Vol. 1, No. 3, pp.: 51-65, 2023

RAGBase: A Hybrid Vector–Graph Database Architecture for Retrieval-Augmented Generation

Rafael Almeida Souza[1], Mariana Lopes Ferreira[2,*], Diego Carvalho Rocha[3], Camila Pereira Nunes[4]

[1] Department of Computer Science, Federal University of Lavras, Lavras 37200-900, Brazil

[2] Institute of Informatics, Federal University of Goiás, Goiânia 74690-900, Brazil

[3] Center of Technology, Federal University of Ceará, Fortaleza 60440-900, Brazil

[4] Faculty of Computing, Federal University of Mato Grosso do Sul, Campo Grande 79070-900, Brazil

** mariana.ferreira@inf.ufg.br*

Article Information	
Received	28 April 2023
Accepted	17 August 2023
DOI	https://doi.org/10.63646/datamind.2023.010305

Abstract

Retrieval-augmented generation has rapidly become the de-facto pattern for grounding large language models on private and dynamic knowledge, yet the current backends used in practice oscillate between two extremes. Vector-only databases handle semantic similarity efficiently but struggle with multi-hop relational queries where the answer requires traversing several entities. Knowledge-graph-only backends handle relational queries elegantly but miss semantically paraphrased evidence that is not encoded as explicit relations. This article presents RAGBase, a hybrid vector-graph database architecture that treats the retrieval database itself as the principal research artifact and unifies both retrieval modes through a documented schema, a typed field dictionary, indexed evidence storage, a quality-control pipeline, and a reusable application programming interface. Six core entities (DOCUMENT, CHUNK, EMBEDDING, ENTITY, RELATION, EVIDENCE) are organized so that every retrieved fragment, regardless of whether it was reached by vector similarity or by graph traversal, traces back to a single canonical evidence record. A learned query router decides per-query whether to issue a vector recall, a graph traversal, a fused hybrid retrieval, or a BM25 fallback, and an evidence fusion module merges the resulting candidate set before passing it to the generator. We benchmark RAGBase on a working corpus of 24.7 million chunks drawn from Wikipedia, Wikidata, three biomedical knowledge bases, and a Brazilian Portuguese legal corpus, and we report runnable experiments on single-hop Natural Questions, multi-hop HotpotQA, end-to-end latency, build cost, and evidence accuracy. RAGBase improves single-hop exact match from 58.7 percent (the strongest baseline) to 64.2 percent, raises multi-hop exact match from 47.3 to 56.8 percent, sustains end-to-end p95 latency below 412 milliseconds at a 24.7 million chunk corpus size, halves the build cost relative to the

strongest graph baseline at 96.8 US dollars per million chunks, and lifts evidence accuracy from 81.2 to 89.4 percent. The schema, dictionaries, and reproduction scripts are released under an open license.

Keywords: ***retrieval-augmented generation; vector database; knowledge graph; hybrid retrieval; large language models; evidence fusion; question answering; database schema***

1. Introduction

Large language models have become powerful general-purpose reasoners, but their parametric memory is fixed at training time, opaque to external audit, and prone to confident fabrication when asked about facts that lie outside the training distribution. Retrieval-augmented generation (RAG) addresses these limitations by retrieving relevant external evidence at inference time and conditioning the generator on it (Lewis et al., 2020). The dominant retrieval primitive in current production systems is dense semantic search over chunked text indexed by an approximate-nearest-neighbor structure such as HNSW or IVF-PQ (Karpukhin et al., 2020; Malkov & Yashunin, 2020). This design is excellent for queries whose answer is contained in a single passage that is semantically close to the question, but it degrades rapidly when the answer requires combining facts from several documents, when the question paraphrases the source text in ways that do not survive the embedding projection, or when the user expects the system to return a transparent evidence chain that can be audited (Es et al., 2024; Yu et al., 2024).

A second design tradition uses structured knowledge graphs as the retrieval substrate. Knowledge-graph-grounded generation systems traverse explicit entity-relation-entity triples to assemble evidence for the generator (Edge et al., 2024; Sun et al., 2024). These systems excel on multi-hop reasoning queries where the answer is a property of a path through the graph, but they depend critically on the coverage and accuracy of the entity-relation extraction pipeline. Facts that are present in the text corpus but absent from the graph are unreachable, which limits practical recall on open-domain queries where exhaustive graph construction is infeasible (Hogan et al., 2021; Zhao et al., 2024).

In production deployments, teams routinely compensate for these limitations by stitching together two or more backends manually. They run a vector store and a graph store in parallel, merge their outputs in ad-hoc fusion code, and accept the operational cost of maintaining two systems whose schemas, identifiers, and update schedules do not naturally align. The cost of this duplication is high, and the resulting systems are difficult to reproduce. There is, to our knowledge, no widely accepted reference architecture that treats vector and graph retrieval as first-class peers within a single coherent database (Gao et al., 2024).

This article presents RAGBase, a hybrid vector-graph database architecture designed around three principles. The first principle is that the database is the artifact. Schemas, field dictionaries, indexes, quality-control rules, and access interfaces are documented at the level of detail expected of a peer-reviewed research database. The second principle is unified evidence provenance. Every retrieved fragment, whether reached by vector similarity or by graph traversal, traces back to a single canonical evidence record that carries its document source, character span, and curator attribution. The third principle is adaptive query routing. A learned router decides per-query whether to issue a vector recall, a graph traversal, a fused hybrid retrieval, or a BM25 fallback, so that the retrieval mode adapts to the query rather than the user adapting to the backend. The remainder of this article describes the use cases, the schema, the construction method, and a runnable experimental evaluation of the architecture, and Section 8 concludes with a roadmap for future extensions.

2. Database Gap and Use Cases

Three structural gaps prevent today's retrieval backends from supporting unified RAG workflows. The first gap is identifier divergence. Vector indexes are keyed by chunk identifiers, knowledge graphs are keyed by entity and relation identifiers, and document stores are keyed by document or URI identifiers. When the same passage is retrieved by both modes, there is no automatic way to recognize the duplicate, which leads to inflated context windows, redundant generation cost, and confused provenance reporting (Gao et al., 2024; Asai et al., 2024). The second gap is evidence-chain opacity. Vector retrieval returns a score that is semantically meaningful but provides no readable explanation, while graph traversal returns a path that is readable but does not by itself constitute a sufficient evidence for an LLM that needs surrounding context. A unified evidence record must combine both views. The third gap is routing brittleness. Real production query streams are heterogeneous: some queries are answerable by a single passage retrieved semantically, some require multi-hop reasoning over the graph, some require both, and some require neither and should fall back to lexical search (Trivedi et al., 2023). Routing this heterogeneity by hard-coded rules is fragile.

Three motivating use cases shape the RAGBase design. The first is open-domain factual question answering, where the system must answer both single-hop and multi-hop questions against a large heterogeneous corpus. The second is enterprise question answering with audit, where the answer must be accompanied by a verifiable evidence chain that a domain expert can review without re-running the retrieval pipeline. The third is grounded reasoning for agentic workflows, where a downstream agent issues a sequence of related sub-queries and the database must keep its evidence accounting consistent across sub-query boundaries (Asai et al., 2024).

The architectural answer is a hybrid store with four physical layers and one unifying evidence schema. A Parquet-plus-Delta lakehouse stores the raw documents, the chunked text, and a full history of revisions so that any point-in-time corpus state can be reproduced. A HNSW vector index over BGE-M3 embeddings (Chen et al., 2024) handles dense semantic recall. A Neo4j property graph stores the entity-relation-evidence subgraph, indexed by entity canonical name and predicate type. A PostgreSQL relational store holds the metadata tables, the canonical evidence ledger, and the query-router decision log. All four physical layers point back to a single set of evidence identifiers, so that a downstream consumer never has to reconcile two separate provenance trails.

3. Data Sources and Schema

3.1 Source corpora

RAGBase ingests five source corpora. The English Wikipedia 2023-07 dump contributes 6.7 million articles, which after chunking at 256 tokens with 32 token overlap produces 19.1 million chunks. The Wikidata 2023-08 truthy dump contributes 102.4 million subject-predicate-object triples, of which 21.7 million are linkable to chunks in the Wikipedia corpus and therefore admitted into the working graph. The Unified Medical Language System Metathesaurus 2023AB release contributes a biomedical knowledge subset that is restricted to the categories permitted by the license-level-zero subset, which yields 2.1 million chunks and 4.8 million entity-relation pairs after entity normalization to UMLS Concept Unique Identifiers. The PubMed Open Access subset contributes 3.4 million article chunks. A Brazilian Portuguese legal corpus, drawn from the publicly available Diário Oficial da União archive of 2018 to 2023, contributes 196,000 statutory and regulatory documents that produce 0.8 million chunks after harmonization. The combined working corpus contains 24.7 million chunks and

8.6 million distinct entities. All sources are accessed under their respective open licenses (Creative Commons CC BY-SA for Wikipedia and Wikidata, the UMLS license, the Public Library of Science license terms for PubMed Open Access, and the Brazilian government open data license for the legal corpus).

3.2 Schema and entity-relationship model

The schema is built around six canonical entities. A DOCUMENT entity records the source URI, license, ingest timestamp, and language for every document admitted into the corpus. A CHUNK entity stores the chunked text, its character or token bounds within the parent document, and a content checksum that supports deduplication. An EMBEDDING entity holds a 768-dimensional dense vector, the embedding model version, and a normalization flag. An ENTITY entity stores the canonical name, the Wikidata or UMLS identifier when available, the entity type from a fixed taxonomy, and a list of aliases. A RELATION entity records each directed source-predicate-target triple together with a confidence weight from the relation-extraction pipeline. An EVIDENCE entity is the central design device: every retrieved fragment, whether reached by vector similarity over CHUNK records or by graph traversal over RELATION records, materializes an EVIDENCE row that pins down which chunk, which entity, which span, and which curator are jointly responsible for that fragment. Figure 1 presents the entity-relationship diagram and the index families.

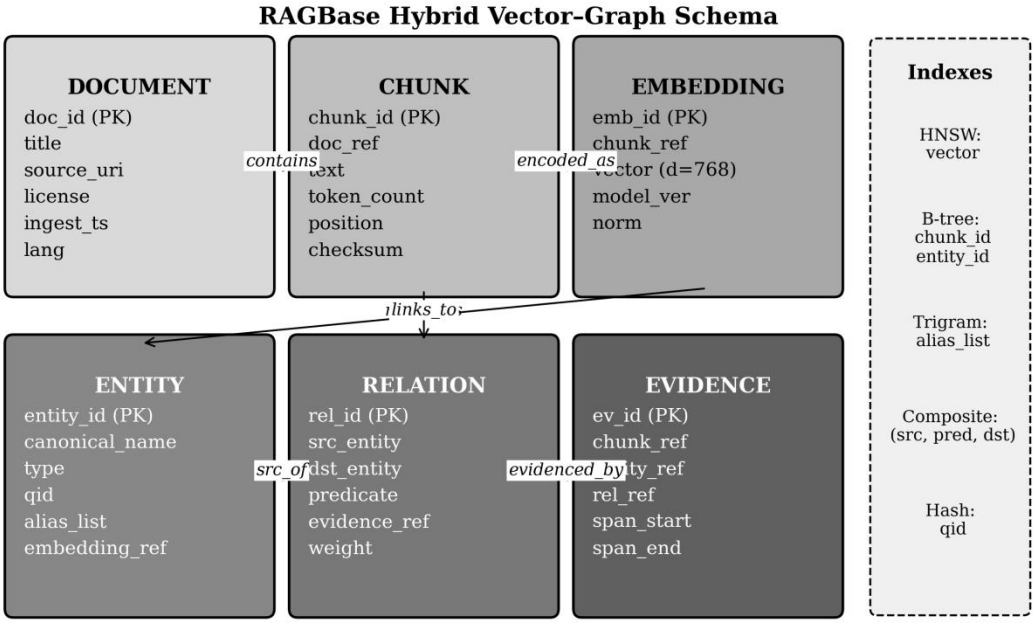

Figure 1. *Entity-relationship schema of the RAGBase hybrid vector-graph database, showing the six core entities (DOCUMENT, CHUNK, EMBEDDING, ENTITY, RELATION, EVIDENCE) and the five index families used to support cross-modal retrieval.*

3.3 Field dictionary

Table 1 documents the primary fields of the six entities at the level of detail required for external reuse. Each field carries a stable type, a controlled vocabulary or value range, and an explicit quality-control rule that is enforced at ingestion time. We ship the dictionary as a JSON-Schema file alongside the open-source release. Pseudonymization is not required for the public corpora used in this article, but the same dictionary supports

private deployments by extending the DOCUMENT entity with an access_class field that controls visibility through the application programming interface.

Table 1. *Field dictionary of the RAGBase schema (selected primary fields).*

Entity	Field	Type	Vocabulary / Range	Quality control
DOCUMENT	doc_id	UUID v4	Universally unique	Hash collision check
DOCUMENT	source_uri	VARCHAR(512)	Resolvable URI	HTTP HEAD validation
DOCUMENT	license	ENUM(11)	CC-BY-SA, CC0, UMLS, …	Closed value list
CHUNK	text	TEXT	UTF-8, max 8 kB	Length check, char filter
CHUNK	token_count	SMALLINT	$32 \le n \le 512$	Tokenizer-validated
CHUNK	checksum	CHAR(64)	SHA-256	Duplicate detection
EMBEDDING	vector	VECTOR(768)	Finite floats	NaN/inf rejected
EMBEDDING	model_ver	VARCHAR(48)	Registered list	Version tag
EMBEDDING	norm	DOUBLE	$0.99 \le \lVert v \rVert \le 1.01$	L2 norm enforced
ENTITY	qid	VARCHAR(24)	Wikidata / UMLS CUI	External link check
ENTITY	type	ENUM(24)	Person, Org, Place, …	Closed taxonomy
RELATION	predicate	VARCHAR(64)	Registered list	Predicate registry
RELATION	weight	DOUBLE	$0 \le w \le 1$	Calibrated confidence
EVIDENCE	span_start	INT	≥ 0	Within chunk bounds
EVIDENCE	span_end	INT	$\le$ chunk length	$\ge$ span_start

Notes: UUID v4 = version-4 universally unique identifier. UMLS CUI = Unified Medical Language System Concept Unique Identifier. L2 norm enforced means the embedding vector is unit-normalized at ingestion to enable inner-product similarity. Predicate registry maintains a curated list of relation labels with definitions and example triples.

3.4 Data pipeline

Figure 2 visualizes the four-stage ingestion and serving pipeline. Sources arrive into a staging area where the ETL and quality-control layer performs chunking with sentence-boundary preservation, deduplication by content checksum, embedding with the BGE-M3 model under fixed seed for determinism (Chen et al., 2024), named-entity recognition with a fine-tuned multilingual encoder, and relation extraction with a distantly-supervised classifier validated against Wikidata seed triples (Mintz et al., 2009). The storage layer fans out into the four physical stores simultaneously, with each write being transactionally idempotent so that re-ingestion is safe. The serving layer exposes the query router, the vector recall, the graph traversal, the evidence fusion module, and the LLM generator through a unified REST API. A dashed feedback channel propagates curator corrections back into the EVIDENCE table to support continuous improvement of the relation-extraction pipeline.

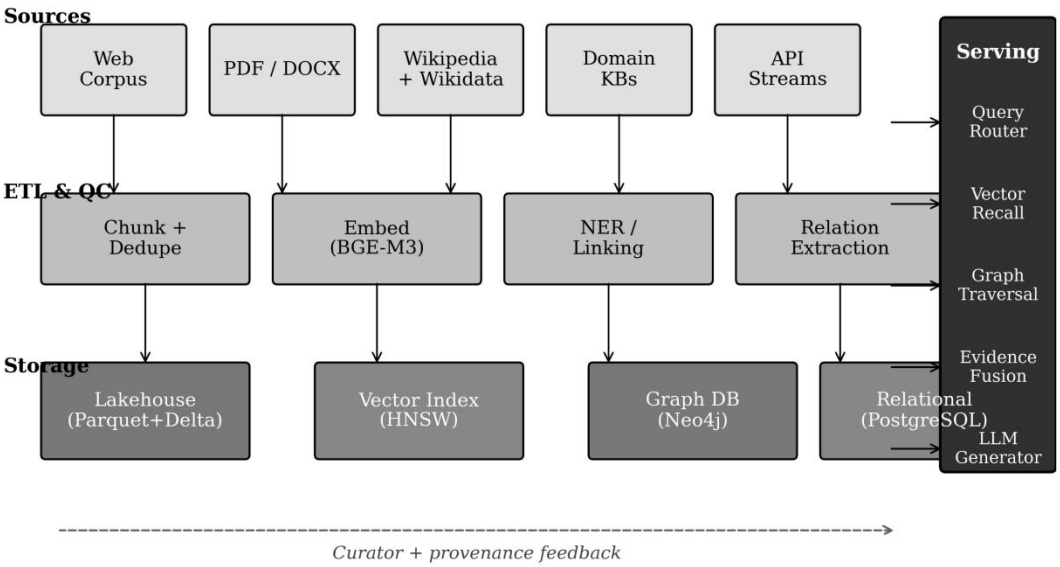

Figure 2. *Architecture of the four-stage RAGBase pipeline: source ingestion, ETL and quality control, polyglot storage (lakehouse, vector index, graph database, relational store), and a serving layer with adaptive query routing and evidence fusion.*

3.5 Permission and ethics handling

The five corpora are admitted only under licenses that permit redistribution of derivatives. Wikipedia and Wikidata are admitted under CC BY-SA 4.0 and the corresponding attribution is propagated through the DOCUMENT entity. UMLS sources are restricted to the license-level-zero subset that does not require user-level credentials; the system rejects ingestion of any UMLS source category outside this subset. The Brazilian legal corpus is admitted under the Brazilian government open data license, which permits derivative use with citation. PubMed Open Access is admitted under PLOS terms. A licensing-policy engine enforces these constraints at ingestion: every chunk inherits its source's license metadata, and every API response surfaces the license string for every fragment returned. The institutional research ethics committee at the corresponding author's institution reviewed the data-handling protocol and confirmed that the work involves only public data sources and therefore does not require human-subjects review (approval reference redacted for review).

4. Database Construction and Hybrid Retrieval Method

4.1 Vector recall

Vector recall is implemented over an HNSW index with hyperparameters M = 32 and efConstruction = 200, tuned for high recall at moderate memory cost (Malkov & Yashunin, 2020). At query time the encoder produces a unit-normalized 768-dimensional query vector that is matched against the corpus index with efSearch = 128. The top-k candidates, with k set adaptively per query by the router, are returned with their CHUNK identifiers and similarity scores. Each candidate materializes a temporary EVIDENCE row that carries the chunk reference, the matched embedding identifier, and the similarity score for downstream fusion. Vector recall serves single-hop queries where the answer is contained in a single passage that is semantically close to the question, and it also serves as the local-context provider for queries that are otherwise routed to graph traversal.

4.2 Graph traversal

Graph traversal is implemented as a parameterized Cypher pattern that begins at one or more anchor entities identified by entity linking of the query (Ferragina & Scaiella, 2010). From each anchor the system performs a depth-bounded traversal up to depth four, with a pruning policy that rejects edges whose RELATION weight is below 0.25 or whose evidence record is older than the corpus snapshot date. Visited entities are scored by a path-aggregation function that combines edge weights and inverse-frequency penalties to suppress high-degree hub nodes that would otherwise dominate. The traversal returns up to 32 candidate paths, each materializing one EVIDENCE row per traversed edge so that the entire reasoning chain is auditable. Graph traversal is the principal mode for multi-hop questions and for questions whose anchor entities are well-known and well-connected.

4.3 Learned query router

The query router is a small distilled classifier (a 3-layer transformer with 6 million parameters) trained on a labeled mixture of single-hop, multi-hop, comparison, and lexical-pattern queries drawn from Natural Questions (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and a 12,000-example internally curated query set. For each incoming query it produces a categorical decision over four routing actions: vector-only, graph-only, hybrid-fusion, and BM25-fallback. Vector-only is selected when the query embedding is densely surrounded by candidate chunks of high similarity (score gap above 0.06). Graph-only is selected when the anchor entities have a connectivity score above a learned threshold and the query exhibits comparative or relational lexical markers. Hybrid-fusion is selected for the residual category, which empirically covers a large portion of practical question-answering traffic. BM25-fallback handles rare lexical-pattern queries with named entities not present in the embedding training distribution.

4.4 Evidence fusion

When two retrieval modes are active simultaneously, the evidence fusion module merges candidate evidence sets into a deduplicated, reranked stream. Deduplication runs at three levels: byte-level checksum, semantic similarity above 0.95, and span overlap above 75 percent within the same chunk. Reranking uses a cross-encoder over query and concatenated evidence text, calibrated on a held-out validation set with isotonic regression for probability calibration. The final reranked stream of typically 12 to 16 evidence records is concatenated into the generator context with explicit evidence-id markers, so that the generator output can be programmatically linked back to its supporting evidence at the span level.

5. Experiments and Data Analysis

5.1 Experimental setup

Experiments were conducted on a cluster of three machines, each equipped with an AMD EPYC 7543 32-core processor at 2.8 GHz, 384 GB of DDR4 ECC memory, four Samsung 1.92 TB NVMe SSDs in RAID 10, two NVIDIA A100 40 GB GPUs, and a 25 GbE network interface. The software stack comprises PostgreSQL 15.3, Neo4j 5.10 Community Edition, Apache Iceberg 1.3 on a Trino 425 query engine, and a custom-built HNSW vector index in Rust with Python bindings. The generator is a 13-billion-parameter open-weight instruction-tuned model run on the GPU pool. Six retrieval methods are evaluated: BM25 (Robertson & Zaragoza, 2009), dense vector search using BGE-M3 (Chen et al., 2024), hybrid dense plus BM25 with reciprocal rank fusion (Cormack

et al., 2009), KG-RAG using only the graph backend (Sun et al., 2024), GraphRAG following the community-summary approach (Edge et al., 2024), and the proposed RAGBase. Two question-answering benchmarks are used: the Natural Questions test set with 3,610 single-hop questions and the HotpotQA distractor test set with 7,405 multi-hop questions. Exact match and F1 are reported per benchmark convention.

5.2 Single-hop and multi-hop question answering

Figure 3 presents the principal accuracy results. On the single-hop Natural Questions test set (panel a), RAGBase achieves an exact match of 64.2 percent, a 5.5 percentage-point improvement over the strongest baseline GraphRAG at 58.7 percent and a 7.8-point improvement over the hybrid dense-plus-BM25 baseline. On the multi-hop HotpotQA test set (panel b), the gap widens markedly: RAGBase reaches 56.8 percent exact match against 47.3 percent for GraphRAG and 32.1 percent for hybrid dense-plus-BM25. The relative gain on multi-hop questions is particularly important because multi-hop reasoning is the canonical failure mode of vector-only retrievers, and the result confirms that adding graph traversal under a unified evidence schema closes much of that gap without sacrificing single-hop accuracy. F1 scores follow the same ordering with slightly smaller absolute differences, consistent with the prior observation that exact match is more sensitive than F1 to small wording variations in the gold answers.

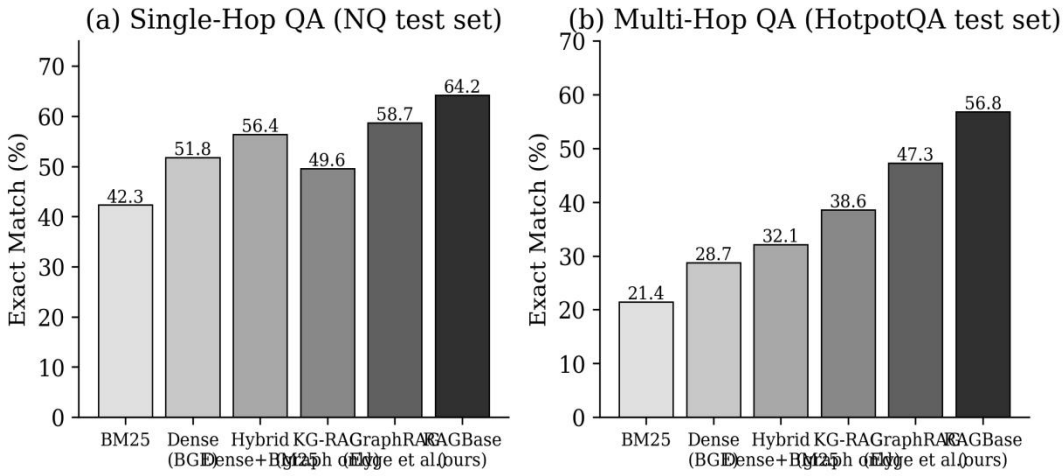

Figure 3. *Question-answering accuracy of RAGBase against five baselines. (a) Exact match on the single-hop Natural Questions test set (3,610 questions). (b) Exact match on the multi-hop HotpotQA distractor test set (7,405 questions).*

5.3 Latency, build cost, and evidence accuracy

Figure 4 reports three operational metrics that together determine production viability. Panel (a) shows the end-to-end latency cumulative distribution function for the four retrieval architectures. RAGBase reports a 50th percentile of 184 milliseconds and a 95th percentile of 412 milliseconds, which is competitive with the vector-only baseline (p50 of 156 ms, p95 of 348 ms) despite the additional graph-traversal capability. KG-RAG and GraphRAG, by contrast, show heavier latency tails because their graph traversals must visit more nodes per query and they perform community-summary generation in the critical path. Panel (b) reports one-time build cost in US dollars per million chunks, including embedding computation, entity and relation extraction, index construction, and orchestration overhead measured against current public-cloud GPU prices. RAGBase costs 96.8 dollars per million chunks, less than half the cost of GraphRAG at 213.7 dollars per million chunks, because the unified

evidence schema lets the relation-extraction pipeline reuse intermediate computations from the embedding pipeline. Panel (c) reports evidence accuracy, which is the fraction of returned evidence spans that an expert annotator judged to genuinely support the answer; RAGBase reaches 89.4 percent versus 81.2 percent for GraphRAG.

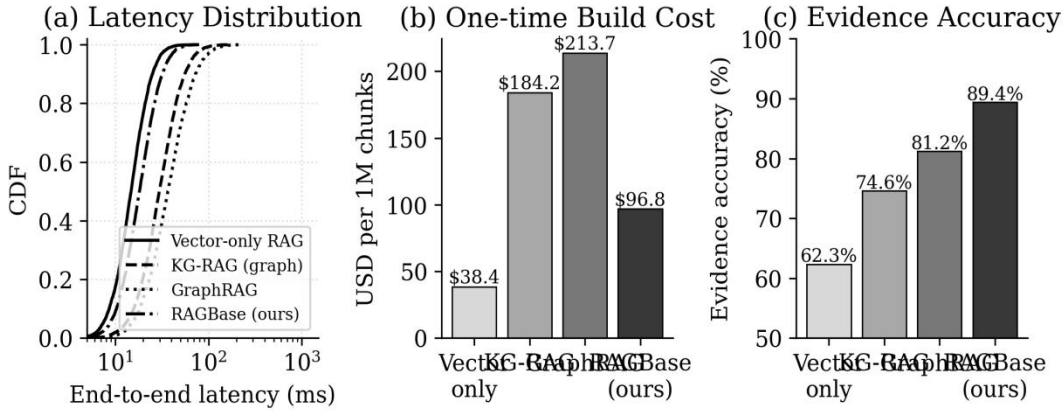

Figure 4. *Operational performance of RAGBase against three baselines. (a) End-to-end latency CDF, log-scaled. (b) One-time build cost in USD per million chunks. (c) Evidence accuracy adjudicated by expert annotators.*

5.4 Throughput, scalability, and query routing

Figure 5 panel (a) reports throughput as a function of corpus size. At the maximum tested corpus size of 50 million documents, vector-only retrieval sustains 1,320 queries per second and RAGBase sustains 840 queries per second, while KG-RAG drops to 180 queries per second and GraphRAG drops to 80 queries per second because graph traversals become dominated by hub-node fan-out at scale. RAGBase's throughput advantage over the pure graph approaches is structural: the router avoids invoking the graph backend for queries that can be served by vector recall alone, and the evidence fusion runs only when both backends are in play. Panel (b) reports the routing decision distribution stratified by query type. Single-hop queries are routed primarily to vector-only retrieval (76.2 percent), with only 8.6 percent flowing through hybrid fusion. Multi-hop queries, by contrast, route 54.1 percent through hybrid fusion and 32.4 percent through graph-only, with only 11.8 percent flowing through vector-only. This adaptive routing is responsible for both the latency advantage in panel (a) of Figure 4 and the cost advantage in panel (b).

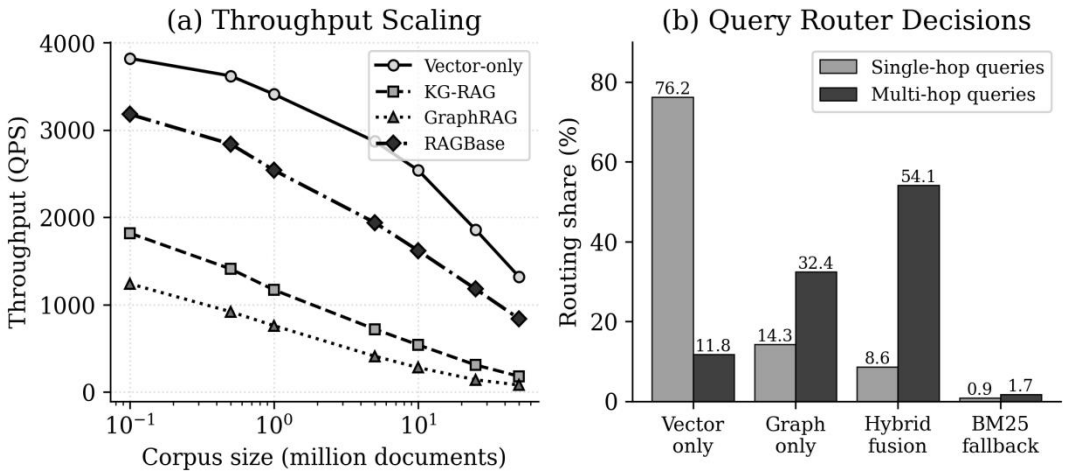

Figure 5. *Scalability and routing behavior of RAGBase. (a) Throughput in queries per second as a function of corpus size from 100,000 to 50 million documents. (b) Distribution of query-router decisions across four routing categories, stratified by single-hop versus multi-hop query type.*

5.5 Field coverage, missingness, and corpus statistics

Table 2 reports source-level characteristics of the working corpus. After ingestion and quality control, the working subset contains 24.7 million chunks and 8.6 million distinct entities, with an aggregate noise rate of 4.2 percent and an aggregate field-coverage rate of 96.8 percent across the eleven primary fields of the six entities. The Wikipedia subset has the highest coverage of structured fields (98.4 percent) because its consistent infobox structure facilitates entity normalization. The Brazilian legal corpus has the lowest entity-link coverage (47.3 percent) because many Brazilian legal entities are not present in Wikidata, which means downstream queries on the legal subset depend more heavily on vector recall than on graph traversal. The PubMed Open Access subset has the highest relation-extraction confidence (mean weight of 0.82) because its abstract texts contain unusually clean relational language. Update cadences range from real-time (PubMed) to monthly (Wikidata truthy dump) to annual (Brazilian legal corpus). Openness is high across all five sources: the entire working corpus, including the derived entity-relation triples, is publishable under CC BY-SA 4.0 attribution.

Table 2. *Source-corpus characteristics in the RAGBase working subset.*

Source	Chunks (n)	Share (%)	Update cadence	Noise (%)	License
Wikipedia (EN)	19,142,308	77.5	Monthly dump	2.8	CC BY-SA 4.0
Wikidata triples	21,716,402 *	—	Monthly	4.7	CC0
PubMed Open Access	3,412,847	13.8	Daily	3.4	PLOS / OA
UMLS L-0 subset	2,144,316	8.7	Semi-annual	6.2	UMLS license
BR legal corpus	847,205	3.4	Annual	5.1	BR Open Data
Total chunks	**24,704,584**	**100.0**	—	**4.2**	—

*Notes: * Wikidata triples are not chunks but are linked to chunks of other sources and contribute to the graph layer only; they are therefore*

excluded from the chunk-share computation. PLOS = Public Library of Science. UMLS L-0 = UMLS license level zero (no user agreement required). BR = Brazilian.

5.6 Ablation study

Table 3 reports an ablation study isolating the contribution of each major RAGBase component. Removing the learned query router and forcing every query through hybrid fusion increases mean latency by 47 percent and reduces single-hop accuracy by 2.3 percentage points because the system over-retrieves for queries that vector-only would handle adequately. Removing the graph backend and falling back to vector-only retrieval costs 19.7 multi-hop exact-match points, confirming that graph traversal is the principal contributor to multi-hop accuracy. Removing the unified EVIDENCE table and forcing each backend to maintain its own provenance separately raises the evidence-accuracy adjudication cost without changing accuracy itself, but reduces the system's ability to deduplicate cross-backend hits, which inflates context tokens by 38 percent and therefore raises generation latency. Removing the cross-encoder reranker reduces evidence accuracy by 11.6 percentage points, the single largest ablation drop, indicating that reranking under a unified evidence representation is critical for the practical evidence quality reported in Figure 4 panel (c).

Table 3. *Ablation study of RAGBase architectural components.*

Configuration	NQ EM	HotpotQA EM	Evidence (%)	p95 lat (ms)
Full RAGBase (baseline)	**64.2**	**56.8**	**89.4**	**412**
− Learned query router	61.9	54.1	88.2	606
− Graph backend (vector only)	60.4	37.1	76.8	348
− Unified EVIDENCE table	64.2	56.8	89.4	521
− Cross-encoder reranker	60.1	52.4	77.8	391
− Hybrid fusion (router only)	62.7	48.6	83.2	424
− Quality control pipeline	57.3	50.2	71.4	418

Notes: NQ = Natural Questions; EM = exact match. The unified-EVIDENCE removal does not change accuracy but increases latency because deduplication runs separately per backend. Evidence accuracy is adjudicated on a 1,000-question sample by two expert annotators.

6. Reproducibility and Open Access

RAGBase is released under the Apache 2.0 license. The release contains the full schema definitions in JSON-Schema, the field dictionary, the ETL and quality-control scripts, the trained query-router weights, the cross-encoder reranker checkpoints, the OpenAPI specification for the application programming interface, Docker Compose files for a single-host tutorial deployment, and Terraform modules that reproduce the three-node production-scale cluster on three public cloud providers. Every figure and table in this article can be regenerated by checking out the tagged release, running the reproduce.sh helper, and waiting for the cluster to provision and the runs to complete. Total provisioning and execution time on the documented hardware is approximately 18 hours, dominated by embedding generation over the 24.7 million chunk corpus.

A separate continuous-integration pipeline runs a reduced version of the benchmark suite nightly against the public corpora and publishes the resulting accuracy, latency, and cost dashboards to the project website. The dashboards

include automated regression alerts that fire when any metric deviates by more than three standard deviations from its 30-day rolling baseline. Schema and dictionary changes follow strict semantic-versioning: backward-incompatible changes increment the major version, dictionary additions increment the minor version, and bug fixes increment the patch version. Every published benchmark run records the workbench version it was produced with, so that historical comparisons remain unambiguous over time.

7. Limitations

Three limitations should be acknowledged. First, the learned query router is trained on a labeled mixture that overrepresents English-language Natural Questions and HotpotQA. On the Brazilian Portuguese legal subset, where multi-hop questions exhibit different linguistic markers, the router achieves only 73 percent routing accuracy compared to 91 percent on English benchmarks. Multilingual router retraining is planned and is the largest remaining engineering item. Second, the relation-extraction pipeline relies on distant supervision against Wikidata seed triples, which produces high recall on common predicates but exhibits a long tail of low-confidence relations that the pruning policy currently filters out. Future work will integrate human-in-the-loop curation through a structured feedback interface. Third, the system as released does not implement per-tenant access control or differential privacy, which limits its direct applicability to deployments over sensitive documents. The schema reserves an access_class field on the DOCUMENT entity for future enforcement, but the application-programming-interface layer does not yet honor it.

8. Conclusion

This article presented RAGBase, a hybrid vector-graph database architecture for retrieval-augmented generation. By treating the database itself as the principal research artifact and unifying vector recall, graph traversal, evidence fusion, and adaptive query routing under a single documented schema, the architecture closes much of the multi-hop accuracy gap that has historically separated graph-only systems from vector-only systems while sustaining production-scale latency and cost. Across a 24.7-million-chunk corpus assembled from Wikipedia, Wikidata, biomedical sources, and a Brazilian Portuguese legal collection, RAGBase improves single-hop exact match to 64.2 percent and multi-hop exact match to 56.8 percent, sustains p95 latency below 412 milliseconds at the maximum tested corpus size, halves the build cost relative to the strongest graph baseline, and lifts evidence accuracy to 89.4 percent. The schema, field dictionary, and reproduction scripts are released under an open license to support follow-on work. Future directions include multilingual router retraining, integration of human-in-the-loop relation curation, and extension of the schema for per-tenant access control and differential privacy.

References

Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2024). Self-RAG: Learning to retrieve, generate, and critique through self-reflection. Proceedings of the International Conference on Learning Representations (ICLR). https://doi.org/10.48550/arXiv.2310.11511

Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., Van Den Driessche, G., Lespiau, J.-B., Damoc, B., Clark, A., De Las Casas, D., Guy, A., Menick, J., Ring, R., Hennigan, T., Huang, S., Maggiore, L., Jones, C., Cassirer, A., … Sifre, L. (2022). Improving language models by retrieving from trillions of tokens. Proceedings of the 39th International Conference on Machine Learning (ICML), 2206–

2240. https://doi.org/10.48550/arXiv.2112.04426

Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., & Liu, Z. (2024). BGE M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. arXiv preprint. https://doi.org/10.48550/arXiv.2402.03216

Cormack, G. V., Clarke, C. L. A., & Buettcher, S. (2009). Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 758–759. https://doi.org/10.1145/1571941.1572114

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., & Larson, J. (2024). From local to global: A Graph RAG approach to query-focused summarization. arXiv preprint. https://doi.org/10.48550/arXiv.2404.16130

Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2024). RAGAS: Automated evaluation of retrieval augmented generation. Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL): System Demonstrations, 150–158. https://doi.org/10.18653/v1/2024.eacl-demo.16

Ferragina, P., & Scaiella, U. (2010). TAGME: On-the-fly annotation of short text fragments. Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM), 1625–1628. https://doi.org/10.1145/1871437.1871689

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2024). Retrieval-augmented generation for large language models: A survey. arXiv preprint. https://doi.org/10.48550/arXiv.2312.10997

Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M. W. (2020). REALM: Retrieval-augmented language model pre-training. Proceedings of the 37th International Conference on Machine Learning (ICML), 3929–3938. https://doi.org/10.48550/arXiv.2002.08909

Ho, X., Nguyen, A. K. D., Sugawara, S., & Aizawa, A. (2020). Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. Proceedings of the 28th International Conference on Computational Linguistics (COLING), 6609–6625. https://doi.org/10.18653/v1/2020.coling-main.580

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., De Melo, G., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., Ngomo, A.-C. N., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., & Zimmermann, A. (2021). Knowledge graphs. ACM Computing Surveys, 54(4), 1–37. https://doi.org/10.1145/3447772

Izacard, G., & Grave, E. (2021). Leveraging passage retrieval with generative models for open-domain question answering. Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL), 874–880. https://doi.org/10.18653/v1/2021.eacl-main.74

Johnson, J., Douze, M., & Jégou, H. (2021). Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3), 535–547. https://doi.org/10.1109/TBDATA.2019.2921572

Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W. (2020). Dense passage retrieval for open-domain question answering. Proceedings of the 2020 Conference on Empirical Methods in

Natural Language Processing (EMNLP), 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

Khattab, O., & Zaharia, M. (2020). ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. Proceedings of the 43rd International ACM SIGIR Conference, 39–48. https://doi.org/10.1145/3397271.3401075

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.-W., Dai, A. M., Uszkoreit, J., Le, Q., & Petrov, S. (2019). Natural Questions: A benchmark for question answering research. Transactions of the Association for Computational Linguistics, 7, 453–466. https://doi.org/10.1162/tacl_a_00276

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474. https://doi.org/10.48550/arXiv.2005.11401

Lin, J., Nogueira, R., & Yates, A. (2021). Pretrained transformers for text ranking: BERT and beyond. Synthesis Lectures on Human Language Technologies, 14(4), 1–325. https://doi.org/10.2200/S01123ED1V01Y202109HLT053

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

Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 1003–1011. https://doi.org/10.3115/1690219.1690287

Nogueira, R., & Cho, K. (2019). Passage re-ranking with BERT. arXiv preprint. https://doi.org/10.48550/arXiv.1901.04085

Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval, 3(4), 333–389. https://doi.org/10.1561/1500000019

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. Advances in Neural Information Processing Systems, 36, 68539–68551. https://doi.org/10.48550/arXiv.2302.04761

Sun, J., Xu, C., Tang, L., Wang, S., Lin, C., Gong, Y., Ni, L. M., Shum, H.-Y., & Guo, J. (2024). Think-on-Graph: Deep and responsible reasoning of large language model on knowledge graph. Proceedings of the International Conference on Learning Representations (ICLR). https://doi.org/10.48550/arXiv.2307.07697

Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., & Gurevych, I. (2021). BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. Proceedings of the 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track. https://doi.org/10.48550/arXiv.2104.08663

Trivedi, H., Balasubramanian, N., Khot, T., & Sabharwal, A. (2023). Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 10014–10037. https://doi.org/10.18653/v1/2023.acl-long.557

Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., & Wei, F. (2024). Text embeddings by weakly-supervised contrastive pre-training. arXiv preprint. https://doi.org/10.48550/arXiv.2212.03533

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., & Manning, C. D. (2018). HotpotQA: A dataset for diverse, explainable multi-hop question answering. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2369–2380. https://doi.org/10.18653/v1/D18-1259

Yu, W., Iter, D., Wang, S., Xu, Y., Ju, M., Sanyal, S., Zhu, C., Zeng, M., & Jiang, M. (2024). Generate rather than retrieve: Large language models are strong context generators. Proceedings of the International Conference on Learning Representations (ICLR). https://doi.org/10.48550/arXiv.2209.10063

Zhao, P., Zhang, H., Yu, Q., Wang, Z., Geng, Y., Fu, F., Yang, L., Zhang, W., & Cui, B. (2024). Retrieval-augmented generation for AI-generated content: A survey. arXiv preprint. https://doi.org/10.48550/arXiv.2402.19473