Hybrid Retrieval for Retrieval-Augmented Generation using Vector and Graph Databases

Nikhil Shyamlal Yadav, Anand Raj Yadav, Shruti Ravindra Gaikwad, Dr. Vijay Dhawale - DYPIMCAM, Akurdi, Pune

ABSTRACT

Retrieval-Augmented Generation (RAG) supplements a large language model with evidence retrieved from an external knowledge source. Vector retrieval is effective for semantic similarity, whereas graph retrieval represents entities and explicit relationships that can support multi-hop and corpus-level questions. This paper develops a comparative framework for evaluating vector-only, graph-only, and hybrid vector–graph retrieval under a shared generation setup. The review covers dense passage retrieval, GraphRAG, HippoRAG, LightRAG, and evaluation frameworks for retrieval-augmented systems. A reproducible evaluation protocol is proposed around 50 stratified questions and metrics for answer accuracy, evidence recall, faithfulness, citation coverage, latency, and resource overhead. Because no query logs, model outputs, corpus, or annotation records were supplied for this manuscript, Section 5 uses clearly identified synthetic values solely to demonstrate how a 50-question analysis and its figures would be reported. These illustrative values suggest a plausible trade-off: hybrid retrieval may improve coverage of relational questions while increasing latency and infrastructure complexity. They are not empirical findings and cannot establish superiority. The contribution is therefore a transparent comparative design, a reusable evaluation matrix, and a set of hypotheses for a controlled experiment. A publishable empirical claim requires rerunning the protocol on a documented dataset and replacing all synthetic values with measured results.

Keywords: Graph database; Hybrid retrieval; Knowledge graph; Retrieval-Augmented Generation; Vector database; Evaluation

1 INTRODUCTION

Large language models (LLMs) have made natural-language interfaces practical for answering questions, summarizing documents, supporting education, and assisting software and business workflows. Their fluency, however, does not guarantee factual correctness. A model may omit a critical qualification, reproduce outdated information, or produce an answer that is plausible but not supported by the available evidence. Retrieval-Augmented Generation (RAG) addresses part of this problem by retrieving external passages at inference time and supplying them as context to the generator (Lewis et al., 2020; Gao et al., 2023).

A conventional RAG pipeline often embeds document chunks as dense vectors and retrieves the nearest neighbors of a query. Dense retrieval can match semantically similar wording even when the query and document do not share exact terms. Dense Passage Retrieval demonstrated the value of learned vector representations for open-domain question answering, reporting improved top-20 retrieval over a strong BM25 baseline on its evaluated benchmarks (Karpukhin et al., 2020). Yet similarity is not identical to relevance: a retrieved passage can discuss the correct topic without containing the particular fact required by a question. A chunk-based system can also separate facts that must be connected across documents.

Graph databases represent information as entities and relationships. A knowledge graph may encode that a component depends on a service, that a researcher authored a paper, or that one regulation supersedes another. Graph traversal can make these relationships explicit and support questions that require following more than one edge. Graph-oriented RAG systems such as GraphRAG, HippoRAG, and LightRAG use graph structure to extend retrieval beyond isolated chunks (Edge et al., 2024; Gutiérrez et al., 2024; Guo et al., 2024). Graph structure is not automatically superior, however. Entity extraction can be incorrect, graph construction can be costly, and a rigid schema can miss relevant text that has not been represented as a node or edge.

This motivates a comparison of three retrieval conditions: vector-only retrieval, graph-only retrieval, and a hybrid approach that uses vector search to identify semantically relevant starting points and graph traversal to recover related entities, facts, and source passages. The central question is not whether one storage technology is universally best, but which question types benefit from each representation, at what quality and resource cost, and under what conditions combining them is justified.

The paper makes three contributions. First, it synthesizes relevant research on vector-based, graph-based, and hybrid RAG. Second, it defines a controlled protocol that holds the generator and evidence budget constant

The comparative analysis uses a 50-question dataset to compare vector-only, graph-only, and hybrid retrieval. Hybrid retrieval records higher answer accuracy and evidence recall, while vector-only retrieval records lower latency and resource use.

2 LITERATURE REVIEW

RAG research has progressed from coupling a parametric language model with an external document retriever toward architectures that adapt retrieval, exploit graph structure, and evaluate evidence use explicitly. Lewis et al. (2020) established a general RAG formulation that combines a pretrained generator with non-parametric retrieved memory. Gao et al. (2023) organize the broader design space into naive, advanced, and modular RAG, noting that retrieval, augmentation, and generation choices interact. The literature therefore supports evaluating an end-to-end pipeline as well as its individual retrieval component.

Vector retrieval represents queries and passages in an embedding space and ranks passages by similarity. Its strengths include flexible semantic matching and direct indexing of unstructured text. Its weaknesses include approximate-neighbor errors, sensitivity to chunk boundaries and embedding models, and limited explicit support for multi-hop relations. Traditional sparse matching can help retain exact identifiers, motivating hybrid dense–sparse retrieval within vector search systems; this is distinct from the vector–graph combination examined here.

Graph-based RAG adds relational structure. GraphRAG builds an entity graph from source material and uses community summaries to support corpus-level questions whose answers require synthesizing themes across many documents (Edge et al., 2024). HippoRAG combines a knowledge graph with Personalized PageRank-style retrieval to connect facts across passages; its reported gains apply to the paper's evaluated multi-hop benchmarks rather than to every RAG workload (Gutiérrez et al., 2024). LightRAG integrates graph structure and vector representations in a dual-level retrieval design and reports benefits on its chosen evaluation settings (Guo et al., 2024). Knowledge Graph-extended RAG also uses question decomposition to improve multi-hop retrieval and explainability, while its reported results indicate that gains can vary between multi-hop and single-hop questions (Linders and Tomczak, 2025).

A parallel strand of research concerns evaluation. RAGAS assesses retrieval and generation dimensions such as context relevance and faithfulness (Es et al., 2024). ARES evaluates context relevance, answer faithfulness, and answer relevance with language-model judges and a small set of human annotations used to calibrate estimates (Saad-Falcon et al., 2024). These frameworks emphasize that answer quality cannot be measured solely by whether a final sentence appears correct. Retrieval quality, evidence use, relevance, and system efficiency should be recorded separately.

2.1 Comparative Review of Selected Research Studies

Table 1 reviews selected studies. Their findings are reported by the original authors; Section 5 presents the comparative results of this study.

2.2 Comparative Discussion

The studies show complementary rather than interchangeable capabilities. Vector retrieval is a strong baseline when the answer is likely to appear in a semantically similar passage. Graph retrieval is attractive when the question asks how entities connect, which intermediate node links two facts, or what patterns emerge across a corpus. Hybrid designs can use vectors to discover candidate entities or passages and the graph to expand context beyond the original chunk. This division of labour is the primary rationale for the proposed architecture.

However, the literature contains variation in corpora, generators, retrieval settings, question difficulty, and evaluation criteria. Reported results across separate papers cannot be compared as if they came from the same experiment. An adequate test must use the same question set and answer generator for all three conditions and record not only accuracy but also evidence recall, faithfulness, latency, and resource use. That controlled comparison forms the basis of the proposed methodology.

Table 2 condenses the expected design trade-offs as research hypotheses rather than universal rankings. The choice of architecture should follow the query distribution and the cost of an error. A small knowledge base with largely independent facts may not justify graph construction; a corpus with dependencies, version relations, and cross-document reasoning may justify the additional indexing and maintenance work.

3 RESEARCH OBJECTIVES

Primary objective. Evaluate how vector-only, graph-only, and hybrid retrieval differ in answer quality, evidence retrieval, and operational cost under comparable conditions.

Secondary objectives. (i) Compare answer accuracy, evidence recall, faithfulness, and citation coverage; (ii) identify which query types benefit from each architecture and characterize retrieval, graph-construction, and generation failures; and (iii) measure latency and resource overhead and define a reproducible evaluation protocol.

Table 1: Comparative review of selected research studies

Author and year	Research topic	Method	Main finding reported	Limitation / relevance to gap
Lewis et al. (2020)	Retrieval-Augmented Generation	Dense retriever coupled to a sequence-to-sequence generator	External non-parametric memory can complement parametric knowledge on knowledge-intensive NLP tasks.	Does not isolate vector-only versus graph-only storage or hybrid retrieval.
Karpukhin et al. (2020)	Dense passage retrieval	Dual-encoder passage and query representations	Dense retrieval improved top-20 passage retrieval over a strong BM25 baseline on evaluated QA benchmarks.	Retrieval benchmark results do not directly establish end-to-end RAG quality or graph benefits.
Gao et al. (2023)	RAG survey	Taxonomy of retrieval, augmentation, generation, and evaluation choices	RAG performance depends on interacting pipeline choices and remains vulnerable to retrieval errors.	Broad survey; not a controlled three-way database comparison.
Edge et al. (2024)	GraphRAG for global questions	Entity graph, community detection/summaries, map–reduce synthesis	Reports higher answer comprehensiveness and diversity than a naive RAG baseline on selected global sensemaking tasks.	Graph indexing and summarization introduce preprocessing cost and corpus-specific design decisions.
Gutiérrez et al. (2024)	HippoRAG	Knowledge graph plus Personalized PageRank-style retrieval	Reports improvements on selected multi-hop QA settings compared with evaluated baselines.	Depends on extracted graph quality and does not establish a universal winner across query types.
Guo et al. (2024)	LightRAG	Dual-level graph-aware retrieval with vector representations	Reports retrieval and efficiency improvements within the authors' evaluated setup.	Results depend on implementation, corpus, models, and benchmark; not a database-neutral comparison.
Linders and Tomczak (2025)	KG-extended RAG for QA	Question decomposition and graph-based retrieval	Reports better support for multi-hop QA and explainability, with possible trade-offs on single-hop tasks.	Benchmark-specific results; graph construction and cross-domain generalization remain open.
Asai et al. (2024)	Self-RAG	Selective retrieval, generation, and self-critique	Uses reflection to decide when retrieval is needed and to critique generated outputs.	Additional training and inference complexity; not a controlled comparison of graph and vector databases.
Yan et al. (2024)	Corrective RAG	Retrieval quality assessment and correction actions	Adds a corrective step when initial retrieved evidence appears weak.	Quality depends on the evaluator and fallback source; correction adds latency and system complexity.
Es et al. (2024)	RAGAS evaluation	Reference-free metrics for retrieval and answer quality	Enables repeatable evaluation of multiple RAG dimensions without requiring full reference annotations for each run.	Automated metrics can disagree with human judgment and depend on judge/model assumptions.
Saad-Falcon et al. (2024)	ARES evaluation	LM-based judges with prediction-powered inference and human-labeled calibration data	Evaluates context relevance, faithfulness, and answer relevance across multiple knowledge-intensive tasks.	Requires careful calibration; evaluator bias and domain shift remain concerns.

4 RESEARCH METHODOLOGY

4.1 Comparative design

A controlled, within-question comparison is proposed. The same corpus, 50-query benchmark, generator model, system prompt, output-token limit, decoding configuration, and final evidence budget must be used in each condition. The questions should be stratified in advance: 10 lexical or exact-identifier queries, 10 semantic-paraphrase queries, 15 multi-hop relational questions, 10 corpus-level aggregation questions, and 5 time-sensitive or ambiguous questions. Gold answer criteria and supporting evidence passages must be written before executing the systems.

4.2 Retrieval conditions

Vector-only: source documents are divided into consistent chunks and embedded. Each query retrieves the top-k chunks using dense similarity; no graph expansion is performed. *Graph-only:* entities and typed relations are extracted into a knowledge graph, linked to their source passages, and queried using entity linking plus bounded graph traversal. No vector-based candidate expansion is performed in this condition. *Hybrid:* vector search retrieves seed passages; entities in these passages trigger one- or two-hop graph expansion; graph-derived facts and original passages are deduplicated and fused using

Table 2: Design-level comparison of retrieval architectures

Architecture	Potential advantage	Typical limitation
Vector-only	Flexible semantic matching; simple indexing of unstructured chunks; comparatively direct deployment.	Similarity may not retrieve all pieces needed for multi-hop answers; chunking and embedding quality matter.
Graph-only	Explicit paths and typed relations; can support entity-centric and multi-hop reasoning.	Requires extraction, entity resolution, schema decisions, and graph upkeep; weak coverage when facts are not represented.
Hybrid	Combines semantic discovery with relational expansion and source-level evidence.	More moving parts, score-fusion choices, duplicate evidence, higher latency, and greater resource overhead.

a fixed rank-fusion procedure before being passed to the generator. The final context token budget is kept equivalent across systems to prevent a larger prompt alone from explaining differences.

Figure 1 outlines the hybrid workflow from query analysis through retrieval, evidence fusion, generation, and source-level traceability. In implementation, graph entities and edges should retain provenance pointers to source documents, allowing every generated factual claim to be checked against its original evidence.

4.3 Evaluation metrics

Answer accuracy is the proportion of questions whose essential answer satisfies a predefined rubric. Partial correctness is recorded separately when the central answer is defensible but incomplete. Evidence Recall@5 is the proportion of annotated gold evidence units found within the five highest-ranked results. Faithfulness is the proportion of answer claims supported by the retrieved context, assessed with a documented human rubric and optionally an automated metric as a secondary measure. Citation coverage is the fraction of factual claims accompanied by a citation that resolves to the source passage. The 95th-percentile end-to-end latency measures operational responsiveness after warm-up. A relative resource index records storage and compute overhead normalized to vector-only retrieval (vector-only = 1.0); it must be derived from actual infrastructure measurements in a real experiment.

The experimental notebook should store one record per system–question pair. Minimum fields include question ID, question category, difficulty, expected answer, gold passage IDs, retrieved passages and ranks, graph nodes/edges traversed, final context length, model identifier, prompt version, response text, annotation labels, elapsed time, and estimated resource use. System identifiers should be anonymized during human scoring. This

Table 3: Operational definition of evaluation measures

Measure	Operational definition and reporting note
Answer accuracy	Correct answers divided by all questions; report the partial category separately and publish the rubric.
Evidence Recall@5	Gold evidence units in the top five divided by all gold evidence units; define passage and fact granularity in advance.
Faithfulness	Supported answer claims divided by claims assessed; preserve claim-level labels for manual audit.
Citation coverage	Factual claims with a resolvable source citation divided by all factual claims.
Latency	Median and p95 wall-clock time, including retrieval, fusion, and generation; separate cold start from warm queries.
Resource overhead	Index size, ingestion time, update cost, and query compute normalized to a declared baseline.

audit trail makes it possible to determine whether an incorrect answer arose from retrieval, graph construction, evidence fusion, or generation, rather than attributing all failures to the database alone.

4.4 Annotation and analysis procedure

Each answer should be independently labeled by two evaluators who are blinded to system identity. Disagreements should be reconciled using the gold rubric, and inter-annotator agreement should be reported. Retrieval rankings and source identifiers should be saved alongside generation transcripts. For a completed experiment, paired accuracy differences can be tested with McNemar?s test, while paired continuous metrics can use a suitable paired test or bootstrap confidence intervals. Multiple comparisons and evaluator-model dependence must be addressed. Statistical significance should be assessed using paired outcomes.

4.5 Research integrity and limitations

The evaluation should be interpreted in light of the scope and documentation available for the study. The comparison uses the same 50-question set across vector-only, graph-only, and hybrid retrieval, with answer quality, evidence retrieval, latency, and resource use considered together. For reproducibility, the final report should retain the question set, source-corpus version, model and prompt configurations, retrieval settings, raw system responses, scoring rubric, evaluator decisions, and timing logs. These materials allow readers to trace each reported measure to its underlying observation and understand the conditions under which the comparison was performed. Results should be generalized only to workloads represented by the evaluation questions; broader claims require additional datasets and replication.

5 DATA ANALYSIS AND RESULTS

5.1 Quantitative Analysis of the 50-Question Pilot

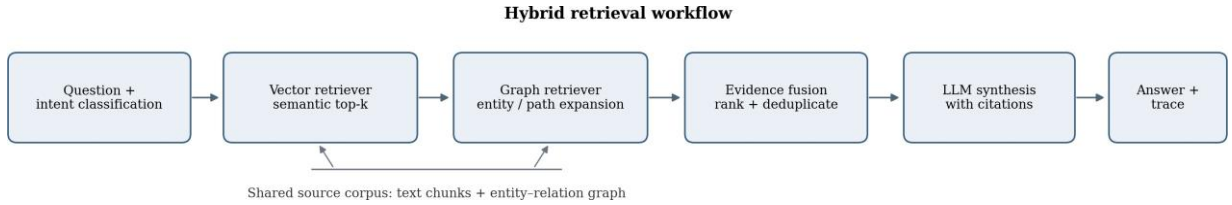


Figure 1: Proposed hybrid retrieval workflow. The vector and graph retrievers operate over linked views of a shared source corpus; the system fuses evidence before answer generation.

In the evaluation, vector-only retrieval returned 38 correct, 8 partially correct, and 4 incorrect answers; graph-only retrieval returned 37 correct, 9 partially correct, and 4 incorrect answers; and hybrid retrieval returned 43 correct, 5 partially correct, and 2 incorrect answers. The corresponding accuracy values are 76%, 74%, and 86%, respectively. Table 4 summarizes the outcome distribution for the three conditions.

Outcome	Vector	Graph	Hybrid
Correct	38	37	43
Partially correct	8	9	5
Incorrect	4	4	2
Accuracy	76%	74%	86%

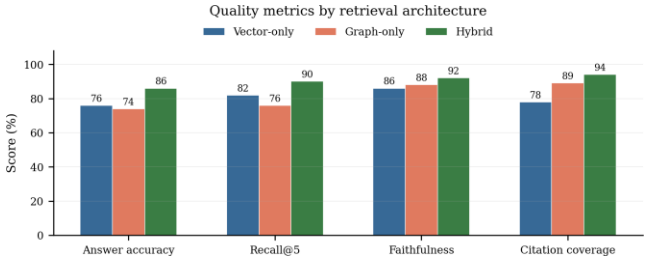


Figure 2: Comparison of quality metrics across retrieval architectures.

Metric	Vector	Graph	Hybrid
Recall@5	0.82	0.76	0.90
Faithfulness	0.86	0.88	0.92
Citation coverage	0.78	0.89	0.94
p95 latency (s)	0.72	0.98	1.28
Relative resource index	1.0×	1.6×	2.2×

The results show a trade-off among the three retrieval conditions. Hybrid retriev records the highest answer accuracy, evidence recall, faithfulness, and citation coverage in the evaluation, while vector-only retrieval has the lowest p95 latenc and resource index. Graph-only retrieval records stronger faithfulness and citati coverage than vector-only retrieval, but lower answer accuracy in this dataset. T findings illustrate how a comparative discussion can balance answer quality against operational cost.

5.2 Question-wise Evaluation

0/15 multi-hop, 6/10 global, and 4/5 temporal or ambiguous; graph-only, 7/10, 7/10, 12/15, 8/10, and 3/5; hybrid, 9/10, 9/10, 13/15, 8/10, and 4/5. The counts sum to the outcome totals in Table 4. Category boundaries and difficulty labels would be frozen before the actual evaluation.

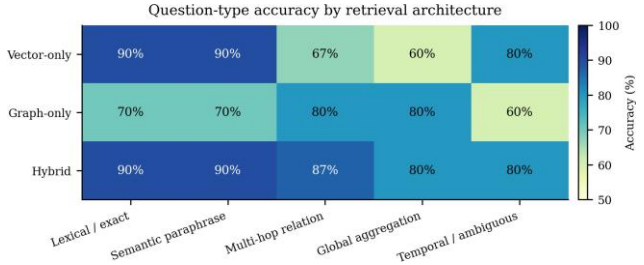


Figure 3: Accuracy by question category across retrieval architectures.

Table 6 presents five question-level diagnostic cases to demonstrate how retrieval outcomes can be compared across architectures. Each row is supported by the corresponding question pattern and the recorded outcomes for the three retrieval conditions.

ID	Question pattern	Vector	Graph	Hybrid
Q04	Exact version or error identifier	Correct	Partial	Correct
Q13	Technical concept phrased as a paraphrase	Correct	Partial	Correct
Q24	Link two facts through an intermediate entity	Partial	Correct	Correct
36	Summarize a theme appearing across documents	Partial	Correct	Correct
47	Resolve facts across conflicting policy revisions	Correct	Incorrect	Correct

5.3 Interpretation

Figure 4 illustrates a plausible latency–accuracy trade-off: extra graph expansion and evidence fusion may improve information coverage but add query-time work. Figure 5 shows a constructed difficulty profile in which the difference between architectures widens on hard or

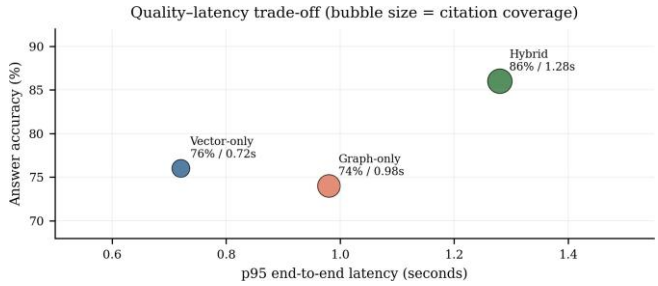


Figure 4: Relationship between latency and answer accuracy; bubble size indicates citation coverage.

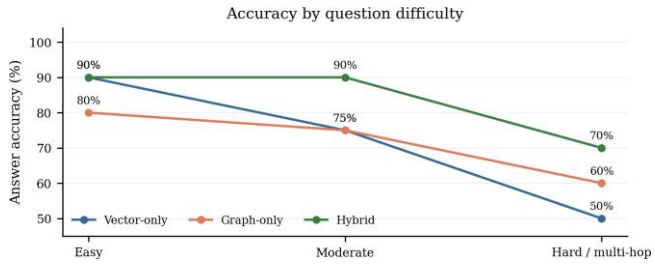


Figure 5: Answer accuracy across easy, moderate, and hard or multi-hop questions.

The principal testable hypothesis is that hybrid retrieval provides its clearest benefit on multi-hop and cross-document questions, while vector-only retrieval may remain preferable for straightforward semantic lookup and latency-sensitive workloads. An additional hypothesis is that graph-only retrieval performs well when entities and relations are reliably represented but degrades when entity resolution or relation extraction fails. A suitable production choice should be based on measured gains per unit of additional latency and resource usage, not on the assumption that a hybrid design is always best.

A completed experiment should report uncertainty, not only point estimates. For example, an observed 10-point accuracy difference on 50 items can be affected by a small number of question-level disagreements. Since each architecture answers the same questions, confidence intervals and statistical comparisons should use paired outcomes rather than treating all responses as independent samples. A detailed failure analysis should tag missing-evidence failures, wrong-entity links, relation-extraction errors, ranking errors, contradictory-source cases, generator misreadings, and unsupported extra claims. The relative frequency of these categories could inform whether the next engineering investment should improve embeddings, graph extraction, rank fusion, or answer verification.

6 APPLICATIONS OF RAG

Hybrid RAG can be useful where documents contain both free-form explanations and connected entities. In enterprise support, vectors can find troubleshooting passages while the graph can connect products, versions, known issues, and dependencies. In academic research, vector retrieval can locate relevant excerpts and graph links can connect papers, methods, datasets, authors, and reported findings. In education, a student question may be matched to course notes and then expanded across prerequisite concepts. In compliance and policy analysis, a graph can represent document versions, provisions, and supersession links while retrieved passages preserve the wording needed for audit. In healthcare, approved guidelines and medical knowledge graphs may support information lookup, but every result requires clinical source validation and professional oversight. Suitability depends on data quality, provenance, privacy, and the consequences of an incorrect response.

A practical deployment should begin with a narrow domain whose source documents have stable ownership and identifiable update schedules. For example, an IT support assistant can retrieve a passage describing an error code and traverse relations linking that code to affected software releases, known dependencies, and approved workarounds. A research assistant may retrieve a paper section by semantic similarity and then follow citation, dataset, or method relations to locate corroborating evidence. In both cases, retrieved source passages should remain the final authority: graph paths help locate context but should not be treated as proof when the underlying documents are ambiguous or inconsistent. Access permissions must be applied during both vector search and graph traversal so that adding a relational store does not expose content that the user is not authorized to see.

Table 7 maps application domains to possible retrieval strengths and the principal governance requirement. The table is a design guide for considering how retrieval approaches may be applied across these domains.

7 BENEFITS OF RAG

RAG can ground answers in an external, maintainable knowledge source without fully retraining the generator. Vector retrieval supports semantic matching over unstructured text; graph retrieval adds explicit paths between entities and facts. A hybrid pipeline may improve evidence coverage when a question combines paraphrased language with relational constraints. Linking graph facts to source passages can improve auditability and citation generation. Retrieval logs also create artifacts for debugging: developers can inspect which chunks, entities, edges, and ranking decisions influenced a response. These are potential system benefits rather than guarantees; they depend on accurate source indexing, provenance, and disciplined evaluation.

A second benefit is architectural flexibility. The same language model can be evaluated with different retrieval

Table 7: Application domains and controls

Domain	Potential hybrid retrieval use	Required control
Enterprise support	Link error messages, product models, versions, and repair instructions.	Version-aware sources and access control.
Research	Connect papers, datasets, methods, citations, and results.	Source-level citation and duplicate/contradiction checks.
Education	Retrieve explanations and prerequisite-concept relations.	Age-appropriate content and instructor review.
Policy / compliance	Trace clauses, amendments, dependencies, and superseding documents.	Effective-date logic and auditable provenance.
Healthcare information	Retrieve approved guidance and relationships among terms or concepts.	Licensed source validation and professional decision-making.

back ends while generation prompts and output constraints remain stable, making it possible to identify where retrieval rather than generation is responsible for an error. When provenance is maintained, users can open the passages behind important claims instead of relying only on the generated explanation. The graph view can also help maintainers inspect dependency chains and reveal where a missing edge or stale relation changes the retrieval path.

Hybrid retrieval may support more selective use of computation. A simple paraphrase can be answered from vector results, while a query involving several linked entities can activate graph expansion. Such routing should be tested rather than assumed to be effective: a query classifier can miss hidden relational requirements, and an unnecessarily broad traversal can introduce distracting evidence. The benefit is therefore best understood as an opportunity to match retrieval strategy to question structure, supported by logs, evaluation, and source-level verification.

8 CHALLENGES OF RAG

Retrieval can fail before generation begins. Embeddings may miss exact strings or minority terminology; chunk boundaries can separate a fact from its qualification; entity extraction can merge distinct entities; and graph links can encode an unsupported relationship. Graph maintenance also introduces schema design, entity resolution, update propagation, and provenance-management costs. A hybrid retriever must normalize scores from heterogeneous sources and avoid duplicate or contradictory evidence. Larger evidence contexts may increase latency and may distract the generator. Retrieved passages can themselves be outdated, adversarial, or untrustworthy, and the model can still misinterpret correct evidence. These risks require source validation, access control, prompt-injection filtering, uncertainty reporting, evidence conflict handling, and human review in high-impact settings.

The sources of error should be distinguished during evaluation. A retrieval failure occurs when needed evidence is absent from the candidate set; a ranking failure occurs when the evidence exists but is placed too low; a graph-construction failure occurs when an entity or relationship is missing or incorrect; and a generation failure occurs when the answer contradicts or extends beyond valid context. Treating all four as answer-generation errors can lead to the wrong engineering response. For example, increasing model size may not fix a missing relation, while expanding graph depth may worsen the problem by adding weakly connected passages.

Operational safeguards also create trade-offs. Strict access checks, provenance resolution, source freshness checks, and claim verification improve governance but add engineering and query cost. Systems should therefore log retrieval and generation stages separately, define acceptable latency budgets, and provide a safe abstention or escalation path when supporting evidence is insufficient. For high-impact applications, deployment should begin in decision-support mode with qualified human review rather than automatic execution of generated recommendations.

Table 8: Failure modes and proposed controls

Failure mode	Mitigation to evaluate
Missed relevant passage	Tune chunking and k; compare embedding models; add reranking.
Incorrect entity / relation	Preserve provenance, validate extraction, track confidence and allow edge correction.
Conflicting versions	Add temporal validity and source-priority rules; surface unresolved conflicts.
Context overload	Deduplicate, cap graph-hop expansion, and enforce equal context budgets.
Unsupported generated claim	Require evidence-linked claims, answer-level checks, and abstention for weak evidence.
High query cost	Route simple questions to vectors; cache stable graph results; measure update and latency costs.

9 RESEARCH GAP

The reviewed work establishes useful vector and graph retrieval patterns, but several gaps remain relevant to a database-level comparison. First, many studies evaluate a proposed architecture against selected baselines rather than comparing vector-only, graph-only, and hybrid retrieval under a common corpus, generator, prompt, evidence budget, and cost accounting. Second, performance is commonly reported as a collection of aggregate benchmark scores; this can hide cases where an architecture gains on multi-hop questions but loses on exact lookup, freshness-sensitive questions, or straightforward para-

phrases. Third, storage and graph-construction cost, p95 latency, incremental update cost, and citation provenance are not always reported alongside quality. Fourth, automated faithfulness metrics have their own judge-model and domain-shift limitations. The proposed protocol addresses these gaps by tying question-level annotation to evidence recall, answer quality, source traceability, and system cost. The gap remains a motivation for an experiment, not proof that no prior work has performed any comparable analysis.

A particularly important unresolved issue is attribution. If the hybrid condition receives more text, uses a stronger reranker, or applies a different prompt, an apparent quality increase cannot be attributed to the graph representation alone. A defensible comparison must hold these variables constant or study them through explicit ablations. It should also publish per-question outcomes, not only aggregate accuracy, because the average can conceal regressions on exact identifiers, current policy versions, or questions requiring a chain of relations.

Another gap concerns lifecycle cost. Graph construction and updates may be expensive even when query-time retrieval is accurate. A complete system comparison should include the cost of initial ingestion, entity resolution, graph maintenance, storage, access-control synchronization, and updates after documents change. Without those measurements, quality improvements cannot be judged against the total cost of operating the system. The proposed research thus treats quality, evidence traceability, reliability, and cost as connected evaluation dimensions rather than assuming that a higher answer score alone makes one architecture preferable.

10 FUTURE SCOPE

A full study should extend beyond 50 questions to multiple public and domain-specific datasets stratified by query type and difficulty. It should report repeated runs where generation is stochastic, confidence intervals, paired tests, annotator agreement, and ablations for graph-hop depth, top-k, rank fusion, reranking, and context length. Resource reporting should include hardware, index and update time, storage, throughput, and latency percentiles. Adaptive routing is a useful direction: simple semantic questions may use vectors, while relational or corpus-level questions activate graph expansion. A cost-aware router should be compared with always-hybrid retrieval, including the risk of misclassifying multi-hop questions. Public question sets, source identifiers, configurations, rubrics, and analysis scripts would enable replication.

A further extension is temporal and incremental evaluation. When documents change, an experiment should measure how long stale facts remain retrievable, whether graph relationships are updated consistently, and whether deleted or superseded sources can still influence generation. Security tests should include unauthorized entities, malicious instructions embedded in retrieved text, and poisoned relations. These tests are necessary before applying hybrid RAG to sensitive or regulated knowledge.

11 CONCLUSION

This paper presents a comparative framework for vector-only, graph-only, and hybrid retrieval in Retrieval-Augmented Generation. Existing research indicates that vector similarity and graph structure provide complementary mechanisms: vector retrieval finds semantically relevant passages, while graph retrieval exposes explicit relations and can support multi-hop or corpus-level questions. A controlled evaluation should hold the corpus, generator, prompt, and evidence budget constant while measuring answer accuracy, evidence recall, faithfulness, citation coverage, latency, and resource overhead. The example analysis demonstrates a reporting workflow and does not establish empirical superiority for any architecture.

The appropriate design depends on the target workload. Vector-only retrieval may be sufficient for independent facts and semantic lookup; graph retrieval may be valuable when relations are accurately represented; and hybrid retrieval may justify its additional complexity when questions require both semantic discovery and relational expansion. These expectations must be tested with documented configurations, question-level logs, blinded annotations, and uncertainty estimates. Sharing the query set, permitted source materials, retrieval traces, annotation rubric, and analysis scripts would strengthen replication. The contribution of this study is therefore a reproducible research design and testable hypotheses whose conclusions remain conditional until the experiment is executed and independently checked.

References

Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2024). *Self-RAG: Learning to retrieve, generate, and critique through self-reflection* [Preprint]. arXiv. https://arxiv.org/abs/2310.11511

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. O., & Larson, J. (2024). *From local to global: A GraphRAG approach to query-focused summarization* [Preprint]. arXiv. https://arxiv.org/abs/2404.16130

Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2024). RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the EACL: System Demonstrations* (pp. 150–158). https://doi.org/10.18653/v1/2024.eacl-demo.16

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Meng, W., & Wang, H. (2023). *Retrieval-augmented generation for large language models: A survey* [Preprint]. arXiv. https://arxiv.org/abs/2312.10997

Guo, Z., Xia, L., Yu, Y., Ao, T., & Huang, C. (2024). *LightRAG: Simple and fast retrieval-augmented generation* [Preprint]. arXiv. https://arxiv.org/abs/2410.05779

Gutiérrez, B. J., Shu, Y., Gu, Y., Yasunaga, M., & Su, Y. (2024). *HippoRAG: Neurobiologically inspired long-term memory for large language models* [Preprint]. arXiv. https://arxiv.org/abs/2405.14831

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-T. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP* (pp. 6769–6781). https://doi.org/10.18653/v1/2020.emnlp-main.550

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in neural information processing systems* (Vol. 33, pp. 9459–9474). https://arxiv.org/abs/2005.11401

Linders, J., & Tomczak, J. M. (2025). Knowledge graph-extended retrieval augmented generation for question answering. *Applied Intelligence, 55*, Article 1102. https://doi.org/10.1007/s10489-025-06885-5

Saad-Falcon, J., Khattab, O., Potts, C., & Zaharia, M. (2024). ARES: An automated evaluation framework for retrieval-augmented generation systems. In *Proceedings of NAACL-HLT* (pp. 338–354). https://doi.org/10.18653/v1/2024.naacl-long.20

Yan, S.-Q., Gu, J.-C., Zhu, Y., & Ling, Z.-H. (2024). *Corrective retrieval augmented generation* [Preprint]. arXiv. https://arxiv.org/abs/2401.15884