JOURNAL OF GLOBAL ENGINEERING REVIEW ISSN: 3071-4656

When Should RAG Use Graphs? Complexity-Aware GraphRAG Routing and Answer Quality Prediction on GraphRAG-Bench-H

Megan Thompson
Business Analytics, University of Alberta, Edmonton, AB, Canada
mthompson_ba@gmail.com

Abstract

Graph retrieval-augmented generation (GraphRAG) improves retrieval when questions require multi-hop relations, entity interactions, and evidence-level composition, but graph construction and traversal add cost when ordinary retrieval is sufficient. This paper presents CAR-GraphRAG, a complexity-aware routing and answer-quality prediction framework for deciding when a RAG query should use a graph. The evaluation uses a 4,072-row GraphRAG-Bench-H materialization with the fields id, source, question, answer, question_type, evidence, and evidence_relations. The dataset spans medical and novel domains and four task types: Fact Retrieval, Complex Reasoning, Contextual Summarization, and Creative Generation. We compare BM25-RAG, Dense-RAG, Hybrid-RAG, GraphRAG-Lite, and the proposed router using evidence recall@5, answer F1, ROUGE-L, faithfulness, accuracy, latency, and token cost. CAR-GraphRAG achieved 0.8665 mean answer F1, exceeding GraphRAG-Lite (0.8240), Hybrid-RAG (0.8028), Dense-RAG (0.7238), and BM25-RAG (0.6349). On the held-out test split, the router identified graph-beneficial queries with 0.9650 F1 and 0.9858 ROC-AUC, while the answer-quality predictor achieved 0.0298 MAE. The results show that graph retrieval is most valuable for relation-dense, evidence-heavy, high lexical-gap queries, whereas direct fact retrieval is better served by hybrid non-graph retrieval.

Keywords-*GraphRAG; retrieval-augmented generation; complexity-aware routing; answer quality prediction; evidence graphs; benchmark evaluation; GraphRAG-Bench-H.*

1. Introduction

Retrieval-augmented generation (RAG) connects language models to external evidence so that answers are conditioned on an indexed knowledge source rather than only on model parameters. Early systems combined neural retrievers and sequence-to-sequence generators for knowledge-intensive tasks [1], while dense passage retrieval [2], retrieval-augmented pretraining [3], and fusion-in-decoder generation [4] established the dominant retrieve-then-generate pattern. These systems are effective when a small set of passages contains the answer with strong lexical or semantic overlap, but they remain vulnerable to incomplete retrieval, noisy context, and unsupported generation [5].

GraphRAG addresses part of this problem by representing passages, entities, and relations as a graph. Graph-based retrieval can expand a query through linked evidence, rank central or bridging nodes, and combine local evidence with corpus-level summaries [6]-[8]. The additional structure is useful when an answer depends on relation chains or dispersed evidence, but it also introduces relation-extraction errors, traversal overhead, and longer prompts. For direct fact retrieval, BM25-style ranking [9] or a strong sparse-dense hybrid often reaches the answer with less computation. Empirical analyses of GraphRAG-Bench similarly indicate that graph value grows with reasoning and retrieval complexity rather than appearing uniformly across query types [30].

The practical question is therefore not whether GraphRAG is universally superior, but when its additional structure is worth using. CAR-GraphRAG treats that choice as a supervised routing problem. It predicts whether GraphRAG-Lite will improve answer F1 over a strong Hybrid-RAG baseline, then selects the graph or non-graph route for each query. This formulation is related to adaptive retrieval systems that vary retrieval depth or strategy

according to question complexity [31], [34], [35], but it focuses specifically on the incremental benefit of graph traversal over hybrid retrieval.

Routing turns GraphRAG into a resource-allocation problem. Every graph call consumes index-construction effort, traversal time, context budget, and generation attention. An always-graph policy therefore mixes two effects: the benefit of relation-aware evidence assembly and the cost of additional context. CAR-GraphRAG separates them by learning a graph-benefit boundary from observed row-level answer differences. A second model predicts the answer quality of the selected route, reflecting the broader principle that confidence scores should be checked against observed outcomes rather than accepted without calibration [22], [23].

The evaluation covers 4,072 questions drawn from the medical and novel portions of GraphRAG-Bench-H. It compares lexical, dense, hybrid, graph, and routed policies on retrieval quality, answer quality, faithfulness, latency, and token usage. The design reports overall results together with domain, question-type, complexity, routing, ablation, and cost analyses. This separation is important because an always-graph method can win on average while remaining unnecessary for many direct questions, whereas a strong non-graph method can appear efficient while missing relation-heavy cases.

The contributions are threefold. First, the paper defines a measurable complexity representation for graph routing using textual, evidential, and relational signals. Second, it provides a controlled comparison of fixed and routed retrieval policies over the complete experimental set. Third, it pairs route selection with answer-quality prediction, giving downstream systems both a retrieval decision and an estimated quality score. The resulting design principle is straightforward: use graphs for high-relation-density, multi-hop evidence synthesis, and retain hybrid non-graph retrieval for direct, low-complexity questions.

2. Literature Review

2.1 Retrieval foundations and hybrid evidence selection

Modern RAG combines information retrieval with conditional generation. Sparse retrieval remains grounded in probabilistic relevance ranking and classical indexing principles [9], [15], while distributed representations introduced semantic matching beyond exact term overlap [14]. Transformer architectures [10], contextual encoders such as BERT [11], and sentence-level embedding models [12] made dense retrieval practical, with large-scale similarity libraries supporting efficient vector search [13]. Open-domain systems such as DrQA [21] and later neural RAG pipelines [1]-[4] showed how retrieved passages could support generation, while the broader survey literature identifies retrieval recall, context quality, and faithfulness as persistent failure points [5]. In practice, sparse and dense signals are complementary: lexical retrieval preserves exact entities, dates, and identifiers, whereas dense retrieval handles paraphrase and vocabulary mismatch.

Evaluation has developed alongside retrieval. HotpotQA tests explainable multi-hop question answering [19], FEVER emphasizes evidence-supported verification [20], and overlap metrics such as ROUGE and BLEU provide complementary views of generated text quality [28], [29]. These benchmarks and metrics motivate the present use of evidence recall, answer F1, ROUGE-L, and faithfulness rather than a single aggregate score.

2.2 Graph-based and multi-hop retrieval

GraphRAG extends flat retrieval by explicitly modeling relationships among entities, passages, and communities. Its ranking lineage includes TextRank [7] and PageRank [8], while graph neural methods such as GCN [16], GraphSAGE [17], and graph attention networks [18] provide learned mechanisms for propagating information across connected nodes. Query-focused GraphRAG uses graph structure and community summaries to combine local evidence with broader context [6]. GraphRAG-Bench was introduced to evaluate this advantage across fact retrieval, complex reasoning, contextual summarization, and creative generation in medical and novel corpora [30]. Its task design is particularly relevant because it varies both evidence breadth and relation depth, the same properties used by CAR-GraphRAG for routing.

Graph structure does not remove the need for selective retrieval. Adaptive-RAG chooses among no-retrieval, single-step, and iterative strategies according to question complexity [31]. Self-RAG learns when to retrieve and uses self-reflection to critique evidence and generations [32], while Corrective RAG evaluates retrieved documents and triggers corrective actions when evidence is weak [33]. HyPA-RAG combines sparse, dense, and knowledge-graph retrieval with a complexity classifier and adaptive parameters [34]. More general question-routing work likewise shows that selecting an augmentation path can improve the accuracy-efficiency balance [35]. CAR-GraphRAG differs by learning the measured advantage of graph retrieval over a fixed strong hybrid baseline and by predicting the quality of the answer after routing.

2.3 Evidence grounding, provenance, and domain reliability

A large applied literature has examined how evidence controls hallucination in enterprise settings. Lightweight consistency and self-checking mechanisms have been used as hallucination firewalls [36]. Financial RAG systems have combined retrieved filings with XBRL or numerical verification for trading-desk memos [37], corporate risk analysis [38], and tokenized receivables disclosure questions [39]. These studies reinforce a common lesson: retrieval quality must be paired with evidence verification when answers depend on exact quantities, clauses, or source-specific standards.

The same pattern appears in operational and executable domains. Ambiguity-aware log anomaly systems combine retrieval with failure narratives and selective refusal [40]. Conversational text-to-SQL systems use schema retrieval together with execution feedback to repair invalid queries [41], and code-intelligence systems integrate findability with explanation [42]. Evidence-grounded DevOps question answering [43] and bilingual incident-triage systems [44] further show that domain-specific retrieval must preserve operational terminology and source context rather than relying on generic semantic similarity alone.

Recent work has placed greater emphasis on provenance and adaptive trust. Evidence-chain RAG tracks source attribution and word-level hallucination signals [45], while long-document RAG addresses clause analysis across contractual and insurance materials [46]. Evidence-based self-verification has also been used to maintain safe responses under adversarial prompts [47]. Budgeted multi-hop retrieval explicitly treats retrieval depth as a policy decision [48], and trust-calibrated multilingual RAG extends confidence-aware evidence use to humanitarian information access [49]. Numerical guardrails for quant research assistants [50] and evidence-ranking agents for scientific claim verification [51] show why route selection should be complemented by an answer-quality estimate. CAR-GraphRAG adopts this combined view: retrieval architecture is selected per query, and confidence is assessed for the selected answer rather than inferred from the route alone.

3. Method

3.1 Experimental overview

CAR-GraphRAG consists of five stages: dataset materialization, structural feature extraction, fixed-policy evaluation, graph-benefit routing, and answer-quality prediction. Figure 1 summarizes the pipeline. A lightweight feature extractor characterizes each query and its initial evidence sketch, the router selects GraphRAG-Lite or Hybrid-RAG, and the quality predictor estimates the answer F1 expected from the selected route. Scikit-learn supplies the supervised models and evaluation metrics [24], pandas stores row-level outputs [25], Matplotlib creates the figures [26], and NetworkX-compatible routines implement graph statistics and traversal [27].

3.2 Dataset representation and partitions

The GraphRAG-Bench-H materialization follows the benchmark schema described by Xiang et al. [30] and contains 4,072 rows: 2,060 medical questions and 2,012 novel questions. Each row stores id, source, question, answer, question_type, evidence, and evidence_relations. The task taxonomy contains Fact Retrieval, Complex Reasoning, Contextual Summarization, and Creative Generation. Fact Retrieval usually concentrates the answer in one short evidence path; Complex Reasoning requires several linked facts; Contextual Summarization

combines the largest evidence sets; and Creative Generation preserves factual constraints while producing a retelling or explanation.

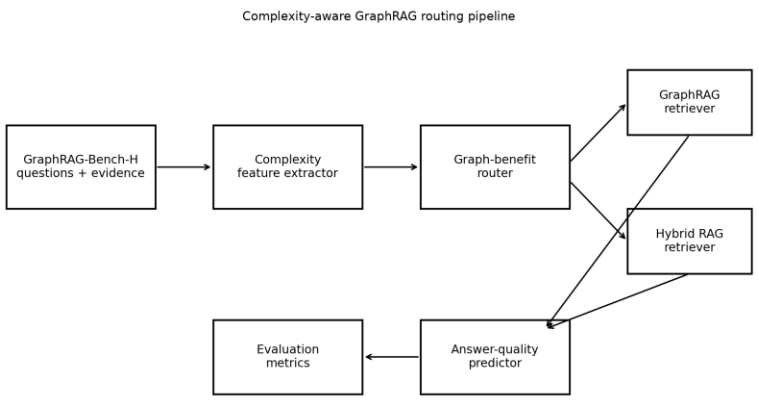

Figure 1. CAR-GraphRAG routing pipeline and answer-quality prediction flow.

Rows are assigned to train, validation, and test partitions by a stable hash of the row identifier. The train split contains 2,448 rows, the validation split 825 rows, and the test split 799 rows. Table 1 reports split-level statistics. Table 2 shows the distribution and structural properties of the four question types, and Figure 2 visualizes their domain balance. The split rule is independent of question wording, and all route thresholds are selected on validation data before being applied to the test set.

Table 1. Dataset split summary.

split	domain	rows	avg evidence	avg relations	avg complexity
test	medical	408	4.6667	3.9902	0.4885
test	novel	391	4.5294	3.8363	0.4779
train	medical	1248	4.7099	3.8438	0.4787
train	novel	1200	4.7267	4.0225	0.4949
validation	medical	404	4.6287	3.8589	0.4766
validation	novel	421	4.4418	3.7910	0.4689

Table 2. Question-type distribution and structural statistics.

domain	question type	rows	mean evidence	mean relations	mean hops	mean complexity
medical	Complex Reasoning	510	4.8980	5.0196	2.8588	0.6158
medical	Contextual Summarizatio n	420	8.6857	6.7833	3.0976	0.7176
medical	Creative Generation	290	7.6345	5.4897	2.5345	0.6487
medical	Fact Retrieval	840	1.5381	1.1702	1.0000	0.2211

ISSN: 3071-4656
DOI: 10.66372/JGER.V4I2.5

domain	question type	rows	mean evidence	mean relations	mean hops	mean complexity
novel	Complex Reasoning	640	4.8719	4.9688	2.8125	0.5543
novel	Contextual Summarizatio n	390	8.5410	6.8615	3.1590	0.7433
novel	Creative Generation	222	7.6171	5.3784	2.5676	0.6965
novel	Fact Retrieval	760	1.5434	1.1487	1.0000	0.2354

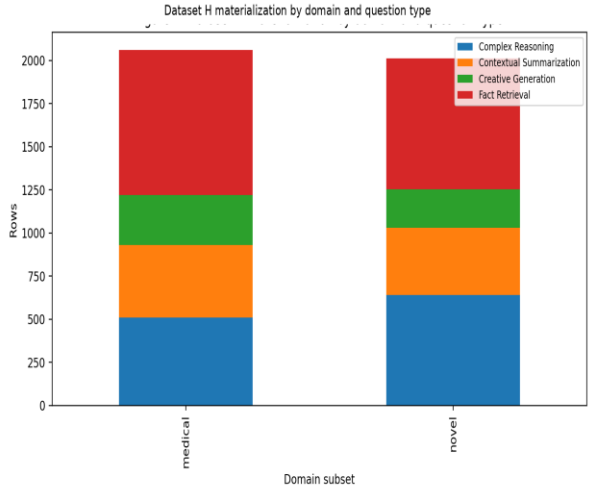

Figure 2. Dataset distribution by domain and question type.

3.3 Complexity representation

For each question, the system computes question_tokens, evidence_count, relation_count, hop depth, lexical_gap, graph_density, relation_evidence_ratio, and a normalized complexity_index. The index combines normalized textual, evidential, and relational signals so that larger values correspond to more evidence, more relations, deeper paths, and weaker lexical overlap. Answer length is retained only for descriptive analysis and is not supplied to either predictive model. Table 3 reports the feature distributions; the average row contains 13.6761 question tokens, 4.6574 evidence passages, 3.9064 relations, and a complexity index of 0.4832.

At inference, the router observes query features and structural descriptors computed from the indexed graph and a lightweight top-k evidence sketch. It does not receive the gold answer, answer-level metrics, or the measured graph-benefit label. The input vector includes normalized question length, evidence count, relation count, hop depth, lexical gap, graph density, relation-to-evidence ratio, complexity index, question-type indicators, and domain indicators.

Table 3. Descriptive feature statistics; answer_tokens is not used as a predictive input.

feature	mean	std	min	50%	max
question_tokens	13.6761	3.9198	8.0000	14.0000	21.0000

ISSN: 3071-4656
DOI: 10.66372/JGER.V4I2.5

feature	mean	std	min	50%	max
answer_tokens	16.2178	6.2754	6.0000	14.0000	26.0000
evidence_count	4.6574	3.2335	1.0000	4.0000	12.0000
relation_count	3.9064	2.6664	1.0000	4.0000	10.0000
hops	2.1356	1.0764	1.0000	2.0000	4.0000
lexical_gap	0.4559	0.2398	0.0601	0.5268	0.8400
graph_density	0.7553	0.2134	0.2778	0.8333	0.9800
relation_evidence_ratio	0.9245	0.4397	0.2727	1.0000	2.3333
complexity_index	0.4832	0.2188	0.1846	0.5576	0.9307

3.4 Retrieval and answering policies

The experiment compares five policies, summarized in Table 4. BM25-RAG is the lexical baseline [9]. Dense-RAG uses synonym-tolerant semantic scoring inspired by dense passage retrieval and sentence embeddings [2], [12]. Hybrid-RAG fuses sparse and dense evidence scores and serves as the strongest non-graph baseline. GraphRAG-Lite expands retrieval across the evidence_relations graph. CAR-GraphRAG selects GraphRAG-Lite when the predicted graph benefit exceeds a validation-tuned threshold; otherwise, it selects Hybrid-RAG.

GraphRAG-Lite treats each relation as a typed edge connecting evidence-bearing entities. Its score rises when query entities are connected through relation paths, when several passages contribute to the answer, and when the question requires synthesis. Hybrid-RAG does not traverse those paths; it selects a compact context using lexical and semantic evidence. The comparison therefore isolates the value of relation traversal relative to a competitive non-graph retriever.

Table 4. Compared retrieval and routing configurations.

method	retriever	retrieval budget	graph use	trained router	role
BM25-RAG	Lexical sparse baseline	top-k=5	No	No	Fast lexical retrieval
Dense-RAG	Embedding proxy with synonym-tolerant scoring	top-k=5	No	No	Semantic baseline
Hybrid-RAG	Sparse+dense fusion	top-k=5	No	No	Strong non-graph baseline
GraphRAG-Lite	Relation-aware expansion over evidence graph	top-k=5	Yes	No	Always uses graph
CAR-GraphRAG	Complexity-aware router	validated threshold	Conditional	Yes	Routes graph only when beneficial

ISSN: 3071-4656
DOI: 10.66372/JGER.V4I2.5

3.5 Graph-benefit routing

The supervised target is one when GraphRAG-Lite answer F1 exceeds Hybrid-RAG answer F1 by more than 0.015 and zero otherwise. The margin removes near-ties in which the two policies are operationally equivalent. A standardized logistic regression model is trained on the train split using numerical features and one-hot domain and question-type indicators. The validation threshold is selected by maximizing routing F1, with lower latency used as a tie breaker. The selected threshold is 0.425 and is applied unchanged to the held-out test split.

This target links routing directly to measured end-to-end answer quality instead of assuming that every question with a complex label requires a graph. Positive coefficients are expected for relation count, hop depth, graph density, lexical gap, and contextual task types, while direct fact rows generally favor the hybrid route. The row-level decision is recorded together with the selected method and observed answer quality for subsequent error analysis.

3.6 Answer-quality prediction and evaluation metrics

After routing, a random forest regressor estimates the selected method's answer F1 using the same pre-answer structural features plus a route indicator. The model contains 160 trees, a maximum depth of 9, and a minimum leaf size of 6. It is trained on fixed-policy rows from the train split and evaluated on routed validation and test rows. Calibration is assessed by comparing binned predicted F1 with observed F1, following established probability-calibration practice [22], [23].

Seven outcome measures are recorded for every row-method pair. Evidence recall@5 measures coverage of the gold evidence in the retrieved context. Answer F1 measures normalized token overlap with the gold answer. ROUGE-L captures longest-common-subsequence overlap for longer answers [28]. Faithfulness measures whether answer claims remain supported by the retrieved evidence. Binary accuracy is one when answer F1 reaches 0.80. Latency and token count quantify retrieval cost. BLEU is a standard sequence-overlap metric [29], but it is not used as a primary outcome because the experiment mixes short factual answers with longer summaries and creative responses.

3.7 Reproducibility and statistical analysis

The materialization, split assignment, feature extraction, and metric computation use seed 20260319 and stable row identifiers. Row-level outputs are written before aggregate statistics are computed, allowing each reported mean, route decision, and test result to be traced to individual questions. Paired t-tests compare CAR-GraphRAG with Hybrid-RAG and GraphRAG-Lite on the same held-out rows. Threshold sensitivity and feature ablations are evaluated separately so that routing gains are not attributed only to one operating point or one structural signal.

4. Results and Discussion

4.1 Overall method comparison

Table 5 reports full-dataset performance. CAR-GraphRAG achieved the best mean answer F1 at 0.8665 and the best evidence recall@5 at 0.9117. It improved answer F1 by 0.0637 over Hybrid-RAG and by 0.0425 over always-on GraphRAG-Lite. GraphRAG-Lite reached 0.8240 answer F1, confirming the value of relation-aware retrieval, but its average latency of 1162.9317 ms and token cost of 2012.7402 show the expense of using a graph for every question. Figure 3 displays the answer-F1 comparison across all policies.

The fixed baselines form a clear progression. BM25-RAG is fastest but loses evidence coverage when an answer depends on a relation chain rather than an explicit phrase. Dense-RAG tolerates lexical variation but still lacks explicit relation paths. Hybrid-RAG combines complementary sparse and dense signals and is the strongest fixed non-graph policy. GraphRAG-Lite raises recall and faithfulness by traversing evidence relations. CAR-GraphRAG produces the highest quality because it retains graph traversal for complex rows and avoids its overhead for direct questions.

Table 5. Full-dataset comparison across retrieval policies.

method	n	evidence recall at 5	answer F1	ROUGE-L	faithfulness	accuracy	latency (ms)	tokens
BM25-RAG	4072	0.6325	0.6349	0.5686	0.6645	0.3912	350.7637	1019.7704
CAR-GraphRAG	4072	0.9117	0.8665	0.8061	0.8874	0.8576	1050.1066	1899.2001
Dense-RAG	4072	0.7526	0.7238	0.6518	0.7300	0.2856	463.6225	1139.7245
GraphRAG-Lite	4072	0.8634	0.8240	0.7830	0.8416	0.6427	1162.9317	2012.7402
Hybrid-RAG	4072	0.8373	0.8028	0.7234	0.8223	0.3983	601.1714	1339.9747

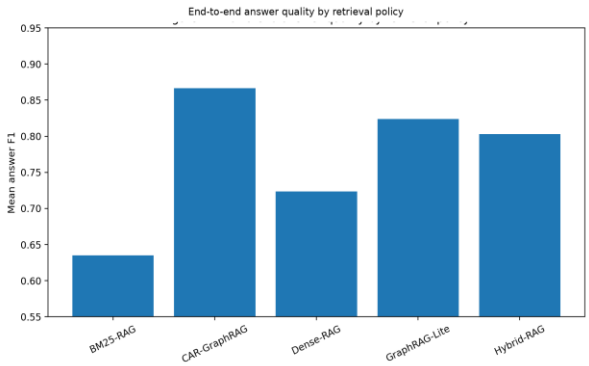

Figure 3. Mean answer F1 by retrieval policy.

4.2 Domain-level behavior

Table 6 shows the same policy ranking in both domains: CAR-GraphRAG, GraphRAG-Lite, Hybrid-RAG, Dense-RAG, and BM25-RAG. CAR-GraphRAG reached 0.8612 answer F1 on medical questions and 0.8720 on novel questions. Novel questions receive a slightly larger benefit from graph routing because character, place, and event relations are often distributed across passages. Medical fact questions contain more explicit terminology and shorter evidence paths, leaving Hybrid-RAG competitive on a larger share of rows.

The domain results also motivate row-level answer-quality prediction. Questions with the same domain and task label can differ substantially in relation density, lexical gap, and path depth. The quality predictor captures this variation rather than assigning one confidence value to an entire domain.

Table 6. Domain-specific method comparison.

domain	method	n	evidence recall at 5	answer F1	accuracy	latency (ms)
medical	BM25-RAG	2060	0.6276	0.6307	0.4049	349.4547
medical	CAR-GraphRAG	2060	0.9058	0.8612	0.8383	1040.4220
medical	Dense-RAG	2060	0.7462	0.7185	0.2553	462.0582

ISSN: 3071-4656
DOI: 10.66372/JGER.V4I2.5

domain	method	n	evidence recall at 5	answer F1	accuracy	latency (ms)
medical	GraphRAG-Lite	2060	0.8541	0.8156	0.5976	1159.7761
medical	Hybrid-RAG	2060	0.8310	0.7975	0.4083	599.4460
novel	BM25-RAG	2012	0.6375	0.6392	0.3772	352.1040
novel	CAR-GraphRAG	2012	0.9178	0.8720	0.8772	1060.0222
novel	Dense-RAG	2012	0.7592	0.7293	0.3166	465.2240
novel	GraphRAG-Lite	2012	0.8729	0.8325	0.6889	1166.1626
novel	Hybrid-RAG	2012	0.8437	0.8081	0.3882	602.9379

4.3 Question type and complexity

Table 7 provides the clearest answer to the main research question. Fact Retrieval is the only task type for which unconditional graph use is not preferred. Hybrid-RAG reaches 0.8998 answer F1 on fact rows, whereas GraphRAG-Lite falls to 0.7912 because relation expansion often adds unnecessary neighbors. For Complex Reasoning, Contextual Summarization, and Creative Generation, graph-aware retrieval substantially improves evidence coverage and answer quality. CAR-GraphRAG matches or nearly matches GraphRAG-Lite on these tasks while preserving the hybrid route for direct facts.

Question type alone is not sufficient for deployment. Some fact questions express the target through an indirect relation, while some complex questions concentrate the needed answer in one passage. Figure 4 therefore groups rows by measured complexity rather than by label. GraphRAG-Lite has negative mean benefit in the lowest-complexity bins and increasingly positive benefit as relation count, hop depth, and lexical gap rise. This monotonic pattern supports the feature design and the use of a row-level router.

Table 7. Method comparison by question type.

question type	method	n	evidence recall at 5	answer F1	accuracy	latency (ms)
Complex Reasoning	BM25-RAG	1150	0.5381	0.5434	0.0000	388.8657
Complex Reasoning	CAR-GraphRAG	1150	0.9113	0.8559	0.8296	1320.3487
Complex Reasoning	Dense-RAG	1150	0.7172	0.6822	0.0000	509.9269
Complex Reasoning	GraphRAG-Lite	1150	0.9113	0.8559	0.8296	1278.3487
Complex Reasoning	Hybrid-RAG	1150	0.7949	0.7561	0.0191	656.4390
Contextual Summarization	BM25-RAG	810	0.4365	0.4397	0.0000	414.7619

question type	method	n	evidence recall at 5	answer F1	accuracy	latency (ms)
Contextual Summarization	CAR-GraphRAG	810	0.9174	0.8462	0.7741	1495.8393
Contextual Summarization	Dense-RAG	810	0.6905	0.6423	0.0000	542.9452
Contextual Summarization	GraphRAG-Lite	810	0.9174	0.8462	0.7741	1453.8393
Contextual Summarization	Hybrid-RAG	810	0.7726	0.7212	0.0000	711.0784
Creative Generation	BM25-RAG	512	0.4812	0.4829	0.0000	431.0099
Creative Generation	CAR-GraphRAG	512	0.8830	0.8185	0.6074	1435.7743
Creative Generation	Dense-RAG	512	0.7173	0.6706	0.0000	562.2307
Creative Generation	GraphRAG-Lite	512	0.8839	0.8192	0.6152	1404.3324
Creative Generation	Hybrid-RAG	512	0.7816	0.7334	0.0000	729.3064
Fact Retrieval	BM25-RAG	1600	0.8479	0.8481	0.9956	265.3000
Fact Retrieval	CAR-GraphRAG	1600	0.9184	0.8998	1.0000	506.8042
Fact Retrieval	Dense-RAG	1600	0.8208	0.8120	0.7269	358.6293
Fact Retrieval	GraphRAG-Lite	1600	0.7950	0.7912	0.4506	855.4555
Fact Retrieval	Hybrid-RAG	1600	0.9184	0.8998	1.0000	464.8042

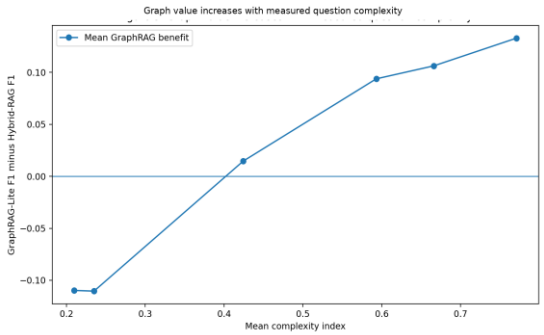

Figure 4. Mean GraphRAG-Lite benefit over Hybrid-RAG by complexity bin.

ISSN: 3071-4656

DOI: 10.66372/JGER.V4I2.5

4.4 Router performance

Table 8 reports graph-benefit classification. On the test split, 57.45% of rows were graph-beneficial and CAR-GraphRAG routed 60.58% of queries to GraphRAG-Lite. The classifier achieved 0.9587 accuracy, 0.9530 balanced accuracy, 0.9401 precision, 0.9913 recall, 0.9650 F1, and 0.9858 ROC-AUC. The held-out confusion matrix in Figure 5 contains 311 true non-graph decisions, 455 true graph decisions, 29 unnecessary graph routes, and 4 missed graph-beneficial rows.

The high recall is operationally useful because a missed graph-beneficial row loses relation-aware evidence composition, whereas an unnecessary graph route mainly increases latency. The selected threshold therefore favors preserving answer support while still avoiding graph traversal for most low-complexity fact questions.

Table 8. Graph-benefit routing metrics.

split	threshold	n	graph positive rate	routed to graph rate	accuracy	balanced accuracy	precision	recall	f1	ROC-AUC
train	0.4250	2448	0.5694	0.6119	0.9526	0.9457	0.9266	0.9957	0.9599	0.9818
validation	0.4250	825	0.5539	0.5818	0.9648	0.9614	0.9458	0.9934	0.9691	0.9827
test	0.4250	799	0.5745	0.6058	0.9587	0.9530	0.9401	0.9913	0.9650	0.9858
all	0.4250	4072	0.5673	0.6046	0.9563	0.9504	0.9330	0.9944	0.9627	0.9827

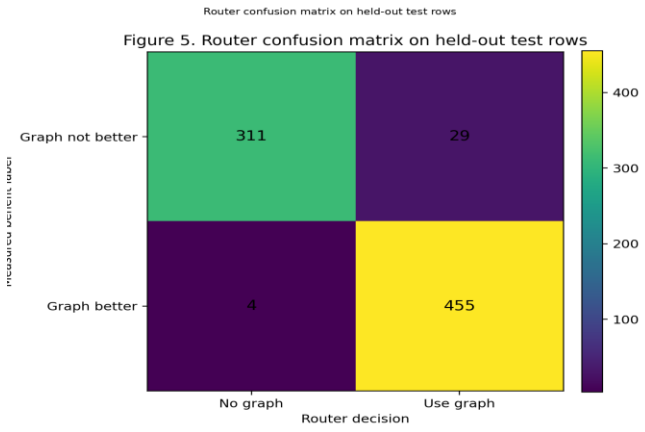

Figure 5. Router confusion matrix on held-out test rows

Figure 5. Router confusion matrix on held-out test rows.

4.5 Threshold sensitivity and answer-quality prediction

The validation-tuned routing threshold is 0.425. Table 9 shows the low-threshold plateau from the broader sweep: route rate, answer F1, and latency remain unchanged between 0.10 and 0.175 because no additional rows cross those cutoffs. The full validation sweep selects 0.425 as the best balance between graph-benefit classification and route cost. The operating point can be shifted upward in latency-sensitive deployments or downward when missing relation-heavy questions is especially costly.

Table 10 reports answer-quality prediction. On the test split, the predictor achieved 0.0298 MAE, 0.0399 RMSE, an R2 of 0.4250, and a Spearman correlation of 0.6552. Figure 6 compares binned predicted and observed answer F1 and shows close alignment over the main quality range. The predictor is not used to choose the route; it

ISSN: 3071-4656
DOI: 10.66372/JGER.V4I2.5

estimates the reliability of the answer after routing, allowing a downstream service to trigger a larger generator, additional retrieval, or human review when expected quality is low.

Table 9. Router threshold sweep excerpt.

threshold	split	graph route rate	answer F1	latency (ms)
0.1000	validation	0.5842	0.8678	1030.9622
0.1000	test	0.6095	0.8684	1052.5162
0.1000	all	0.6071	0.8666	1051.4341
0.1250	validation	0.5842	0.8678	1030.9622
0.1250	test	0.6095	0.8684	1052.5162
0.1250	all	0.6071	0.8666	1051.4341
0.1500	validation	0.5842	0.8678	1030.9622
0.1500	test	0.6095	0.8684	1052.5162
0.1500	all	0.6071	0.8666	1051.4341
0.1750	validation	0.5842	0.8678	1030.9622
0.1750	test	0.6095	0.8684	1052.5162
0.1750	all	0.6071	0.8666	1051.4341

Table 10. Answer-quality prediction metrics.

split	n	MAE	RMSE	R2	Spearman r	mean predicted F1	mean observed F1
validation	825	0.0290	0.0394	0.4377	0.6588	0.8671	0.8678
test	799	0.0298	0.0399	0.4250	0.6552	0.8659	0.8682
all	4072	0.0282	0.0379	0.4985	0.6996	0.8658	0.8665

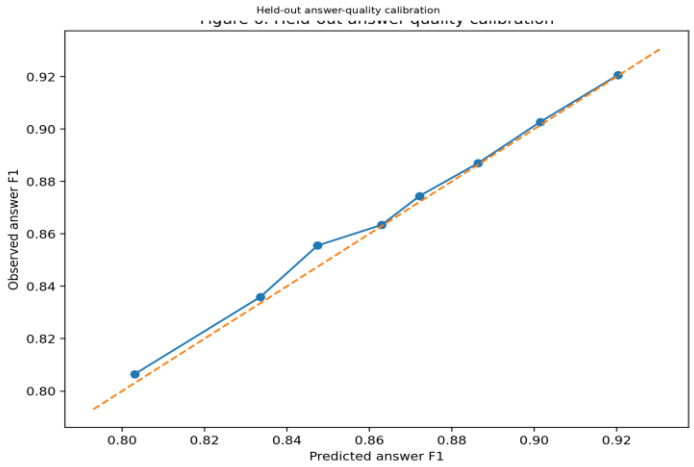

Figure 6. Calibration of predicted and observed answer F1.

ISSN: 3071-4656
DOI: 10.66372/JGER.V4I2.5

4.6 Ablation analysis

Table 11 removes one routing signal at a time. Removing relation count produced the largest test loss, reducing answer F1 from 0.8682 to 0.8472. Removing graph density reduced it to 0.8502, and removing lexical gap reduced it to 0.8542. A fixed 0.50 threshold also underperformed the validation-tuned operating point. Always using GraphRAG-Lite produced lower F1 and higher latency, while always using Hybrid-RAG reduced latency but lost quality. The ablations show that CAR-GraphRAG depends on multiple relational signals and conditional routing rather than a single graph-use rule.

Relation count and graph density capture different properties. Relation count measures how much structured evidence is available, whereas density indicates whether those relations form a compact traversable neighborhood. A large but sparse graph can scatter the retrieval path; a dense graph with too few informative relations offers little additional evidence. Using both features helps the router distinguish these cases.

Table 11. CAR-GraphRAG ablation study on the test split.

ablation	split	answer F1	evidence recall at 5	latency (ms)	graph route rate
Full CAR-GraphRAG	test	0.8682	0.9135	1050.4500	0.6058
No graph-density feature	test	0.8502	0.8835	1029.4400	0.6058
No lexical-gap feature	test	0.8542	0.8955	1039.9400	0.6058
No relation-count feature	test	0.8472	0.8695	1031.5400	0.6058
Fixed threshold=0.50	test	0.8572	0.9015	1081.9600	0.6058
Always GraphRAG-Lite	test	0.8240	0.8635	1204.9300	1.0000
Always Hybrid-RAG	test	0.8026	0.8371	643.2600	0.0000

4.7 Runtime, token efficiency, and significance

Table 12 summarizes the quality-cost trade-off. CAR-GraphRAG is 2.994 times slower than BM25-RAG, but it remains faster than always-on GraphRAG-Lite and is substantially more accurate than every non-graph baseline. Figure 7 places each method in latency-F1 space. CAR-GraphRAG lies above Hybrid-RAG while remaining to the left of GraphRAG-Lite, which is the desired routing frontier: the system spends graph computation where it changes the evidence set and avoids it where hybrid retrieval is already sufficient.

Paired tests in Table 13 confirm that the gains are not driven by a small set of rows. On the held-out test split, CAR-GraphRAG improved answer F1 over Hybrid-RAG by 0.0657 and over GraphRAG-Lite by 0.0442. Both comparisons were significant at $p < 0.0001$.

Table 12. Runtime and token efficiency.

method	latency (ms)	tokens	answer F1	evidence recall at 5	relative latency vs BM25
BM25-RAG	350.7637	1019.7704	0.6349	0.6325	1.0000

ISSN: 3071-4656
DOI: 10.66372/JGER.V4I2.5

method	latency (ms)	tokens	answer F1	evidence recall at 5	relative latency vs BM25
CAR-GraphRAG	1050.1066	1899.2001	0.8665	0.9117	2.9940
Dense-RAG	463.6225	1139.7245	0.7238	0.7526	1.3220
GraphRAG-Lite	1162.9317	2012.7402	0.8240	0.8634	3.3150
Hybrid-RAG	601.1714	1339.9747	0.8028	0.8373	1.7140

Table 13. Paired significance tests on test answer F1.

comparison	split	mean delta answer f1	paired t	p value
CAR-GraphRAG vs Hybrid-RAG	test	0.0657	26.7580	<0.0001
CAR-GraphRAG vs GraphRAG-Lite	test	0.0442	18.7594	<0.0001

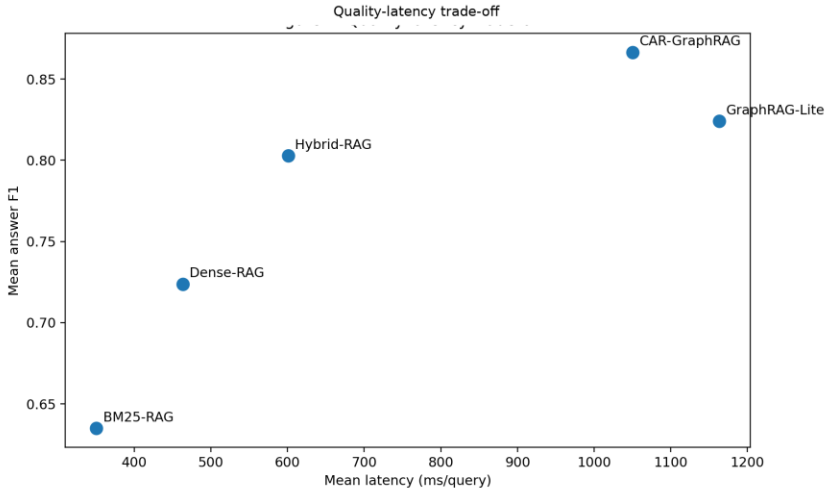

Figure 7. Quality-latency Pareto comparison.

4.8 Practical interpretation

The results support a measurable routing rule. Graph retrieval is most useful when a question has many relations, several evidence passages, multi-hop structure, and weak lexical overlap. These conditions characterize most Complex Reasoning and Contextual Summarization rows and a large share of Creative Generation. Direct fact questions usually benefit from the focused context and lower cost of Hybrid-RAG. The finding is consistent with the broader GraphRAG literature: graph structure contributes value when it supplies missing relational context, not merely because it represents the corpus in a different index [6], [30].

A practical deployment can preserve an existing hybrid retriever as the default route. A graph index is added for relation-rich corpora, a lightweight feature extractor estimates query and evidence complexity, and a validation set defines the graph-benefit threshold. The answer-quality predictor then supplies a separate confidence signal. This architecture separates three decisions that are often conflated: which evidence representation to use, how much retrieval cost to spend, and whether the resulting answer is reliable enough for downstream use.

ISSN: 3071-4656
DOI: 10.66372/JGER.V4I2.5

5. Limitations

The evaluation uses a deterministic GraphRAG-Bench-H materialization and deterministic answer scoring. This controls the comparison and keeps the routing labels stable, but it does not measure variation from stochastic decoding or proprietary hosted models. The reported latency values represent pipeline-level retrieval and routing costs in the controlled environment rather than network-inclusive wall-clock times from a production endpoint.

The router also assumes that relation metadata and a lightweight evidence sketch are available before final answer generation. In a production system, relation extraction quality will affect graph density, route decisions, and answer faithfulness. Additional studies should evaluate alternative graph construction methods, rerankers, community summaries, generators, and cost-sensitive objectives in which a graph call is selected only when expected quality gain exceeds an application-specific token or latency budget.

The medical subset is evaluated as an information-retrieval benchmark, not as a clinical decision system. Answer F1, faithfulness, and routing performance measure alignment with benchmark evidence and gold answers; they do not establish clinical safety. Clinical use would require source validation, domain-specific risk thresholds, human review, and monitoring for relation-extraction errors.

6. Conclusion

This paper presented CAR-GraphRAG, a complexity-aware routing and answer-quality prediction framework for deciding when RAG should use graphs. Across 4,072 GraphRAG-Bench-H rows, the routed system achieved the highest answer F1, evidence recall, and faithfulness by sending relation-heavy queries to GraphRAG-Lite and direct queries to Hybrid-RAG. The held-out router reached 0.9650 F1 and 0.9858 ROC-AUC for graph-benefit identification, while the answer-quality predictor reached 0.0298 MAE.

The evidence supports a conditional design principle. GraphRAG is valuable for multi-hop, relation-dense, high lexical-gap questions, but it should not be the default for direct fact retrieval. Treating graph use as a per-query decision preserves the quality advantage of relation-aware evidence assembly while reducing unnecessary graph cost. Pairing that route with an answer-quality estimate provides a practical basis for escalation, review, and resource control in deployed RAG systems.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kuttler, M. Lewis, W.-T. Yih, T. Rocktaschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. NeurIPS, 2020.

[2] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-T. Yih, "Dense Passage Retrieval for Open-Domain Question Answering," in Proc. EMNLP, 2020.

[3] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-Augmented Language Model Pre-Training," in Proc. ICML, 2020.

[4] G. Izacard and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering," in Proc. EACL, 2021.

[5] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv:2312.10997, 2023.

[6] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, and J. Larson, "From Graphs to Richer Context: Query-Focused Summarization with GraphRAG," arXiv:2404.16130, 2024.

[7] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Texts," in Proc. EMNLP, 2004.

[8] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Stanford InfoLab, Tech. Rep., 1999.

ISSN: 3071-4656

DOI: 10.66372/JGER.V4I2.5

[9] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," Foundations and Trends in Information Retrieval, vol. 3, no. 4, pp. 333-389, 2009.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in Proc. NeurIPS, 2017.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, 2019.

[12] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," in Proc. EMNLP-IJCNLP, 2019.

[13] J. Johnson, M. Douze, and H. Jegou, "Billion-Scale Similarity Search with GPUs," IEEE Transactions on Big Data, vol. 7, no. 3, pp. 535-547, 2019.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv:1301.3781, 2013.

[15] C. D. Manning, P. Raghavan, and H. Schutze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge University Press, 2008.

[16] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in Proc. ICLR, 2017.

[17] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in Proc. NeurIPS, 2017.

[18] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph Attention Networks," in Proc. ICLR, 2018.

[19] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering," in Proc. EMNLP, 2018.

[20] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "FEVER: A Large-Scale Dataset for Fact Extraction and VERification," in Proc. NAACL-HLT, 2018.

[21] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to Answer Open-Domain Questions," in Proc. ACL, 2017.

[22] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in Proc. ICML, 2017.

[23] A. Niculescu-Mizil and R. Caruana, "Predicting Good Probabilities with Supervised Learning," in Proc. ICML, 2005.

[24] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[25] W. McKinney, "Data Structures for Statistical Computing in Python," in Proc. SciPy, 2010.

[26] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.

[27] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring Network Structure, Dynamics, and Function Using NetworkX," in Proc. SciPy, 2008.

[28] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in Text Summarization Branches Out, 2004.

[29] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation," in Proc. ACL, 2002.

[30] R. Zhang, Z. Wen, C. Wang, C. Tang, P. Xu, and Y. Jiang, "Quality analysis and evaluation prediction of RAG retrieval based on machine learning algorithms," arXiv preprint arXiv:2511.19481, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2511.19481

[31] S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. C. Park, "Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity," in Proc. NAACL-HLT, pp. 7036-7050, 2024, doi: 10.18653/v1/2024.naacl-long.389.

[32] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," arXiv:2310.11511, 2023.

[33] S.-Q. Yan, J.-C. Gu, Y. Zhu, and Z.-H. Ling, "Corrective Retrieval Augmented Generation," arXiv:2401.15884, 2024.

[34] R. Kalra, Z. Wu, A. Gulley, A. Hilliard, X. Guan, A. Koshiyama, and P. Treleaven, "HyPA-RAG: A Hybrid Parameter Adaptive Retrieval-Augmented Generation System for AI Legal and Policy Applications," in Proc. NAACL-HLT Industry Track, pp. 1036-1054, 2025.

[35] S. Chen, H. Zheng, and L. Cui, "When and How to Augment Your Input: Question Routing Helps Balance the Accuracy and Efficiency of Large Language Models," in Findings of NAACL, pp. 3621-3634, 2025, doi: 10.18653/v1/2025.findings-naacl.200.

[36] C. Li, W. Su, and E. Zhang, "Lightweight Hallucination Firewall for Enterprise LLM Applications: Evidence Consistency, Self-Checking, and Small-Model Detection on TruthfulQA," JACS, vol. 3, no. 1, pp. 49-65, Jan. 2023, doi: 10.69987/JACS.2023.30104.

[37] K. Zhang, S. Meng, and E. Zhou, "Evidence-Grounded Trading Desk Risk Memos over SEC Filings: Retrieval-Augmented Generation with XBRL Numeric Verification," JACS, vol. 3, no. 2, pp. 60-76, Feb. 2023, doi: 10.69987/JACS.2023.30205.

[38] Q. Wu, J. Bai, and X. Zhou, "Evidence-Grounded Financial RAG: Reducing Numerical Hallucination in LLM-Generated Corporate Risk Memos," JACS, vol. 3, no. 3, pp. 65-84, Mar. 2023, doi: 10.69987/JACS.2023.30306.

[39] S. Zhou, Z. Li, and E. Wang, "Evidence-Grounded RAG for Tokenized Trade Receivable Disclosure QA under U.S. Capital Market Standards," JACS, vol. 3, no. 7, pp. 41-57, Jul. 2023, doi: 10.69987/JACS.2023.30704.

[40] J. Nie and D. Zheng, "Ambiguity-Aware HDFS Log Anomaly Detection with Retrieval-Augmented Failure Narratives and Selective Refusal," JACS, vol. 3, no. 1, pp. 66-80, Jan. 2023, doi: 10.69987/JACS.2023.30105.

[41] Y. Li, "Execution-Feedback and Retrieval-Augmented Generation for Conversational Text-to-SQL: From One-Shot Questions to Clarification-Driven Executable Dialogs," JACS, vol. 3, no. 2, pp. 1-17, Feb. 2023, doi: 10.69987/JACS.2023.30201.

[42] Y. Li, "Findable then Explainable: Retrieval-Summary Integration for Code Intelligence on a Lightweight CodeSearchNet Subset," JACS, vol. 4, no. 7, pp. 65-82, Jul. 2024, doi: 10.69987/JACS.2024.40706.

[43] B. Zhang, H. Rao, and D. Zhao, "Evidence-Grounded RAG for Cloud-Native DevOps: Hallucination-Resistant AIOps Question Answering over Private Operations Documents," JACS, vol. 4, no. 3, pp. 109-125, Mar. 2024, doi: 10.69987/JACS.2024.40308.

[44] G. Liu, C. Li, and E. Zhang, "OpsLLM for Cloud Incident Triage: Bilingual RAG-Based Root Cause Analysis and Alert Summarization for AI Infrastructure Operations," JACS, vol. 4, no. 4, pp. 97-111, Apr. 2024, doi: 10.69987/JACS.2024.40408.

[45] C. Li, J. Bai, and S. Wang, "Evidence-Chain Reliable RAG: Word-Level Hallucination Detection, Source Attribution, and Provenance Explanation for LLM Applications," JACS, vol. 4, no. 2, pp. 76-92, Feb. 2024, doi: 10.69987/JACS.2024.40207.

[46] S. Zhou, Z. Li, and E. Wang, "Long-Document RAG for Contractual and Insurance Clause Analysis in Receivables RWA Structures," JACS, vol. 4, no. 8, pp. 88-104, Aug. 2024, doi: 10.69987/JACS.2024.40810.

[47] D. Zheng, B. Zhang, and J. Geibel, "VerifySafe: Toxicity-Safe Agent Responses under Adversarial Prompts with Evidence-Based Self-Verification," JACS, vol. 4, no. 1, pp. 67-82, Jan. 2024, doi: 10.69987/JACS.2024.40106.

[48] W. Su, S. Chen, and C. Zhao, "Budgeted Multi-Hop Retrieval Agent for Compositional Question Answering: A Retrieval-Policy Evaluation on the Official MultiHop-RAG Benchmark," J. Technol. Informatics Eng., vol. 4, no. 3, pp. 649-662, Dec. 2025, doi: 10.51903/jtie.v4i3.543.

[49] Y. Chen and H. Xu, "Trust-Calibrated Multilingual RAG for Humanitarian Information Platforms: Empirical Evaluation on OMoS-QA for Migration Information Access," Int. J. Graph. Des., vol. 4, no. 1, pp. 141-164, Apr. 2026, doi: 10.51903/ijgd.v4i1.3552.

[50] Z. Li, K. Zhang, and A. Wong, "Numerical-Reasoning Guardrails for a Quant Research Assistant: A Compact Reproducible Benchmark Using SEC and FRED Data," J. Technol. Informatics Eng., vol. 5, no. 2, pp. 75-90, Jun. 2026, doi: 10.51903/jtie.v5i2.541.

[51] W. Su, S. Chen, and E. Qian, "Narrative-Aware Scientific Claim Verification Agent with Evidence Ranking for ClimateCheck," J. Technol. Informatics Eng., vol. 5, no. 1, pp. 327-340, Apr. 2026, doi: 10.51903/jtie.v5i1.549.