are reconciled by the dedup phase, but neither sees canonicalization decisions the other triggered, so the batch pipeline merges less aggressively than

the sequential one at the same threshold τ_c. Extraction, canonicalization and question-generation prompts are in the released construction code rather than reproduced here.

Table 14 compares the two on the same 8K corpus.

Table 14: Batch vs sequential EDC construction on 8K corpus. Strategy ordering is preserved despite structurally different graphs.

Method	Entities	Edges	Time	S1	S2	S3	S4	S4−S1
Sequential (paper)	46,304	116,978	∼70h	.531	.557	.556	.562	+.031
Batch EDC	79,109	166,734	1.4h	.531	.548	.571	.573	+.042

All three entity-aware strategies are above S_1 in both constructions, so the direction survives a structurally different graph; which of them separates significantly, and on which question set, does not, and §4.2 treats that as the result. The 50× speedup is what makes the 89K-scale evaluation affordable at 45 hours.

I.2 Canonicalization Threshold Ablation

We vary the canonicalization threshold across {0.70, 0.80, 0.85, 0.90, 0.95} at 8K scale using batch construction, rebuilding the graph at each value; Table 15 reports the result.

Table 15: Canonicalization threshold ablation (R@50, 8K corpus, batch EDC). S4−S1 is positive at all thresholds. Each threshold is a separate construction run, so S_1 is identical in every row, as it must be: it queries the paper index alone, and the rows differ only in the entity index. The $\tau_c = 0.85$ row is the primary configuration, so its eight scores are those of Table 10 and Table 1 and not a second measurement of them.

	Standard (274 Qs)						Relational (373 Qs)					
Threshold	Entities	S1	S2	S3	S4	S4−S1	S1	S2	S3	S4	S4−S1	S2−S1
0.70	54,241	.531	.510	.558	.577	+.046	.199	.208	.214	.219	+.020	+.009
0.80	73,550	.531	.556	.571	.584	+.054	.199	.221	.214	.215	+.016	+.022
0.85	79,109	.531	.548	.571	.573	+.042	.199	.231	.204	.206	+.007	+.032
0.90	82,156	.531	.552	.570	.576	+.045	.199	.232	.208	.212	+.013	+.033
0.95	83,579	.531	.561	.561	.567	+.036	.199	.224	.199	.203	+.004	+.025

$S_4 - S_1$ is positive at all five thresholds (+.036 to +.054 on Standard, +.004 to +.020 on Relational), and traversal never falls below flat retrieval anywhere in the sweep. S_2 does, at $\tau_c = 0.70$ (.510 against .531), where over-merging makes entity embeddings too broad to discriminate; on Relational questions it instead peaks at 0.90 (.232), where more distinct entities survive. The best setting for S_4 on Standard is 0.80 (.584), against .573 at the 0.85 the paper uses.

The entity-paper gap is not comparable across constructions. The epg is monotone in entity count, because a larger entity index gives every question a better-matching nearest entity while leaving $\max_p \mathrm{sim}(q, p)$ untouched. Across this sweep the mean epg rises from +.007 to +.046 on Standard and from +.021 to +.060 on Relational as $|V_E|$ goes from 54,241 to 83,579. That Relational shift of +.040 comes with no change to the questions, the ground truth or the retrieval outcome, so a movement of this size is attributable to entity count alone, and it is larger than the within-graph contrasts §4.4 rests on. Mean epg also rises with corpus scale on Relational questions. An epg value is therefore only interpretable against other questions scored on the same graph. It is a within-graph ranking signal, not an absolute quantity, and epg magnitudes should not be compared across canonicalization thresholds, construction methods or corpus sizes. This does not affect the per-question results, which are all computed within one graph, but it does mean a practitioner must calibrate any epg threshold on their own index.

Table 16: Embedding model ablation (R@50) on the primary batch graph. The graph is held fixed and only node embeddings change, so the MiniLM row is the configuration of Table 1. † marks a paired bootstrap interval excluding zero.

	Standard (274 Qs)						Relational (373 Qs)					
Model	S1	S2	S3	S4	S2−S1	S3−S1	S1	S2	S3	S4	S2−S1	S3−S1
MiniLM (primary)	.531	.548	.571	.573	+.017	+.041†	.199	.231	.204	.206	+.032†	+.005
BGE-large	.559	.582	.589	.585	+.023	+.030	.202	.235	.220	.221	+.033†	+.018†
Qwen3-0.6B	.580	.579	.610	.605	−.000	+.031†	.229	.235	.240	.239	+.006	+.011†

I.3 Embedding Model Ablation

Does a stronger encoder make the graph redundant? A concern with the primary evaluation is that the graph advantage might compensate for weak embeddings rather than providing genuine structural value. If better paper embeddings closed the gap, the graph would be redundant with better flat search. We re-embed every node with two stronger 1024d models, BGE-large-en-v1.5 (MTEB Retrieval 54.29) and Qwen3-Embedding-0.6B (MTEB English v2 Retrieval 61.83), holding the graph fixed at the primary build so that only the encoder changes. Flat retrieval does improve: S_1 rises from .531 to .559 and .580 on Standard questions, and from .199 to .229 under Qwen3 on Relational. The graph advantage does not close. On Relational questions traversal separates from flat retrieval under both stronger encoders where it does not under MiniLM, +.018 and +.011 against +.005; on Standard it separates under MiniLM and Qwen3 but not BGE-large, where the interval spans zero. The entity-similarity channel does not track encoder strength: its Relational gap is +.032 under MiniLM, +.033 under BGE-large and +.006 under Qwen3.

What a stronger encoder changes. On the primary graph, replacing MiniLM with Qwen3 shrinks the entity-similarity channel's gap on Relational questions by −.027 [−.042, −.012] and leaves traversal's unchanged, +.006 [−.009, +.022]. Under BGE-large neither channel moves significantly. Better paper embeddings therefore compete with the entity-similarity channel for the same signal, while graph neighborhood expansion reaches papers that share no vocabulary with the question and a stronger encoder does not recover those. On Standard questions no difference reaches significance under either encoder. This is an interpretation of the measurements and not a separate result, and we have not isolated it experimentally: S_3 differs from S_2 in three ways at once, adding traversal, phrase extraction and a different entity-matching rule, so no contrast here attributes the change to traversal alone.

We compute the change in each strategy's gap over S_1 directly, rather than differencing the two tables. For every question answered under both models we take $(S_x - S_1)_{\text{Qwen3}} - (S_x - S_1)_{\text{MiniLM}}$ and bootstrap over questions (1,000 resamples). A negative value means the strategy's advantage shrinks when embeddings improve. Table 17 reports this for all three construction settings.

Table 17: Change in each strategy's gap over S_1 when MiniLM is replaced by the stronger encoder named in each row, paired per question. Negative means the advantage shrinks under stronger embeddings. † marks an interval excluding zero.

Graph and encoder	$S_2 - S_1$	$S_3 - S_1$	$S_4 - S_1$
Standard (274 questions)			
8K sequential, Qwen3	+.003 [−.049, +.055]	−.001 [−.029, +.028]	−.014 [−.053, +.027]
8K batch 0.85, Qwen3	−.017 [−.058, +.023]	−.010 [−.054, +.033]	−.016 [−.061, +.028]
8K batch 0.85, BGE	+.006 [−.033, +.044]	−.010 [−.056, +.036]	−.016 [−.063, +.032]
89K batch, Qwen3	−.003 [−.049, +.042]	−.009 [−.048, +.029]	−.005 [−.044, +.035]
Relational (373 questions)			
8K sequential, Qwen3	−.019 [−.036, −.003]†	+.001 [−.012, +.014]	−.012 [−.027, +.002]
8K batch 0.85, Qwen3	−.027 [−.042, −.012]†	+.006 [−.009, +.022]	+.003 [−.014, +.019]
8K batch 0.85, BGE	+.001 [−.014, +.016]	+.013 [−.004, +.030]	+.012 [−.005, +.030]
89K batch, Qwen3	−.017 [−.031, −.003]†	+.014 [−.001, +.028]	+.010 [−.005, +.025]

Three cells reach significance, and all three are the same one: $S_2 - S_1$ shrinks on Relational questions when MiniLM is replaced by Qwen3, on the sequential graph, on the batch graph and at 89K. That is the entity-similarity channel giving ground as paper embeddings improve, which is what competing for the same signal would predict. No traversal cell reaches significance in any setting, and on Standard questions nothing does. We therefore claim no effect of encoder strength on the traversal channel in either direction.

I.4 Additional Baselines: BM25 and PPR

We evaluate BM25 (lexical retrieval) and Personalized PageRank (Haveliwala, 2002; Tong et al., 2006) following HippoRAG (Gutiérrez et al., 2024) (graph-global diffusion seeded from top-5 entities by cosine similarity) on the 8K corpus. One caveat about matching. BM25-papers reads no graph, so it is comparable to any row. BM25-KG and PPR were run on the sequential graph, while the S_1–S_4 rows are the primary batch configuration, so the structural control is not matched to the graph the entity-aware strategies traverse. S_1 is unaffected, since it reads no graph either.

Table 18: Additional baselines (R@50, 8K corpus), referenced from §4.1. The S_1–S_4 rows are the primary batch configuration; BM25-KG and PPR were built on the sequential graph. BM25 substantially underperforms embedding-based S_1. PPR is marginally above S_1 on Relational and below it on Standard.

Method	Analogue	Standard	Relational
BM25-papers	S1 (lexical)	.423	.101
BM25-KG	S2 (lexical+entities)	.447	.104
PPR (HippoRAG)	Graph-global	.460	.202
Phrase-flat	S_3's phrases, no graph	.486	.195
S_1	Flat RAG (embedding)	.531	.199
S_2	Hybrid (embedding)	.548	.231
S_3	Entity-aware	.571	.204
S_4	Hybrid-scored	.573	.206

Table 18 gives the three controls. The phrase-flat row takes S_3's own extracted phrases, averages their vectors into one query and retrieves papers by cosine with no graph: it reaches .486 on Standard against flat retrieval's .531 and .195 against .199 on Relational, so those phrases do not retrieve on their own and the traversal gain is not query expansion. S_1 is ahead of BM25 by $+.107$ on Standard and $+.094$ on Relational, so the graph advantage documented in the main text is measured against an already-strong embedding baseline rather than a weak one. Adding the extracted entity text to the BM25 index moves it in the same direction as the entity-embedding channel moves flat retrieval, $+.024$ on Standard against $S_2 - S_1$'s $+.018$, so entity-mediated retrieval is not specific to embedding similarity. PPR is marginally above S_1 on Relational (.202 against .199) and well below S_4 on Standard (.460 against .573), which we read as global graph diffusion diluting relevance in a dense typed-entity KG where targeted 2-hop expansion from matched entities does not.

I.5 Bootstrap Confidence Intervals

Retrieval is deterministic given a fixed graph, a fixed set of embedding vectors and a fixed interpreter hash seed. The one dependence on the last is in S_2, which walks a set of paper identifiers and stops once k candidates have accumulated; Python randomises string hashing per process, so the iteration order of that set, and therefore which papers make the cut, varies between runs. Setting PYTHONHASHSEED pins it, and the movement it produces is inside the floor of Appendix I.6. Two other sources of variation are handled separately: sensitivity to embedding precision, quantified in Appendix I.6, and sampling variance over the question set, which is what this subsection estimates.

We resample the question set with replacement (10,000 resamples, seed=42) and, because all strategies are scored on the same questions, we draw question indices once per replicate and apply the same indices to every strategy. Intervals are on paired differences from S_1, and Table 19 reports them for all three metrics.

The questions are not independent. All 647 come from 7 source papers, and questions drawn from one source share its bibliography, its phrasing and often its ground-truth papers, so resampling questions treats clustered observations as independent ones. Resampling the 7 source papers instead leaves every significant result of Table 1 significant. On Standard questions the traversal intervals narrow, to [+.020, +.082] for S_3 and [+.020, +.087] for S_4, which indicates an effect consistent across sources; on Relational questions the S_2 interval widens by a factor of 2.9, to [+.013, +.093], which indicates one that is not. With 7 clusters this check is itself underpowered, so we report it alongside the question-level intervals and not in place of them.

Table 19: Paired bootstrap 95% confidence intervals for strategy differences against S_1, on the 8K **sequential** graph (10,000 resamples). † marks an interval excluding zero. These are not the intervals of Table 1, which are the primary batch configuration; the two differ on Standard questions, and §4.2 tests that difference directly.

		Standard (274 Qs)		Relational (373 Qs)	
Metric	Diff	Mean	95% CI	Mean	95% CI
R@10	$S_2 - S_1$	−0.031	[−0.069, +0.007]	+0.002	[−0.007, +0.012]
	$S_3 - S_1$	+0.004	[−0.016, +0.022]	+0.002	[−0.001, +0.005]
	$S_4 - S_1$	+0.025	[−0.002, +0.052]	+0.007	[+0.001, +0.013]†
R@50	$S_2 - S_1$	+0.026	[−0.006, +0.060]	+0.016	[+0.002, +0.030]†
	$S_3 - S_1$	+0.025	[−0.003, +0.054]	+0.014	[+0.004, +0.024]†
	$S_4 - S_1$	+0.031	[−0.001, +0.062]	+0.013	[+0.003, +0.023]†
MRR	$S_2 - S_1$	−0.001	[−0.034, +0.033]	−0.005	[−0.040, +0.029]
	$S_3 - S_1$	+0.002	[−0.001, +0.004]	+0.002	[+0.000, +0.004]
	$S_4 - S_1$	+0.023	[+0.009, +0.038]†	−0.000	[−0.017, +0.015]
EPG Cohen's d		−0.050	[−0.617, +0.512]	+0.245	[−0.025, +0.523]

On Relational questions all three R@50 differences exclude zero, as does S_4's R@10 gain. EPG discrimination does not: its interval is [−0.025, +0.523]. On Standard questions no R@50 difference reaches significance, because the high tie rate reported in Appendix D leaves the paired differences at zero for most of the sample. On the sequential graph S_4's Standard R@50 interval is [−0.001, +0.062], which includes zero, so we make no claim there; on the primary batch graph the same contrast is +.043 [+.007, +.079] and does exclude zero. Two differences are negative: S_2 is below S_1 on Standard R@10 (−0.031) and on Relational MRR (−0.005), neither significant. The S_3 MRR intervals have a lower bound at +0.0001 on Relational, within resampling noise of zero, so we do not mark them.

I.6 Hardware Nondeterminism

Retrieval has no random component, but it is sensitive to the numerical precision of the embeddings. We re-ran one configuration (8K batch, canonicalization 0.85, Qwen3 embeddings) on CPU rather than GPU, with no other change, and compared per-question results. Table 20 reports the effect. Individual questions change often, because top-K selection turns a difference below our reporting precision into a different retrieved set. The aggregate movement is contrast-specific, and 0.015 bounds it for every contrast we measured. On Relational questions, where the reported effects are the significant ones, each contrast moves 0.003 to 0.004, against the 0.006 to 0.011 those same contrasts measure on this configuration. On Standard questions the movement is 0.002 for $S_4 - S_1$, 0.005 for $S_2 - S_1$ and 0.009 for $S_3 - S_1$. This was measured on one configuration, the 8K batch graph at canonicalization 0.85 with Qwen3 embeddings, which is not the configuration the primary results use. We have not run the same check on the sequential MiniLM graph, so these figures bound hardware sensitivity for the configuration named and are an extrapolation for any other.

The sign of every contrast is preserved. The ordering is preserved on Relational questions but not on Standard, where S_3 leads S_4 on GPU and trails it on CPU, so orderings between the two traversal strategies are inside hardware noise. The magnitudes move.

Table 20: Same configuration run on GPU and on CPU, reported as the movement each contrast shows between the two devices. *Questions changed* counts questions whose R@50 differs at all. Only the within-pair movement is comparable across tables, so we give that and not the two absolute gaps.

Question set	Contrast	GPU → CPU movement	Questions changed
Standard (274)	$S_2 - S_1$	+.005	9
	$S_3 - S_1$	−.009	18
	$S_4 - S_1$	+.002	18
Relational (373)	$S_2 - S_1$	−.003	51
	$S_3 - S_1$	−.003	122
	$S_4 - S_1$	−.004	120

J Domain Extension: Biomedical

We instantiate the KGRAF-Bench framework on a biomedical corpus to test cross-domain transferability. Starting from 3 highly cited recent biomedical surveys: (1) "A Review on Knowledge Graphs for Healthcare" (arXiv:2306.04802), (2) "A Survey of Large Language Models for Healthcare" (arXiv:2310.05694), (3) "A Survey of Large Language Models in Medicine" (arXiv:2311.05112), the toolkit generates a corpus and KG following the same layered construction, sized in Table 21:

Table 21: Biomedical corpus construction, following the layered scheme of Table 6.

Layer	Method	Papers	Role
L1	References cited by 3 biomedical surveys	781	Ground-truth relevant
L2	Co-citation neighbors of top-50 most-cited L1	3,504	Hard distractors
L3	Keyword-venue search	4,102	Corpus scale
	Total	8,387	

All papers pulled from Semantic Scholar. The corpus file holds 8,387 papers and the constructed graph holds 8,368: 19 papers failed extraction and are absent from the graph. The KG is constructed using batch EDC, so it uses the batch schema (5 entity types, 7 edge types) rather than the sequential schema of the main evaluation.

The toolkit generated 111 questions on this corpus, and every one resolves at least one ground-truth paper inside the corpus, so all 111 are scoreable and we evaluate on all of them. An earlier version of this appendix reported only the first 50 in generation order. We report the full set, because the restriction was not principled and it changes the result: the first-50 gaps are +.002 and +.014, below the 0.015 floor, where the full set's are above it. Table 22 gives the comparison.

Table 22: Biomedical domain extension (R@50) on all 111 generated questions, discussed below. The ML/AI column is the Standard question set on the primary configuration, the nearest available comparison since the biomedical questions are Standard-type. † marks an interval excluding zero.

Strategy	Biomed L1-only (781)	Biomed 8K	ML/AI 8K (paper)
S1 (flat RAG)	.488	.326	.531
S2 (hybrid)	.500	.334	.548
S3 (entity-aware)	.512	.349	.571
S4 (hybrid-scored)	.512	.347	.573
S2−S1 [95% CI]	+.011 [−.018, +.040]	+.008 [−.018, +.034]	+.017 [−.016, +.052]
S3−S1 [95% CI]	+.023 [−.006, +.052]	+.023 [−.003, +.049]	+.041 [+.006, +.075]†
S4−S1 [95% CI]	+.024 [−.006, +.052]	+.021 [−.005, +.048]	+.042 [+.006, +.079]†
EPG Cohen's d	−0.06	−0.29	+0.10

Two things differ from the ML/AI result. The traversal advantage is already present without distractors, $S_4 - S_1 = +.024$ on the 781-paper L1-only corpus, and it does not grow when L2 and L3 are added:

every strategy degrades (S_1.488 to .326, S_4.512 to .347) and the gap closes slightly to $+.021$. On this corpus the advantage is therefore a property of the domain instantiation rather than of the distractor pool, which is the opposite of what a corpus-composition account predicts and of what we measured in ML/AI. And the entity-paper gap does not discriminate in either condition, $d = -0.06$ and -0.29, both negative and neither separable from zero.

On 111 questions against 274 and 373 in the primary evaluation, the intervals are correspondingly wide and none excludes zero, so we draw no conclusion from the direction of any of this. The point estimates clear the 0.015 reproducibility floor. What the extension establishes is that the construction pipeline runs on another literature; it does not establish the retrieval finding there, and it does not support the corpus-composition account.

K Question Quality Audit

The first author manually rated 50 randomly sampled questions (25 Standard, of which 17 global and 8 local, plus 25 Relational; seed=42) against four quality criteria, answerability, unambiguity, non-triviality and representativeness, plus a descriptive difficulty rating. Ratings were made from the question text, its source section and survey, and the ground-truth paper titles, blind to retrieval results and strategy performance. Table 23 reports the per-criterion means. This is an expert audit by one annotator, so it carries no inter-annotator agreement; two further expert annotators will rate the same sample for the camera-ready version.

Table 23: Expert question audit, 50 sampled questions rated 1–5 by the first author against the rubric in §K. The pass rate counts the four quality criteria and excludes the difficulty rating, which is descriptive rather than a quality bar; requiring $\geq$3 on difficulty as well gives 88% overall, 76% Standard and 100% Relational. One annotator, so no agreement statistic; see §K.

Criterion	All (50)	Standard (25)	Relational (25)
Answerable	4.90	5.00	4.80
Unambiguous	3.88	4.08	3.68
Non-trivial	3.74	3.56	3.92
Representative	4.16	4.28	4.04
Difficulty	3.28	2.72	3.84
GT Relevant	100%	100%	100%
Pass rate ($\geq$3 on all four quality criteria)	98%	96%	100%

49/50 questions pass all criteria ($\geq$3). The single flagged question (LoRA) is trivially easy, not nonsensical. Relational questions are rated harder (3.84 vs 2.72) and slightly more ambiguous (3.68 vs 4.08), as expected by design. The ground-truth relevance row is independent of the generation pipeline, since the rater is a human and the questions were produced by an LLM: in all 50 cases the annotator judged the cited papers relevant to the question, which supports the citation-as-relevance assumption on this sample. What one annotator cannot establish is that a second would agree, which is the gap the camera-ready annotations are meant to close.