Print ISSN: 3104-932X
Online ISSN: 3104-9338

Research on Factual Hallucination Detection and Mitigation in Large Language Models Based on Retrieval-Augmented Generation

Zehua Zhou

Reedy High School, 3003 Stonebrook Parkway, Frisco, TX, USA

Received: October 5, 2026

Revised: October 9, 2026

Accepted: October 19, 2026

Published online: October 22, 2026

To appear in: *International Journal of Advanced AI Applications*, Vol. 2, No. 11 (November 2026)

Corresponding Author: Zehua Zhou (lai@cis.ac.cn)

Abstract. Large language models (LLMs) can generate fluent, persuasive responses that nevertheless contain unsupported, outdated, or contradictory factual claims. Retrieval-augmented generation (RAG) reduces dependence on model parameters by supplying external evidence, but retrieval alone does not guarantee factuality: the evidence may be incomplete, irrelevant, conflicting, poorly ranked, or ignored during generation. This paper develops a transparent, evidence-grounded framework for detecting and mitigating factual hallucinations in RAG-based LLM applications. The framework treats factuality as a claim-level decision problem rather than a single response-level label. It combines query decomposition, hybrid retrieval, evidence quality assessment, atomic-claim extraction, claim-to-evidence verification, uncertainty calibration, source-linked generation, and risk-sensitive response control. A proposed evaluation protocol connects claim precision, evidence coverage, retrieval recall, citation correctness, abstention quality, latency, and human review. Existing methods—including SelfCheckGPT, FActScore, RAGAS, and Chain-of-Verification—are mapped to complementary parts of the pipeline rather than treated as interchangeable solutions. The paper is a framework and evaluation-design study; it does not report new model experiments or claim measured performance improvements. Its main contribution is a reproducible blueprint for testing whether an integrated system can reduce unsupported factual claims while preserving answer usefulness, traceability, and computational efficiency.

Keywords: *Large Language Models; Factual Hallucination; Retrieval-Augmented Generation; Claim Verification; Uncertainty Estimation; Factuality Evaluation*

1. Introduction

1.1. Research Background

Large language models have become general-purpose interfaces for question answering, summarization, analytical writing, coding assistance, and information support. Their fluency can make generated statements appear authoritative even when a claim is inaccurate or not supported by available evidence. Research on hallucination in natural language generation distinguishes several failure modes, including content that conflicts with source material and content that is not supported by the source or reference context [1]. Recent surveys further emphasize that factuality is multidimensional: the same response may contain correct statements, unsupported details, and errors of attribution at once [2].

The issue is especially consequential in knowledge-intensive applications. A system may be asked to explain a policy, summarize a scientific article, compare product specifications, or answer questions about a changing body of documentation. In these settings, factual correctness depends not only on language-model parameters, but also on access to authoritative, relevant, and current information. Benchmarks such as TruthfulQA show why fluent generation cannot be equated with truthfulness: a model may reproduce misconceptions learned from training data rather than provide a reliable answer [3].

Retrieval-augmented generation (RAG) offers a practical path toward grounding answers in external information. Instead of relying entirely on parametric memory, a RAG system retrieves documents or passages and conditions generation on that material [4]. This design can provide traceable evidence and make updates possible without retraining the underlying model. Nevertheless, retrieval changes the failure surface rather than eliminating it. A search component may miss the best source, surface a stale page, rank a distractor highly, or return passages that agree only superficially with the query. A generator may then overstate what the retrieved text establishes, combine facts from different entities, or add unsupported names, dates, numbers, and causal explanations.

Accordingly, the central problem is not simply to retrieve more text. It is to determine which generated claims are supported by which pieces of evidence, detect when the evidence is insufficient or contradictory, and select a response action that reflects the strength of that evidence. This paper proposes a layered framework in which detection and mitigation are coupled: a factuality signal is useful only when it can trigger an appropriate action, such as targeted re-retrieval, claim revision, citation insertion, uncertainty qualification, or abstention.

1.2. Problem Statement and Research Questions

Existing approaches address different fragments of the problem. Sampling-based consistency methods can detect disagreement between independently generated answers without requiring an external database [5]. Atomic-fact evaluation can identify which propositions in a long answer have support from a knowledge source [6]. RAG evaluation frameworks separate retrieval quality from generation faithfulness [7], while verification-oriented generation methods instruct a model to draft, formulate verification questions, answer those questions, and revise its response [8]. These methods offer useful components, but practical systems require a coordinated policy for combining their signals under different evidence conditions and risk levels.

This paper addresses three research questions:

RQ1. How can factual hallucinations in RAG-based LLM responses be identified at the atomic-claim level using retrieved evidence and explicit support labels?

RQ2. How can detection outcomes be converted into mitigation actions that account for source quality, evidence conflict, uncertainty, and the consequences of error?

RQ3. How should an integrated approach be evaluated so that factuality gains are measured alongside retrieval quality, citation validity, answer utility, abstention behavior, and inference cost?

1.3. Contributions and Scope

The paper makes three contributions. First, it defines a practical taxonomy of factual failure modes across retrieval, evidence interpretation, claim generation, and source attribution. Second, it proposes an integrated RAG control pipeline that connects retrieval diagnostics, claim decomposition, evidence verification, and risk-sensitive mitigation. Third, it specifies an evaluation protocol using established factuality and RAG assessment ideas, with explicit baselines, metrics, ablations, and reproducibility requirements. The analysis is grounded in published work [1-10], but it does not introduce an evaluated implementation, new dataset, or newly measured performance result. All experimental steps below are proposed for future empirical validation, not reported as completed tests.

2. Theoretical Background and Related Work

2.1. What Constitutes a Factual Hallucination?

For this study, a factual hallucination is a proposition in a model-generated response that is

false, unsupported by the evidence available to the system, or presented with greater certainty than the evidence justifies. This operational definition intentionally separates truth in the world from support in the system context. A statement may happen to be true but still be unsupported by retrieved evidence; conversely, a response may faithfully summarize a source that itself is outdated or erroneous. Evaluation should record these situations separately rather than merge them into one binary label.

Four categories are particularly relevant to RAG. (1) Retrieval omission occurs when the corpus contains suitable evidence but the retriever fails to return it. (2) Evidence misinterpretation occurs when the correct source is present but the generator draws an invalid inference, confuses scope, or misreads a condition. (3) Unsupported elaboration occurs when a response adds plausible details that are absent from the retrieved evidence. (4) Attribution failure occurs when a citation does not support the adjacent claim, refers to the wrong entity, or points to a source that cannot be independently checked. Contradiction and temporal failure are cross-cutting categories: different sources may disagree, or an otherwise relevant source may no longer describe the current state of affairs.

The distinction between intrinsic and extrinsic hallucination in natural language generation is useful here [1]. Intrinsic failures conflict with the supplied context; extrinsic failures introduce material that cannot be verified from that context. A RAG system may exhibit both. A response that reverses a retrieved date is intrinsically inconsistent with the evidence, while one that appends an unmentioned statistic is an unsupported elaboration even if it sounds plausible. Labeling these cases separately helps determine whether mitigation should focus on better retrieval, stricter evidence use, or external verification.

2.2. Retrieval-Augmented Generation

A conventional RAG pipeline includes document preparation, indexing, query processing, retrieval, context construction, and answer generation. Documents are divided into passages or chunks, transformed into searchable representations, and stored in lexical or vector indexes. At inference time, the question is processed and related passages are retrieved; the selected context is then included in the prompt used by the generator [4]. A hybrid retriever can combine lexical relevance with semantic similarity, while a reranker can reorder candidates using a deeper query–passage relevance model.

The architecture offers two fundamental advantages. It gives a model access to non-parametric knowledge that can be inspected and refreshed, and it creates the possibility of attaching a source to a generated claim. However, retrieval correctness and generation

faithfulness are distinct dimensions. The presence of a relevant passage does not guarantee that the generator will use it correctly. Conversely, a faithful generator cannot produce a well-supported answer if the necessary evidence was never retrieved. Evaluation frameworks such as RAGAS explicitly separate properties of the retrieved context and the generated answer, illustrating the need for multi-dimensional assessment [7].

Longer context is not automatically better. Liu et al. report that language models can use relevant information less reliably when its position changes within a long context, with degraded performance when evidence appears in the middle of the input [9]. Thus, simply increasing the number of retrieved passages may increase token cost and distractor exposure without proportionate factual benefit. A factuality-oriented RAG system should optimize the relevance, diversity, freshness, and sufficiency of evidence—not only the total amount of context retrieved.

2.3. Detection Approaches

2.3.1. Consistency-Based Detection

SelfCheckGPT uses the intuition that repeated samples from a model are more likely to agree on familiar, grounded information than on unstable fabricated details. By comparing sampled passages or sentence-level claims, it estimates the likelihood that content is non-factual without requiring a dedicated external database [5]. This approach is useful where model access is limited or where retrieval cannot be assumed. Its limitations are important: consistent outputs can still repeat the same misconception, and disagreement can arise from stylistic variation or ambiguity rather than a factual error. Consistency should therefore be interpreted as a signal, not a proof of truth.

2.3.2. Atomic-Claim Evaluation

FActScore addresses the mixed quality of long-form answers by decomposing a generation into atomic facts and measuring the proportion supported by a reliable knowledge source [6]. Claim-level analysis is suitable for RAG because an answer may contain both supported and unsupported propositions, making a single answer-level label too coarse. Atomic claims also allow targeted repair: the system can preserve verified statements while rewriting or removing only the unsupported portion. The quality of decomposition and the definition of an acceptable evidence match remain critical, especially for temporal, comparative, numerical, and causal claims.

2.3.3. Claim Verification and Entailment

Fact verification research treats a claim as an object to be checked against evidence, a task exemplified by the FEVER benchmark [10]. In an RAG setting, a claim verifier can classify the relationship between each atomic claim and retrieved passages as supported, contradicted, or not verifiable. The third category must not be conflated with contradiction: absence of evidence may indicate retrieval failure, incomplete documentation, or a genuinely unknowable answer. A verifier should also preserve the source passage and the reasoning trace used for its label so that errors can be audited.

2.4. Mitigation Approaches

2.4.1. Evidence-Grounded Generation

The first mitigation is to constrain generation to information supported by retrieved sources. The prompt can request source-linked claims, prohibit unsupported numbers or names, distinguish direct evidence from inference, and instruct the model to state that evidence is insufficient when necessary. Such instructions improve operational clarity but should not be relied on alone: a model can still overlook instructions or make unsupported inferences. Evidence-grounded generation works best when paired with post-generation verification and an action policy.

2.4.2. Retrieval Repair and Query Refinement

When a claim is unsupported, the system should diagnose whether the required fact was absent from the candidate set, buried in a long context, or present but misinterpreted. If retrieval recall is the likely issue, it can reformulate the query, add entity aliases, use a different retrieval channel, broaden the search, or retrieve from a more authoritative source. Where available, metadata filters should constrain retrieval by publication date, jurisdiction, document version, or source type. Retrieval repair should be bounded by a retry budget so that repeated searching does not cause unbounded latency or amplify low-quality evidence.

2.4.3. Verification Before Finalization

Chain-of-Verification (CoVe) proposes a multi-step procedure in which a model drafts an answer, develops verification questions, answers those questions independently, and then produces a revised response [8]. This pattern can be adapted to RAG by creating one or more check questions for every material claim and resolving them against retrieved passages before finalization. Independent verification is especially useful for dates, quantities, named entities, and multi-part answers, though the verifier itself can make mistakes. Critical claims should

therefore use evidence from sources separate from the initial draft when feasible.

2.4.4. Calibrated Abstention

A dependable system must sometimes decline to provide a direct answer. Abstention may be appropriate when the evidence is missing, sources conflict without a principled way to resolve them, source recency is unclear, or a high-impact claim remains below a chosen support threshold. A useful abstention is informative rather than opaque: it briefly states which part cannot be verified, identifies the evidence gap, and indicates what additional source or clarification would be needed. Because excessive abstention makes a system unhelpful, the evaluation must measure both erroneous answers and avoidable refusals.

2.5. Research Gap

Prior work provides strong building blocks for factuality measurement, retrieval evaluation, and multi-step verification, but several integration challenges remain. First, the retrieval layer and the answer-verification layer are often measured separately, making it difficult to attribute errors to the stage that caused them. Second, a single support score does not express the impact of an error: an unsupported phrase in a casual explanation is not equivalent to an invented legal requirement or medical contraindication. Third, source quality, recency, contradiction, and citation alignment are often treated as secondary diagnostics even though they directly affect whether evidence warrants a claim. Fourth, many evaluations emphasize factuality while leaving latency, token cost, and abstention utility under-specified. The framework below connects these concerns within one auditable workflow.

3. Proposed Evidence-Grounded Framework

3.1. Design Principles

The proposed framework is designed around five principles. Evidence first: important factual statements should be traceable to a source rather than justified only by model confidence. Granularity: checks should operate on atomic claims so that partial errors can be repaired without discarding an entire answer. Separation of roles: retrieval, generation, verification, and response policy should have explicit interfaces. Risk sensitivity: the threshold for answering should reflect the potential consequence of error. Auditability: the system should retain query, retrieved passages, source metadata, claim labels, decision reasons, and final citations in a reproducible record.

These principles do not imply that every sentence requires a costly multi-model verification

pass. Low-risk tasks may use lightweight checks, while high-impact or weakly supported answers receive stronger verification. The system can allocate compute adaptively based on evidence insufficiency, model disagreement, entity ambiguity, temporal sensitivity, and the estimated impact of the claim. This design treats factuality as a controlled workflow rather than a single classifier added after generation.

3.2. End-to-End Pipeline

The pipeline begins with a user question and ends with an answer whose factual claims have been checked, qualified, or withheld. The stages are: (1) question analysis; (2) evidence retrieval; (3) evidence filtering and ranking; (4) grounded draft generation; (5) atomic-claim extraction; (6) claim-to-evidence verification; (7) risk-sensitive mitigation; and (8) citation and response audit. The steps may run iteratively: verification findings can trigger another retrieval round or require the generator to rewrite a specific claim. The system should store stage outputs so that a failed response can be diagnosed after deployment. The stages are summarized in Table 1.

Table 1. Proposed RAG factuality-control pipeline

Stage	Primary output / control
Question analysis	Extract intent, entities, time frame, requested evidence type, and risk level.
Retrieval	Combine lexical and semantic search; preserve document IDs, timestamps, and passage boundaries.
Evidence filtering	Remove duplicates, down-rank stale or low-authority sources, and detect conflicts.
Grounded draft	Generate only source-linked claims; mark inference and missing information explicitly.
Claim verification	Split the draft into atomic claims and label each supported, contradicted, or unresolved.
Mitigation + audit	Re-retrieve, rewrite, qualify, cite, escalate or abstain; validate citations and record decisions/costs.

3.3. Query Analysis and Evidence Retrieval

Query analysis should identify the target entities, requested relation, temporal scope, and the level of specificity required. For example, a question about a school policy should include the institution and the relevant school year; a question about a regulation should include jurisdiction and effective date. When a query is ambiguous, the system may ask for clarification rather than retrieving a large set of loosely related documents. This step reduces entity substitution and accidental blending of similarly named people, organizations, or policies.

Retrieval should use a documented collection of sources appropriate to the task. A hybrid

search strategy combines keyword match—useful for exact names, identifiers, and phrases—with semantic search for paraphrased or conceptually related passages. A reranking stage can prioritize passages that directly address the requested relation. Source metadata should include title, publisher, publication or update date, document version, URL or stable identifier, and where possible an authority label. The system should not treat ranking score alone as proof of truth; ranking estimates relevance, not factual reliability.

Evidence sufficiency should be checked before generation. A retrieval module can ask whether the returned passages cover the question's required slots: who, what, when, where, quantity, condition, and exception. A query with multiple parts may require separate evidence for each part. If only some slots are supported, the generator should answer the supported portion and explicitly qualify the remainder, or trigger another search. Context compression must preserve the wording and qualifiers that determine a claim's meaning; dropping a "may," "except," or effective date can invert the source's implication.

3.4. Atomic-Claim Extraction and Verification

After a draft is generated, a claim extractor divides it into minimal propositions that can be independently checked. "The policy was introduced in 2022 and applies to all students" should become at least two claims: one about the introduction date and one about scope. Compound claims are difficult to verify as a unit because one component may be supported while another is not. The extractor should preserve relationships between claims and the source spans from which the draft was generated.

For each claim, the verifier retrieves or selects the most relevant evidence and assigns one of three primary labels: supported, contradicted, or unresolved. "Supported" means that the cited evidence entails or directly substantiates the claim within the source's stated scope. "Contradicted" means the evidence states an incompatible fact or condition. "Unresolved" means the current evidence neither establishes nor refutes the claim. This three-way decision avoids the false equivalence between absence of retrieved support and demonstrated falsity.

The verifier should evaluate entailment at a semantic level rather than rely solely on lexical overlap. Dates, units, negation, quantifiers, modal verbs, conditionals, comparisons, and entity identity deserve explicit checks. A statement that "some participants improved" does not support "all participants improved"; a source describing a proposal does not establish that it was enacted; and a historical statement may not support a present-tense claim. If the verifier relies on another language model, its output should include a short rationale tied to exact

evidence spans and should be calibrated against human annotations before being used as a hard gate.

A separate citation-alignment check then verifies that every citation points to a resolvable source and supports the nearby proposition. Source existence and claim support are different checks: a valid URL does not imply that the cited page substantiates the statement. The final answer should maintain a mapping from claim IDs to source IDs and evidence spans so that reviewers can trace exactly why each claim was retained, rewritten, or removed.

3.5. Mitigation Policy and Risk Tiers

The mitigation controller combines support labels, source quality, model uncertainty, and estimated impact. Its actions should be deterministic enough to audit while allowing task-specific configuration. A single low-risk unresolved detail can often be removed or qualified. A central claim with weak support should trigger targeted re-retrieval. A contradiction between high-quality sources may require a dated comparison or an explicit presentation of the disagreement. If a high-impact claim remains unresolved, the system should abstain from giving a definitive answer and explain the evidence limitation.

Action A — retain and cite. Use when one or more appropriate sources directly support the claim and no material contradiction is detected.

Action B — qualify. Use when evidence supports a narrower statement than the draft, or when the source is limited by date, population, jurisdiction, or uncertainty.

Action C — repair and verify. Use targeted retrieval or query decomposition when key evidence is missing, then regenerate only the affected claims and repeat verification.

Action D — disclose conflict. Use when credible sources disagree. State the relevant dates or source contexts instead of silently selecting a convenient answer.

Action E — abstain or escalate. Use when evidence is insufficient for a consequential claim, critical citations cannot be validated, or automated checks remain unstable after a bounded retry budget.

Risk tiers should be defined by deployment context rather than by the model alone. A low-stakes brainstorming task may tolerate a looser threshold if uncertainty is clearly signaled. Legal, medical, financial, educational-policy, and safety-related responses require stricter source criteria, more explicit temporal checks, and human oversight when the evidence is ambiguous. The framework is not a substitute for domain-specific governance; it is a technical mechanism that helps surface unsupported claims before the final response is delivered.

3.6. Operational Algorithm

The control logic can be summarized as follows. First, analyze the query and assign an initial risk tier. Second, retrieve a bounded candidate set using hybrid search and rerank the results. Third, assess evidence coverage and source metadata; if coverage is insufficient, refine the query within a fixed retry budget. Fourth, generate a draft that associates factual statements with source identifiers. Fifth, split the draft into atomic claims and verify each claim against the evidence. Sixth, route unsupported or contradicted claims to targeted retrieval, rewriting, qualification, conflict disclosure, abstention, or human review according to the risk tier. Seventh, validate citations and produce an audit record. This sequence should terminate when all material claims pass the configured support threshold or the retry budget is exhausted.

An implementation should distinguish hard constraints from model-generated recommendations. For example, a citation URL that fails to resolve can be treated as a hard failure, while a semantic entailment score may be a probabilistic signal subject to calibration. The policy layer must define minimum evidence requirements and prevent the generator from overriding them merely by restating a claim more confidently. Logs should include model and index versions, retrieval parameters, timestamps, support labels, retry count, and final action to allow reproducibility and incident review.

4. Proposed Evaluation Methodology

4.1. Study Design and Hypotheses

A controlled evaluation is proposed to test the framework against component baselines. The unit of analysis should be a question–answer pair with gold evidence and claim-level annotations where available. The experiment should compare the same underlying generator across different retrieval and verification configurations so that observed differences can be attributed to the pipeline rather than to unrelated model changes. This is a proposed protocol; the present manuscript does not claim that the experiment has been run.

H1. An integrated retrieval-plus-claim-verification pipeline is expected to reduce the proportion of unsupported factual claims relative to generation without post-generation verification.

H2. Evidence-aware mitigation is expected to improve citation support and claim-level factual precision compared with a RAG baseline that provides retrieved passages but does not validate the generated claims.

H3. Risk-sensitive routing is expected to produce a better trade-off between factuality and answer coverage than a single global abstention threshold, although the benefit may vary by task and domain.

H4. Increasing retrieval context size without evidence filtering is not expected to improve factuality monotonically, because long or noisy contexts can introduce distractors and make relevant evidence harder to use [9].

4.2. Data and Task Construction

The evaluation should combine established public benchmarks with a curated RAG test set. TruthfulQA can test whether responses reproduce common misconceptions rather than provide truthful answers [3]. FEVER offers a fact-verification setting with claims and supporting or refuting evidence [10]. FActScore provides a claim-level factuality framework for longer-form generations [6]. RAGAS metrics can be used to inspect dimensions such as retrieval relevance and faithfulness in a RAG pipeline [7]. These resources measure different aspects of behavior and should not be combined into a single score without documenting their definitions and limitations.

A domain-specific RAG set should contain questions whose answers can be determined from a frozen collection of dated, authoritative documents. It should include direct fact lookup, multi-hop questions, temporal questions, questions with conflicting sources, unanswerable questions, and queries containing ambiguous entities. For each question, annotators should record the gold answer or acceptable answer set, gold supporting passages, source dates, required claim slots, and whether the available corpus is sufficient to answer. The corpus and test questions must be split to prevent leakage between tuning and evaluation, and benchmark licensing and terms of use should be followed.

The source collection should be snapshot-based. Each document should have a stable ID, version, capture date, and provenance record. When the task concerns facts that change over time, the gold annotation must define a reference date; otherwise, an updated source could make a correct historical answer appear incorrect or vice versa. Annotation guidelines should specify how to treat partial support, inferential claims, numerical rounding, and multiple acceptable citations. A subset should be independently double-annotated and adjudicated to estimate agreement and reveal ambiguous cases.

4.3. Baselines and Ablation Conditions

At minimum, the proposed evaluation should compare the following systems under the same

questions and generator settings: (1) parametric-only generation without retrieval; (2) standard top-k RAG without post-generation verification; (3) RAG with source citations requested in the prompt; (4) RAG with claim decomposition and a support verifier; (5) RAG with verification and mitigation; and (6) the full risk-sensitive pipeline with targeted re-retrieval and calibrated abstention. Where applicable, SelfCheckGPT-style sampling can be included as a complementary black-box detection baseline [5], while CoVe-style verification can be tested as a multi-step revision baseline [8].

Ablation tests should remove one component at a time: hybrid retrieval, reranking, source-quality filtering, claim decomposition, contradiction detection, citation alignment, re-retrieval, and risk-sensitive routing. Retrieval depth should be varied to test whether additional passages improve coverage or simply increase distractor exposure. A context-order ablation should move relevant evidence from the beginning to the middle and end of the prompt, motivated by evidence that long-context performance can depend on information position [9]. The study should keep the underlying language model, decoding parameters, and source corpus constant where possible.

4.4. Evaluation Metrics

No single metric captures the whole problem. Evaluation should report claim-level factual precision, evidence coverage, contradiction detection, retrieval quality, citation validity, answer utility, abstention quality, and computational cost separately. Automated metrics can reduce evaluation cost, but a manually annotated subset remains necessary because a verifier may share the same biases as the generator and because semantic entailment is difficult to measure perfectly.

Claim factual precision. The number of generated factual claims labeled supported divided by all factual claims judged; also report the unsupported and contradicted fractions separately.

Evidence coverage / recall. The proportion of gold supporting evidence retrieved within the candidate set or final context, reported at defined k values and with duplicate passages handled consistently.

Citation precision. The proportion of citations that resolve and substantively support their linked claims. Citation existence should be reported separately from semantic support.

Verification performance. Precision, recall, F1, and confusion matrix for the supported / contradicted / unresolved classifier, including results by claim type and source quality.

Answer utility. Human or benchmark assessment of relevance, completeness, coherence, and whether the response answers the question without unsupported expansion.

Abstention quality. Coverage at a fixed error rate, risk–coverage curves, and the rate of unnecessary abstention on answerable questions.

Efficiency. End-to-end latency, number of retrieval calls, model calls, context tokens, and estimated inference cost per question.

A useful summary metric may be a risk-weighted unsupported-claim rate, but the weighting scheme must be specified before evaluation and reported alongside unweighted counts. It should not obscure serious failures behind high performance on many trivial claims. Results should include confidence intervals or bootstrap uncertainty where appropriate and should be stratified by question type, source age, evidence conflict, answer length, and risk tier. If automated verification is itself assessed by another language model, a human-audited subset should estimate how often that judge agrees with expert annotators.

4.5. Reproducibility and Validity Controls

To make results reproducible, the study should publish or document the source corpus snapshot, chunking strategy, embedding model, lexical index configuration, retriever and reranker versions, generation prompt, decoding settings, verifier prompt or model, action thresholds, retry limits, and evaluation scripts. Each result should be linked to a system configuration rather than just a model name. If a closed-source model is used, its access date and exposed version identifier should be recorded because behavior can change over time.

Internal validity can be threatened by inconsistent model settings or by using different source collections for different baselines. Construct validity can be threatened when an automatic evaluator treats lexical overlap as factual support, or when a response-level score masks a few severe atomic errors. External validity may be limited if the corpus covers only English web documents or a single domain. A robust evaluation should include multiple task styles and disclose domains where the approach performs poorly. Human evaluators should be blinded to the system condition where practical, and adjudication rules should be written before score comparison.

4.6. Ethical and Data Governance Considerations

Retrieval can expose sensitive or copyrighted material, reproduce low-quality sources, or leak information across access boundaries. A deployment should enforce access controls at retrieval time, respect the license and terms governing indexed documents, minimize retention

of private queries, and avoid revealing restricted source passages in final answers. For sensitive domains, review should consider whether a generated claim could cause harm even if it is technically supported by one source. Provenance and uncertainty do not replace professional judgment or legal and organizational obligations.

The system should avoid presenting a probabilistic verifier label as an objective guarantee. Users should be able to distinguish an answer directly supported by a source from an inference or a response produced under incomplete evidence. When an error is reported, the audit trail should support correction of the source index, verifier rules, and response policy. Monitoring should measure both overconfident answers and excessive refusals, because either failure can undermine user trust and practical utility.

5. Discussion

5.1. Why Retrieval Alone Is Insufficient

RAG is a grounding architecture, not a guarantee of truth. The retriever estimates relevance, whereas the generator decides what to say, and neither operation inherently proves that a claim is factually correct. The system may retrieve a relevant but outdated source, mix facts from two entities, or fail to use a passage that is actually in the context. Long-context evidence also creates a selection problem: increasing the candidate set may improve recall while making evidence harder to attend to or increasing distraction [9]. A robust design therefore needs retrieval quality checks, claim-level verification, and an explicit policy for unresolved evidence.

A further issue is that evidence support and world truth are not identical. An answer may be perfectly faithful to a flawed source. For this reason, the framework treats source authority, date, provenance, and conflict resolution as first-class signals. For changing facts, the system should prefer sources that are authoritative for the relevant jurisdiction or domain and state the date of the information. Where high-quality sources disagree, presenting the disagreement may be more honest than compressing it into an artificially certain conclusion.

5.2. Balancing Factuality, Coverage, and Cost

Stricter verification generally adds model calls, retrieval latency, and engineering complexity. Requiring a citation for every clause can create a visually dense answer without necessarily improving factual precision; requesting several independent checks for every low-risk statement may add cost with little marginal value. The more practical approach is selective verification: devote the strongest checks to claims that are consequential, central to the answer,

unsupported by direct passages, temporally sensitive, numerically specific, or disputed by other sources. Low-risk claims with direct evidence can follow a lighter path, provided the policy is transparent and periodically evaluated.

The system must also balance precision against answer coverage. If it abstains whenever support is imperfect, it may become safe but unhelpful. If it always answers, it may appear useful while propagating unsupported claims. Risk–coverage analysis makes this trade-off visible by plotting error as the system answers a larger fraction of questions. The appropriate operating point depends on task consequences and user expectations rather than a universal threshold. The risk-sensitive controller proposed here is intended to make that choice explicit, tunable, and auditable.

5.3. Failure Modes and Limitations of the Proposed Framework

The framework has several limitations. First, claim extraction may split a statement incorrectly or omit a qualifier that is essential to its meaning. Second, the verifier may produce false support judgments, particularly for multi-hop reasoning, causal claims, tables, or documents containing complex exceptions. Third, retrieval quality depends on corpus coverage and source quality; no verification algorithm can recover evidence that is unavailable and no model can guarantee that every authoritative source is correct. Fourth, risk scores and impact tiers require domain-specific design and may introduce subjective assumptions. Fifth, repeated re-retrieval may create a feedback loop in which the system searches until it finds a passage consistent with its initial answer. A stopping rule and source-diversity policy are therefore necessary.

Sixth, evaluation benchmarks have finite scope. A model may perform well on standard fact-verification datasets yet fail on long answers, recent events, niche topics, multilingual sources, or the interaction of tables and prose. Seventh, judge models can share errors with the generator, creating correlated failure that is invisible when automated scores are treated as ground truth. The proposed protocol addresses these issues through human audits, source snapshots, error stratification, and component ablations, but the framework's effectiveness remains an empirical question until it is implemented and tested.

5.4. Directions for Future Research

Future work should test whether claim support thresholds can be calibrated across domains and model families, and whether independent verifier architectures reduce correlated error. More attention is needed for multilingual retrieval, temporal contradiction, structured data,

mathematical and causal statements, and source provenance across document transformations. Another direction is to train or adapt models to emit claim-level evidence pointers during generation, reducing the gap between answer creation and post-hoc verification. A third is to study human–AI collaboration: interfaces should expose uncertain claims and the evidence gap in ways that support efficient review rather than overwhelming reviewers with alerts.

Further studies should also explore whether source-level reliability estimates can be learned without reinforcing popularity bias, and how to evaluate conflicting sources when no single authority is definitive. Privacy-aware retrieval, permission-sensitive provenance, and defenses against malicious or prompt-injected documents are important operational extensions. These topics are complementary to factuality detection: a system that retrieves adversarial instructions or unauthorized data may generate a claim that looks evidence-supported while violating the intended security boundary.

6. Conclusion

This paper has proposed an evidence-grounded framework for detecting and mitigating factual hallucinations in large language models using retrieval-augmented generation. The central argument is that retrieval is necessary for many knowledge-intensive tasks but is not sufficient for factual reliability. An effective system must evaluate evidence coverage and quality, decompose answers into atomic claims, verify claim-to-source support, check citation alignment, and select mitigation actions according to uncertainty and risk. The framework combines these operations in an auditable pipeline and defines a proposed evaluation protocol that measures factual precision alongside retrieval quality, citation validity, answer utility, abstention, and cost.

The paper is conceptual and methodological. It provides a research blueprint based on established literature and deliberately reports no fabricated dataset, experiment, or performance improvement. Empirical work is needed to determine which components yield the greatest benefit, how their effects vary by domain, and where the cost of verification is justified. By making those questions testable and by preserving a trace from claims to evidence and decisions, the proposed framework aims to support more transparent and dependable RAG-based LLM applications.

References

[1] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of Hallucination in Natural Language Generation. ACM

Computing Surveys, 55(12), Article 248, 1–38. https://doi.org/10.1145/3571730

[2] Wang, Y., Wang, M., Manzoor, M. A., Liu, F., Georgiev, G. N., Das, R. J., & Nakov, P. (2024). Factuality of Large Language Models: A Survey. Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 19519–19529. https://doi.org/10.18653/v1/2024.emnlp-main.1088

[3] Lin, S., Hilton, J., & Evans, O. (2022). TruthfulQA: Measuring How Models Mimic Human Falsehoods. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 3214–3252. https://doi.org/10.18653/v1/2022.acl-long.229

[4] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Kuttler, H., Lewis, M., Yih, W.-T., Rocktaschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems, 33, 9459–9474. https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[5] Manakul, P., Liusie, A., & Gales, M. (2023). SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 9004–9017. https://doi.org/10.18653/v1/2023.emnlp-main.557

[6] Min, S., Krishna, K., Lyu, X., Lewis, M., Yih, W.-T., Koh, P., Iyyer, M., Zettlemoyer, L., & Hajishirzi, H. (2023). FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 12076–12100. https://doi.org/10.18653/v1/2023.emnlp-main.741

[7] 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: System Demonstrations, 150–158. https://doi.org/10.18653/v1/2024.eacl-demo.16

[8] Dhuliawala, S., Komeili, M., Xu, J., Raileanu, R., Li, X., Celikyilmaz, A., & Weston, J. (2024). Chain-of-Verification Reduces Hallucination in Large Language Models. Findings of the Association for Computational Linguistics: ACL 2024, 3563–3578. https://doi.org/10.18653/v1/2024.findings-acl.212

[9] Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., & Liang, P. (2024). Lost in the Middle: How Language Models Use Long Contexts. Transactions of the Association for Computational Linguistics, 12, 157–173. https://doi.org/10.1162/tacl_a_00638

[10] Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). FEVER: A Large-scale Dataset for Fact Extraction and VERification. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 809–819. https://doi.org/10.18653/v1/N18-1074