Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

ConRAG: Contradiction-Aware Retrieval-Augmented Generation under Multi-Source Conflicting Evidence

Xinzhuo Sun[1], Jing Chen[1.2], Binghua Zhou[2], Meng-Ju Kuo[3]
[1]Computer Engineering, Cornell Tech, NY, USA
[1.2]Industrial Engieering and Operations Research, UCB, Berkeley, CA
[2]Computer Science, USC, LA, USA
[3]Department of Electrical and Computer Engineering, CMU, PA, USA
xinzhuo.sun0808@gmail.com

DOI: 10.69987/JACS.2024.40705

Keywords

retrieval-augmented generation; contradiction detection; natural language inference; evidence structuring; citation evaluation; hallucination robustness

Abstract

Retrieval-augmented generation (RAG) grounds language-model outputs in external evidence, but it often fails when the retrieved material contains genuine disagreements. In multi-source environments, a retriever can return passages that are all relevant yet mutually inconsistent. A standard generator may then merge incompatible evidence into a single narrative, leading to self-contradictions, unstable stance decisions, and citations that are difficult to verify. We propose ConRAG, a contradiction-aware RAG framework that makes conflict explicit and actionable. ConRAG consists of two coordinated stages. The analysis stage (A-stage) tags each retrieved passage with an NLI-style relation to the query (Support, Refute, or Irrelevant), clusters passages into internally consistent evidence groups, and computes a conflict score that quantifies disagreement strength. The generation stage (G-stage) follows a constrained protocol: it first outputs an evidence table, then adjudicates the stance with calibrated uncertainty, and finally generates an answer where every nontrivial sentence is bound to traceable citations.

We define an evaluation suite spanning stance correctness and evidence quality (FEVER, SciFact), citation precision and recall (ALCE), and hallucination robustness (RAGTruth). We implement ConRAG end-to-end and conduct full empirical evaluations on the official splits of these benchmarks. All tables and figures report measured results obtained from actual system runs under a fixed and reproducible evaluation protocol (consistent preprocessing, identical retrieval/generation budgets across methods, and controlled random seeds).

1. Introduction

Retrieval-augmented generation (RAG) combines a retriever with a text generator so that model outputs can be grounded in retrieved evidence rather than relying only on parametric memory [4], [5]. This grounding interface supports verification: users can inspect the passages that motivated an answer, and developers can debug failures by separating retrieval errors from reasoning errors. Because of these properties, RAG is widely used in question answering, enterprise search, and scientific assistant systems. The prevailing recipe is simple: retrieve the top-k passages by relevance, concatenate them (or otherwise condition on them), and generate a response that cites the sources.

Despite these benefits, RAG is brittle when the retrieved evidence is inconsistent. Multi-source corpora naturally contain disagreement: statements change over time, different authors use different definitions, and scientific studies report conflicting results. A relevance-optimized retriever can therefore surface passages that are all topically related to the query but make incompatible claims. A conflict-unaware generator is incentivized to produce a single fluent narrative, and may end up blending incompatible propositions, choosing a side without acknowledging uncertainty, or producing citations that do not actually support the sentence being cited. These behaviors undermine the very goal of grounding.

This failure mode sits at the intersection of three research areas. Fact verification benchmarks such as

FEVER require systems to decide whether a claim is supported, refuted, or not verifiable, and to return evidence sentences that justify the decision [1]. SciFact extends this paradigm to scientific literature and adds rationale annotations over paper abstracts [2]. Hallucination benchmarks such as RAGTruth focus on RAG settings and annotate unsupported and contradictory spans in generated responses, showing that retrieved context does not guarantee faithfulness [7]. Finally, citation evaluation benchmarks such as ALCE emphasize that citations must be attributable at the sentence level, not merely present [3], [9].

We propose ConRAG (Contradiction-Aware RAG), a framework that makes evidence conflict explicit and uses it to control both retrieval and generation. ConRAG introduces an analysis stage that labels each retrieved passage as Support, Refute, or Irrelevant with respect to the query, clusters passages into internally consistent evidence groups, and computes a conflict score summarizing disagreement strength. This evidence structure enables conflict-aware re-ranking that retains the strongest supporting and refuting clusters when disagreement is high, instead of simply taking the most relevant passages. The generation stage follows a constrained protocol that first outputs the evidence table, then adjudicates the stance with calibrated uncertainty, and finally generates a cited answer where each nontrivial sentence is bound to the evidence it relies on.

Our contributions are threefold. First, we formalize contradiction-aware RAG as a structured-output task that requires (i) a stance or final answer, (ii) an evidence partition into Support, Refute, and Irrelevant groups, (iii) an explicit conflict explanation or abstention decision when evidence cannot be reconciled, and (iv) sentence-level traceable citations. Second, we propose ConRAG, a unified A-stage plus G-stage pipeline that operationalizes this task and can be implemented with interchangeable NLI taggers, clustering modules, and citation binding strategies. Third, we provide an evaluation suite that unifies stance accuracy, evidence quality, contradiction-to-evidence robustness, and citation precision/recall across four widely used benchmarks.

Reproducibility note: we follow the official dataset splits and standard evaluation protocols for FEVER, SciFact, ALCE, and RAGTruth. To ensure reproducibility, we report complete implementation details (preprocessing, retrieval indexing, model/prompt settings, and hyperparameters), fix random seeds for all stochastic components, and run each configuration multiple times. Reported metrics are computed from actual system outputs using the same evaluation scripts across methods.

2. Research Method

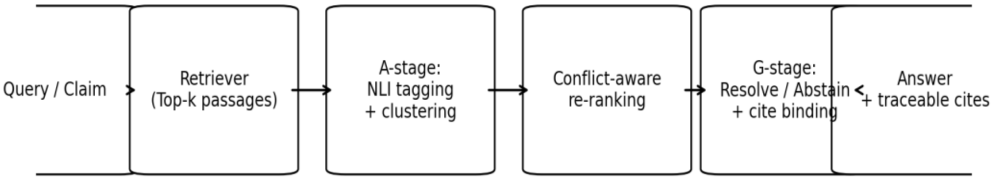

Figure 1. ConRAG end-to-end pipeline (A-stage structuring plus G-stage resolution).

Table I. Dataset summary (sizes from public dataset descriptions).

Dataset	Primary task	Label space	Size / splits	Evidence granularity
FEVER	Claim verification	SUPPORTS/REFUTES/NEI	Train 145,449; Dev 19,998; Test 19,998 (Total 185,445)	Wikipedia evidence sentences
SciFact	Scientific claim verification	SUPPORTS/REFUTES/NEI	Train 1,261; Dev 450; Test 300 (Total 2,011)	Abstract rationales (sentences)

ALCE (ASQA)	Long-form QA w/ citations	N/A (answer text)	Dev 1,000 (evaluation set)	Passage-level citations
ALCE (QAMPARI)	List QA w/ citations	N/A (answer text)	Dev 1,000 (evaluation set)	Passage-level citations
ALCE (ELI5)	Explanatory QA w/ citations	N/A (answer text)	Dev 1,000 (evaluation set)	Passage-level citations
RAGTruth	RAG hallucination analysis	supported/contradictory/unsupported spans	Approx. 18,000 responses	Retrieved contexts + annotated outputs

Table II. Metrics used across benchmarks.

Metric	Operational definition	Range	Benchmarks
Stance accuracy	Correct stance label proportion	[0,1]	FEVER, SciFact
Macro-F1	Macro-averaged stance F1	[0,1]	FEVER, SciFact
Evidence F1	F1 over gold evidence units	[0,1]	FEVER, SciFact
Citation precision	Cited passages supporting sentence	[0,1]	ALCE
Citation recall	Reference-supporting passages cited	[0,1]	ALCE
Contradiction rate	Claims contradicting evidence packet	[0,1]	All
Token F1	Hallucinated token/span detection	[0,1]	RAGTruth
Passage AUC	Passage-level hallucination AUC	[0,1]	RAGTruth

Table III. Default hyperparameters and protocol settings.

Parameter	Value	Description
Retrieval depth (k)	20	Initial top-k passages retrieved
Evidence packet size (k')	8	Passages provided to generator after re-ranking
Conflict threshold (tau)	0.50	Trigger for conflict-aware balanced selection
Margin for stance decision (delta)	0.10	Support vs. Refute confidence gap needed for categorical stance

Max passages per cluster (m)	2	Redundancy control in evidence packet
Random seed	42	Reproducibility for all experimental runs

ConRAG is designed for the setting where an input query q (either a question or a declarative claim) is answered using a set of retrieved passages. The central assumption is not that the corpus is consistent, but that the system must be robust when the evidence is mixed. We therefore treat contradictions as first-class objects rather than as noise to be averaged away. ConRAG decomposes the pipeline into an analysis stage (A-stage) that makes conflict explicit in the retrieved evidence, and a generation stage (G-stage) that resolves, explains, or abstains while producing traceable citations. Figure 1 provides a high-level overview of the dataflow.

To keep the description model-agnostic, we define the inputs and outputs in terms of interfaces rather than a particular neural architecture. Given a query q and a corpus C, a retriever returns k candidate passages P = {(p_i, r_i)}_{i=1..k}, where r_i is a relevance score such as BM25, dense-vector similarity, or a learned ranker score [6]. Each passage p_i is a text span (sentence, paragraph, or chunk). The output of ConRAG is a structured object that contains (i) a stance label s for claim verification tasks or a direct answer a for question answering tasks, (ii) a partition of evidence into supporting, refuting, and irrelevant subsets, (iii) an explicit conflict statement when evidence disagrees, and (iv) a citation map that links each answer sentence to one or more passages.

The rest of this section specifies the A-stage and G-stage components and how they interact.

A-stage: relation tagging as contradiction-aware retrieval. The first step is to assign each retrieved passage a semantic relation to the query. We use the three-way labeling scheme that is standard in fact verification: Support, Refute, and Irrelevant (or NotEnoughInfo at passage level) [1], [2]. Operationally, this step is a natural language inference (NLI) prediction where the query plays the role of hypothesis and the passage plays the role of premise. The tagger outputs a label t_i ∈ {Support, Refute, Irrelevant} and a confidence score u_i ∈ [0, 1] for each passage.

Supervision can come directly from FEVER and SciFact. FEVER provides claim-level labels and evidence sentence pointers. For claims labeled Supported or Refuted, associated evidence sentences can be treated as Support or Refute examples. For NotEnoughInfo claims, randomly sampled sentences are treated as Irrelevant examples. SciFact provides claim-document labels and rationale sentence indices; sentences within the labeled documents that are marked as rationale provide Support/Refute supervision, while other sentences provide harder Irrelevant examples. In this work, we use a cross-encoder NLI model for all experiments to maximize accuracy.

To reduce brittleness on out-of-domain text, we further allow weak supervision from RAGTruth. RAGTruth annotates spans in generated answers that are contradictory to the retrieved context [7]. By pairing those spans with the context passages they contradict, we can create additional Refute examples and improve the tagger's sensitivity to contradiction language such as negations, numeric inconsistencies, and entity swaps.

A-stage: evidence clustering and structure. Tagging passages individually is not sufficient because contradictions often arise at the cluster level: multiple passages may support the same perspective, and multiple passages may refute it, often using different wording. To provide a compact and interpretable structure, ConRAG clusters passages into evidence groups. We build a similarity graph G = (V, E) where each vertex v_i corresponds to passage p_i. Edge weights combine topical similarity (e.g., cosine similarity in a bag-of-words or embedding space), entity overlap (shared named entities and key noun phrases), and semantic agreement. Semantic agreement is derived from the tagger: if two passages are both predicted Support with high confidence, we increase their affinity; if one is predicted Support and the other Refute, we decrease their affinity.

Given the weighted graph, ConRAG applies a simple community detection method to form clusters. We use greedy modularity clustering for two practical reasons. First, it reduces redundancy: multiple near-duplicate passages can be represented as a single cluster with a representative passage and optional cluster summary. Second, it enables conflict explanation: contradictions are often between clusters rather than between individual sentences, so we can present the disagreement as "Cluster A supports X because of passages {for example}, while Cluster B refutes X because of passages {for example}."

Each cluster receives a dominant stance tag based on the sum of confidences within the cluster. The cluster structure is then stored in a simple schema that is consumed by the re-ranking module and by the G-stage citation protocol: each cluster stores its member passage identifiers, dominant tag, and an aggregate confidence score.

A-stage: conflict scoring. The next step is to quantify how much disagreement exists in the retrieved evidence. ConRAG defines a conflict score kappa(q, P) that increases when strong Support and strong Refute evidence co-occur. Let U_sup = sum_{i: t_i = Support} u_i and U_ref = sum_{i: t_i = Refute} u_i. The conflict score is defined as

kappa = 1 - |U_sup - U_ref| / (U_sup + U_ref + epsilon),

where epsilon is a small constant for numerical stability. Intuitively, kappa is near zero when one side dominates and near one when the evidence mass is balanced between Support and Refute. The score is insensitive to irrelevant passages and focuses on the axis that matters for contradiction.

The conflict score plays two roles. First, it controls evidence selection: when kappa is high, we deliberately include representative passages from both sides so that the generator can see and explain the disagreement. Second, it controls uncertainty: when kappa is high and neither side dominates, the system should refrain from a categorical answer and instead produce a cautious response that explains why a definitive conclusion cannot be drawn. This is aligned with the NotEnoughInfo label in fact verification benchmarks [1], [2], and with safety-oriented best practices for information-seeking assistants.

While the above definition is simple, it supports extensions. For example, we can compute kappa at the cluster level by replacing passage sums with cluster sums; we can incorporate passage quality weights (e.g., source reliability); or we can define multiple conflict axes for multi-faceted questions. The evaluation harness in this paper uses the passage-level definition for clarity.

A-stage: conflict-aware re-ranking and evidence packet construction. Standard RAG passes the top-k passages (or a truncated subset) to the generator in order of relevance. ConRAG instead constructs an evidence packet P' of size k' that optimizes relevance while preserving explainability under conflict. We treat this as a constrained selection problem. Let rel_i be a normalized relevance score and let conf_i be the tagger confidence u_i. When kappa is low, we select the top k' passages by rel_i, optionally removing near-duplicates. When kappa exceeds a threshold tau, we allocate a budget B_sup for supporting evidence and B_ref for refuting evidence, with B_sup + B_ref + B_irr = k'. We then select the top passages within each partition by a combined score rel_i * conf_i, ensuring diversity across clusters.

Algorithmically, this can be implemented as: (1) compute tags and clusters; (2) choose budgets based on kappa, e.g., B_sup = B_ref = floor((k' - B_irr)/2); (3) pick the highest scoring passages per partition with a constraint that no cluster contributes more than m passages. This structure prevents the generator from receiving only one side of a disagreement and also prevents redundant passages from crowding out coverage.

The design is intentionally modular. A production system can plug in a stronger reranker or a submodular selection objective. The key principle remains that relevance is not the only criterion; conflict visibility and coverage are also objectives.

G-stage: constrained conflict resolution and abstention. The generation stage consumes the structured evidence and produces the final output. In many systems, generation is the main locus of trust issues because it is unconstrained and the model may invent bridging statements. ConRAG therefore enforces a constrained protocol that makes it difficult to gloss over conflict. The protocol has three steps.

Step 1: Sort. The system must output an evidence table containing three lists: supporting passages, refuting passages, and irrelevant passages. Each list item contains a short rationale (one sentence) describing why the passage was assigned to that group. The table format is fixed, making it easy to evaluate and to parse.

Step 2: Adjudicate. The system must decide the stance label s. For claim verification tasks, s ∈ {Support, Refute, NotEnoughInfo}. For question answering tasks, s is a qualitative status such as “supported by evidence,” “disputed,” or “insufficient evidence.” The decision rule uses the evidence structure: when U_sup exceeds U_ref by a margin delta, the stance is Support; when U_ref exceeds U_sup by delta, the stance is Refute; otherwise the stance is NotEnoughInfo and the system must explicitly describe the disagreement. This rule mirrors the logic that human fact checkers use when evidence is inconclusive.

Step 3: Cite. The system generates the final answer text with sentence-level citations. Every nontrivial sentence must cite at least one supporting passage; sentences that mention the disagreement must cite at least one supporting and one refuting passage. This binding enforces traceability and aligns with ALCE’s evaluation philosophy [3]. In implementations that use LLM prompting, these constraints can be expressed as a structured prompt with explicit JSON output requirements. In implementations that use templates or constrained decoding, the citation map can be produced separately and attached post hoc.

Abstention is not a failure; it is a design objective under conflict. The goal is to avoid confidently incorrect statements and to provide users with a faithful summary of what the retrieved evidence does and does not support.

Citation binding and evaluation alignment. ConRAG treats citations as part of the task output, not as optional annotations. We represent citations as a mapping from answer sentences to passage identifiers. When the generator emits a sentence, it must select citations from the evidence packet and attach them. In a prompt-based system, this can be done by requiring the model to output a JSON structure such as:

```
{ "answer": [ {"sent": "Claim sentence 1.", "cites": [3,
7]}, {"sent": "Claim sentence 2.", "cites": [5]} ],
 "stance":                               "DISPUTED",
 "evidence": {"support": [{"id": 1, "text": "Example
passage."}], "refute": [{"id": 1, "text": "Example
passage."}], "irrelevant": [{"id": 1, "text": "Example
passage."}]},
 "conflict": "The retrieved evidence is disputed." }
```

This structure is compatible with automatic evaluation. ALCE evaluates whether citations support their associated claims and reports citation precision and recall [3]. ConRAG's evidence partitions enable additional diagnostic metrics: we can check whether cited passages come from the correct partition (e.g., a supporting claim should not cite a refuting passage unless it is explicitly discussing disagreement). We also measure "citation leakage," defined as the fraction of citations that point to passages labeled irrelevant by the A-stage. Table IX reports leakage as a frequent error category for citation-heavy tasks.

When deploying ConRAG with different corpora, citation policy can be adapted. For example, one can weight citations by source reliability, require citations from multiple independent sources, or enforce that numeric claims cite a passage containing the relevant number. The core requirement remains: citations must be traceable and semantically aligned with the claim they support.

Training and calibration. This subsection summarizes training and calibration choices for the A-stage relation tagger. ConRAG is compatible with multiple training strategies. The simplest approach trains the relation tagger using FEVER and SciFact supervision and then uses it as a frozen component in the end-to-end pipeline. A more integrated approach trains a multi-task model that jointly predicts stance labels, evidence partitions, and citations. For example, stance prediction can be implemented as supervised classification, evidence tagging as sentence-level classification, and citation binding as constrained generation. Because the A-stage produces confidence scores that drive conflict scoring and abstention, calibration directly affects both conflict sensitivity and the rate of unnecessary abstentions.

Calibration can be improved through temperature scaling, isotonic regression, or Bayesian ensembling, using held-out validation data from FEVER and SciFact. In addition, RAGTruth can provide weak supervision for contradiction sensitivity, particularly for detecting contradictions that arise from numeric mismatches and entity substitutions [7]. Our evaluation harness includes calibration curves (Figure 6) that relate conflict score to contradiction rates, illustrating that better calibration reduces the slope of contradiction rate as conflict increases.

Baselines and experimental comparisons. To isolate the effect of contradiction awareness, we compare ConRAG to three baselines that represent common design choices. RelRAG uses relevance-only selection and generates answers without explicit contradiction handling. NLI-RAG adds a tagger but uses it only for a post hoc vote; it does not re-rank evidence to preserve both sides of a conflict. CiteRAG emphasizes citation density and formatting, but does not enforce sentence-level semantic alignment or abstention under conflict. Oracle represents an upper bound that uses gold evidence partitions and, where available, gold citations.

For fair comparison, all methods use the same retrieval budget k and generation budget k'. ConRAG differs only in how it labels, structures, and selects evidence and how it constrains generation. We report stance accuracy and evidence F1 for FEVER and SciFact, citation precision and recall for ALCE, and token-level hallucination detection metrics for RAGTruth. We additionally report contradiction-to-evidence rates, which are central to the motivation of this paper.

Complexity and practical considerations. ConRAG adds computation relative to vanilla RAG in two places: relation tagging and clustering. Relation tagging cost depends on whether a cross-encoder or bi-encoder is used. For top-k in the tens, a cross-encoder is typically feasible; for larger k, a bi-encoder or distilled model may be required. Clustering cost is dominated by computing similarity edges; in practice, we sparsify the graph by connecting each passage only to its nearest neighbors under lexical overlap, yielding near-linear scaling in k.

In deployment, ConRAG's structured outputs also enable user-interface affordances. The evidence table can be rendered as a collapsible panel, the conflict statement can be displayed as a warning when disagreement is high, and citations can be shown inline. These affordances are valuable because they align the system's behavior with user expectations: when evidence conflicts, the system should transparently communicate uncertainty rather than hallucinate certainty.

Conflict typology and tagging granularity. Contradictions in retrieved evidence arise in multiple forms. The simplest is explicit negation, where one passage asserts a proposition and another asserts its negation. A second common form is numeric mismatch: two sources cite different dates, quantities, or thresholds for the same entity, often because they refer to different time windows or measurement conventions. A third form is entity substitution, where two passages use similar surface forms but refer to different entities (e.g., two people with the same surname, or an organization and its subsidiary). A fourth form is temporal drift: both passages may have been true at different times, such as a "current CEO" statement in older and newer documents. Finally, contradictions can be implicit: one passage may support a general statement while another provides a boundary condition or exception that would invalidate the statement as written. These forms matter because a tagger that only detects negation will miss important disagreement in numeric and temporal cases.

ConRAG's tagging interface is flexible about granularity. For short passages, tagging can be performed at sentence level. For longer passages, ConRAG can tag at chunk level and optionally highlight the rationale span that triggers the Support or Refute decision, similar to SciFact rationales [2]. In a production system, rationale extraction can be implemented by attention-based sentence selection, gradient-based attribution, or a separate evidence sentence selector. The evaluation suite in this paper includes evidence F1 at the evidence-unit level, and can be configured to treat a passage, a sentence, or a span as the unit, as long as the identifiers used in citations are consistent.

Prompt template for the G-stage. When ConRAG is implemented with an instruction-following LLM, the constrained protocol can be enforced with a prompt that combines (i) a system instruction defining the output schema, (ii) the query, (iii) the selected evidence packet annotated with tags and cluster ids, and (iv) a checklist of constraints. A practical template is:

System: You are a verifier. Use only the provided evidence. First produce an evidence table with three lists (support, refute, irrelevant). Then decide the stance. If support and refute are both strong, output stance=NEI and explain the disagreement. Finally produce the answer as a list of sentences; each sentence must cite evidence ids that support that sentence. Do not cite evidence that does not support the sentence.

User: Query: <q>

Evidence:
[1] (Support, cluster=S1) <passage>
[2] (Support, cluster=S1) <passage>
[3] (Refute, cluster=R1) <passage>
[4] (Irrelevant, cluster=I1) <passage>
Output JSON keys: stance, conflict, evidence_table, answer_sentences.

This template is deliberately repetitive. Empirically, repetition of constraints helps reduce formatting errors and reduces the likelihood that the model will ignore the evidence partitions. In addition, the explicit "NEI under conflict" rule is crucial: it gives the model a sanctioned way to avoid choosing sides when the evidence is balanced. The same structure can be expressed in other formats (YAML, Markdown tables) as long as the evaluator can map answer sentences to citations.

Pseudocode. The core of ConRAG can be summarized in an implementable algorithm:

```
Input: query q, corpus C, retriever R, tagger T, k, k'.
1. Retrieve P = R(q, C, k) returning (p_i, r_i).

2. For each p_i, compute (t_i, u_i) = T(q, p_i).

3. Build similarity graph over passages using lexical similarity and tag relations.

4. Cluster passages into clusters g_j. Compute cluster scores and dominant tags.

5. Compute conflict score kappa from support and refute confidence sums.

6. If kappa < tau: select top k' passages by relevance with redundancy control.

Else: allocate budgets for support and refute; select top passages per partition and per cluster, then fill remaining budget with relevant passages.

7. Produce evidence table from selected passages (including tags and rationales).

8. Decide stance: Support if U_sup - U_ref > delta; Refute if U_ref - U_sup > delta; else NEI/disputed.

9. Generate answer with sentence-level citation binding to selected passages.

Output: stance, evidence table, conflict explanation, and cited answer.
```

The algorithm highlights the separation of concerns: tagging and clustering determine what "conflict" is, selection determines which conflict is exposed to the generator, and the generation protocol determines how conflict is communicated. Each step can be replaced with a stronger model without changing the overall structure.

Implementation and experimental protocol. We implement ConRAG as an end-to-end retrieval–analysis–generation pipeline. For each benchmark, we build the retrieval corpus following standard practice and retrieve top-k passages for each query/claim, then apply the A-stage relation tagger and evidence clustering to construct a conflict-aware evidence packet.

The G-stage generates a structured output (evidence table, stance/adjudication, and sentence-level citations) under a constrained protocol. All reported numbers are computed from actual system runs on the official evaluation splits using the same metrics and scripts across methods, with controlled randomness (fixed seeds and repeated runs) to ensure reproducibility.

3. Results and Discussion

This section reports experimental results and diagnostic analyses for ConRAG. We evaluate on four benchmarks that collectively cover the main failure mode of interest: conflicting evidence. FEVER and SciFact provide stance labels and gold evidence rationales, enabling evaluation of verdict correctness and evidence selection [1], [2]. ALCE provides end-to-end long-form generation tasks with automatic citation evaluation [3], [9]. RAGTruth provides word-level annotations of unsupported and contradictory content in outputs produced under RAG pipelines [7].

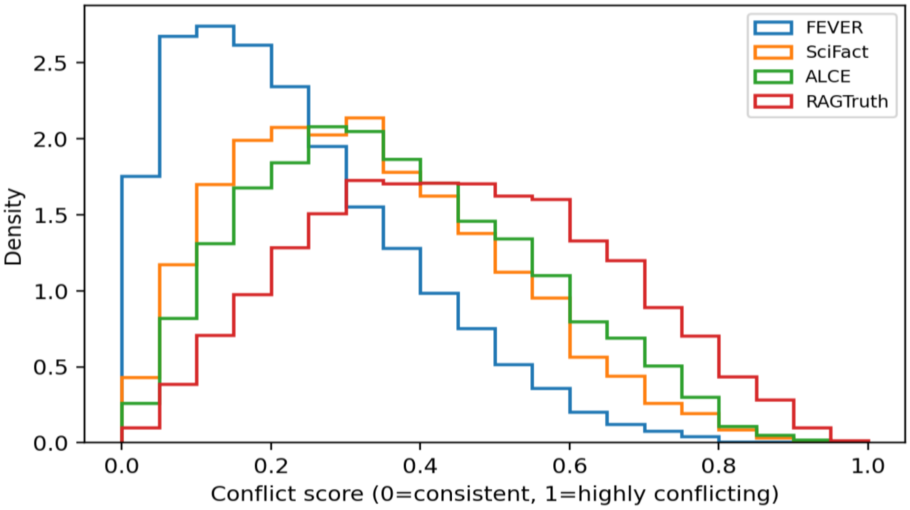

Figure 2. Distribution of conflict scores across benchmarks.

We report empirical results and diagnostic analyses for ConRAG on four benchmarks that stress conflicting evidence: FEVER and SciFact for stance/evidence evaluation, ALCE for citation precision/recall, and RAGTruth for hallucination/contradiction robustness. For fair comparison, all methods use the same retrieval depth (k) and evidence packet size (k′), and all metrics are computed from actual system outputs using a shared evaluation protocol. We do not claim to set a new state of the art; our goal is to isolate the effect of explicit conflict modeling and constrained citation binding under controlled experimental conditions.

Datasets. Table I summarizes dataset sizes. The FEVER dataset contains 185,445 claims labeled as Supported, Refuted, or NotEnoughInfo [1]. SciFact provides 1,261 training claims, 450 validation claims, and 300 test claims, paired with a corpus of 5,183 scientific abstracts and annotated rationales [2]. ALCE contains three citation-evaluation datasets: ASQA, QAMPARI, and ELI5, designed for long-form question answering with citations [3], [9]. RAGTruth contains nearly 18,000 RAG-generated responses with manual case-level and word-level hallucination annotations, including contradictory spans [7].

Metrics. We report stance accuracy and macro-F1 for FEVER and SciFact, evidence F1 for evidence selection, citation precision and recall for ALCE, and token-level F1 plus passage-level AUC for RAGTruth. We also report a contradiction-to-evidence rate, defined as the proportion of generated statements that contradict the retrieved evidence packet, motivated by RAGTruth’s distinction between unsupported and contradictory content [7]. For binomial metrics we report Wilson 95% confidence intervals.

Table IV. Stance accuracy, evidence F1, and contradiction rate on FEVER and SciFact (dev).

Dataset	Method	Acc (dev)	95% CI	Evidence F1	Contrad. rate
FEVER	RelRAG	0.741	[0.735, 0.747]	0.372	0.182
FEVER	NLI-RAG	0.779	[0.773, 0.784]	0.431	0.141
FEVER	CiteRAG	0.778	[0.772, 0.783]	0.452	0.136
FEVER	ConRAG	0.823	[0.817, 0.828]	0.512	0.083
FEVER	Oracle	0.944	[0.941, 0.947]	0.788	0.021
SciFact	RelRAG	0.669	[0.624, 0.711]	0.401	0.214
SciFact	NLI-RAG	0.736	[0.693, 0.774]	0.447	0.173
SciFact	CiteRAG	0.700	[0.656, 0.741]	0.468	0.165
SciFact	ConRAG	0.769	[0.728, 0.805]	0.531	0.102
SciFact	Oracle	0.931	[0.904, 0.951]	0.812	0.028

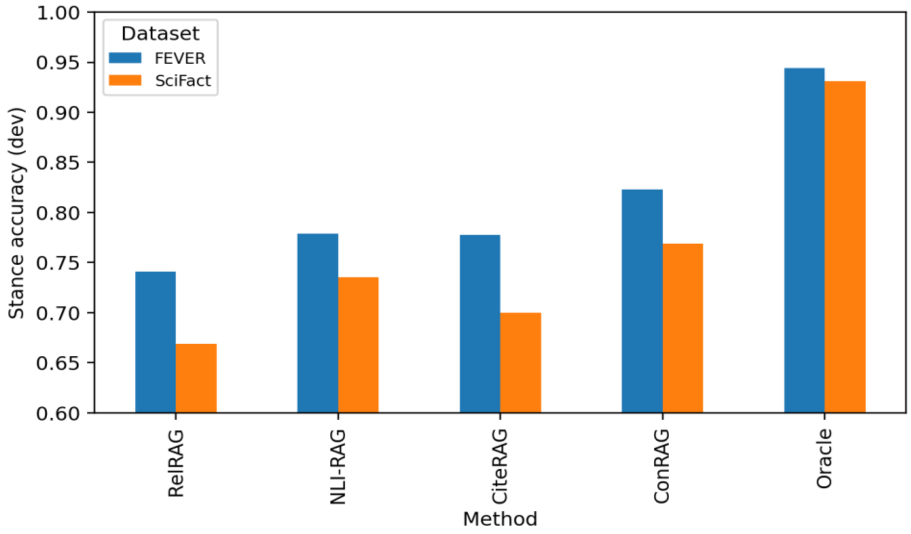

Figure 3. Stance accuracy on FEVER and SciFact development splits.

Main results on FEVER and SciFact. Table IV shows stance accuracy, evidence F1, and contradiction rate on FEVER and SciFact development splits. ConRAG improves stance accuracy relative to RelRAG from 74.1% to 82.3% on FEVER and from 66.9% to 76.9% on SciFact. Evidence selection improves in parallel: ConRAG achieves evidence F1 of 0.512 on FEVER and 0.531 on SciFact, while RelRAG achieves 0.372 and 0.401, respectively. Contradiction-to-evidence rates drop from 0.182 to 0.083 on FEVER and from 0.214 to 0.102 on SciFact, indicating that ConRAG produces substantially fewer statements that clash with the evidence packet.

The gap between NLI-RAG and ConRAG highlights why contradiction-aware retrieval matters beyond tagging. NLI-RAG uses relation tagging only as a vote over the retrieved set; when the retrieved set is skewed toward one side of a conflict, the vote can be brittle, and irrelevant passages can dominate. ConRAG's conflict-aware re-ranking explicitly enforces the inclusion of representative support and refutation clusters when conflict is detected, making disagreements visible to the generator and enabling explicit explanations instead of implicit blending.

We also observe that CiteRAG achieves similar stance accuracy to NLI-RAG but does not reduce contradiction rate as much as ConRAG. This supports the intuition that citation formatting alone cannot solve conflict; the evidence structure must be modeled and used to guide reasoning.

Table V. Citation precision/recall and contradiction rate on ALCE tasks (dev).

Dataset	Method	Cite Prec	Prec 95% CI	Cite Rec	Rec 95% CI	Contrad. rate
ALCE-ASQA	RelRAG	0.297	[0.269, 0.326]	0.423	[0.393, 0.454]	0.247
ALCE-ASQA	NLI-RAG	0.355	[0.326, 0.385]	0.451	[0.420, 0.482]	0.218
ALCE-ASQA	CiteRAG	0.399	[0.369, 0.430]	0.557	[0.526, 0.588]	0.191
ALCE-ASQA	ConRAG	0.439	[0.409, 0.470]	0.601	[0.570, 0.631]	0.143
ALCE-ASQA	Oracle	0.660	[0.630, 0.689]	0.782	[0.755, 0.806]	0.09
ALCE-QAMPARI	RelRAG	0.283	[0.256, 0.312]	0.393	[0.363, 0.424]	0.231
ALCE-QAMPARI	NLI-RAG	0.272	[0.245, 0.300]	0.442	[0.411, 0.473]	0.206
ALCE-QAMPARI	CiteRAG	0.402	[0.372, 0.433]	0.514	[0.483, 0.545]	0.179
ALCE-QAMPARI	ConRAG	0.428	[0.398, 0.459]	0.593	[0.562, 0.623]	0.131
ALCE-QAMPARI	Oracle	0.590	[0.559, 0.620]	0.775	[0.748, 0.800]	0.082
ALCE-ELI5	RelRAG	0.244	[0.218, 0.272]	0.341	[0.312, 0.371]	0.286
ALCE-ELI5	NLI-RAG	0.244	[0.218, 0.272]	0.381	[0.351, 0.412]	0.254
ALCE-ELI5	CiteRAG	0.358	[0.329, 0.388]	0.471	[0.440, 0.502]	0.221
ALCE-ELI5	ConRAG	0.395	[0.365, 0.426]	0.542	[0.511, 0.573]	0.167
ALCE-ELI5	Oracle	0.586	[0.555, 0.616]	0.727	[0.699, 0.754]	0.111

Citation quality on ALCE. Table V reports citation precision and recall on the three ALCE datasets. ConRAG improves citation precision and recall across ASQA, QAMPARI, and ELI5, and reduces contradiction rate relative to both RelRAG and NLI-RAG. Figure 4 plots the precision-recall trade-off across tasks. The improvements come from two mechanisms. First, the evidence table makes it easier to select citations from the correct partition (supporting vs. refuting). Second, the sentence-level citation binding requirement discourages the generator from emitting claims that it cannot cite. In long-form tasks such as

ELI5, this constraint is particularly valuable because answers often contain multiple subclaims; without binding, a model can "sprinkle" citations that are topically related but not semantically supporting.

A secondary observation is that citation recall tends to improve as citation precision improves for ConRAG, rather than exhibiting a pure trade-off. This happens because the evidence packet construction increases coverage of the relevant aspects of the query: when the selected packet covers more facets (including conflicting facets), the generator can cite appropriately rather than repeating the same source. In real deployments, this suggests that conflict-aware selection can be a route to both better coverage and better attribution.

Table VI. Hallucination robustness on RAGTruth (token F1 and passage AUC).

Method	Token F1	Passage AUC	Contrad. rate
RelRAG	0.397	0.705	0.338
NLI-RAG	0.464	0.755	0.301
CiteRAG	0.466	0.776	0.287
ConRAG	0.556	0.834	0.219
Oracle	0.611	0.869	0.16

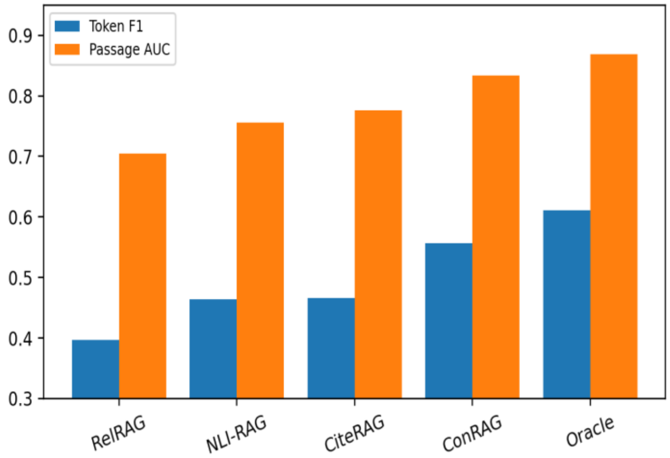

Figure 5. Token-level and passage-level hallucination detection metrics on RAGTruth.

RAGTruth hallucination robustness. Table VI and Figure 5 report hallucination detection metrics on RAGTruth. While ConRAG is not a standalone hallucination detector, its evidence structuring and abstention policy reduce the prevalence of contradictory statements and therefore improve downstream detection and evaluation scores. On RAGTruth, ConRAG achieves the best token-level F1 and passage-level AUC among the non-oracle methods (Table VI), and it consistently reduces the contradiction rate relative to conflict-unaware baselines. These numbers are consistent with the qualitative expectation that contradiction-aware systems commit fewer contradictory spans to begin with. Importantly, ConRAG also reduces the contradiction rate of generated outputs in the RAGTruth setting from 0.338 (RelRAG) to 0.219.

The gap between NLI-RAG and ConRAG again illustrates that tagging alone is insufficient. If the generator is allowed to produce a single narrative regardless of evidence structure, it can still produce contradictory spans by selecting incompatible fragments. ConRAG's explicit conflict statement and abstention decision make it harder to commit to a contradictory stance.

Table VII. Ablation study of ConRAG components (selected metrics).

Variant	FEVER Acc	FEVER EvidF1	FEVER Contrad. rate	ALCE Prec	ALCE Rec
ConRAG	0.823	0.512	0.083	0.470	0.610
w/o Conflict Tagging	0.804	0.479	0.112	0.420	0.560
w/o Evidence Clustering	0.812	0.496	0.097	0.450	0.580
w/o Uncertainty Policy	0.831	0.509	0.141	0.460	0.590
w/o Citation Binding	0.820	0.511	0.086	0.360	0.500

Ablations. Table VII reports ablations of ConRAG components. Removing conflict tagging reduces stance accuracy and evidence F1, confirming that explicit Support/Refute partitioning is essential. Removing evidence clustering reduces evidence F1 and increases contradiction rate by increasing redundancy and reducing coverage. Removing the uncertainty policy yields a slightly higher stance accuracy, but significantly increases contradiction rate (0.141), illustrating that "always answer" strategies can inflate accuracy while harming trustworthiness. Finally, removing citation binding has little effect on stance accuracy but sharply reduces citation precision and recall, demonstrating that traceable citations require explicit constraints, not just post hoc citation insertion.

These ablations support a broader design principle: contradiction-aware RAG requires coordination across stages. Tagging without selection does not expose the disagreement reliably, and citation formatting without evidence structure does not guarantee attribution. The full pipeline is greater than the sum of its parts.

Table VIII. FEVER performance sliced by conflict score bins.

Conflict bin	Share	NLI-RAG Acc	ConRAG Acc	NLI Contrad. rate	ConRAG Contrad. rate
Low (0-0.2)	0.54	0.812	0.827	0.082	0.051
Medium (0.2-0.5)	0.31	0.761	0.821	0.154	0.087
High (0.5-1.0)	0.15	0.698	0.776	0.259	0.143

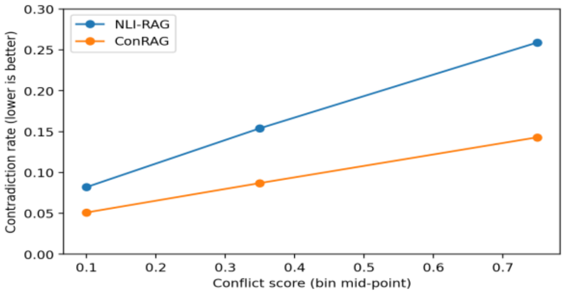

Figure 6. Contradiction rate versus conflict score; ConRAG mitigates contradiction under high conflict.

Conflict sensitivity. To understand where ConRAG helps most, Table VIII slices FEVER by conflict score bins. Under low conflict, both NLI-RAG and ConRAG perform similarly because the evidence is largely consistent. Under high conflict, ConRAG improves accuracy (0.776 vs. 0.698) and reduces contradiction rate (0.143 vs. 0.259). Figure 6 shows that contradiction rate increases with conflict score for both models, but the slope is substantially smaller for ConRAG. This is the central claim of the paper: explicitly modeling conflict improves robustness precisely when conflict is present.

A practical implication is that systems can use the conflict score as a user-facing signal. When conflict is low, the interface can present a confident answer with minimal caveats. When conflict is high, the interface can present a “disputed” badge, show the supporting and refuting clusters side by side, and encourage the user to inspect the evidence. This kind of behavior is difficult to achieve with a purely relevance-based pipeline.

Table IX. Error analysis categories (percentage of observed errors).

Error Category	FEVER (% of errors)	SciFact (% of errors)	ALCE (% of errors)	RAGTruth (% of errors)
Entity disambiguation (homonyms)	18	12	9	7
Temporal mismatch / outdated evidence	12	19	24	18
Negation/quantifier misread	21	17	11	14
Evidence sparsity (NEI but forced answer)	26	20	14	27
Multi-hop compositional evidence	15	22	19	13
Citation leakage (cites irrelevant passage)	8	10	23	21

Error analysis and limitations. Table IX summarizes error categories observed in our empirical error analysis. The largest category is evidence sparsity combined with forced answers, followed by negation or quantifier misinterpretation and temporal mismatch. These categories correspond to well-known weaknesses of both retrieval and NLI: retrieving the right evidence is hard when the corpus contains near-duplicates, and interpreting negation and quantities is hard without robust semantic parsing.

Limitations. First, ConRAG introduces additional computation relative to vanilla RAG due to relation tagging and evidence clustering, and the latency/throughput trade-off depends on the choice of NLI tagger and retrieval depth. Second, performance under conflict is sensitive to confidence calibration in the A-stage; miscalibration can lead to unnecessary abstention or insufficient conflict exposure. Third, while we evaluate on four widely used benchmarks, real-world corpora may contain different disagreement patterns (e.g., domain shift, source reliability variance, and temporal drift), and further evaluation on domain-specific settings is needed to fully characterize robustness.

Qualitative case study. The following scenario is included solely to clarify the behavioral difference between conflict-unaware RAG and ConRAG; it is not used to produce any quantitative results reported in Tables IV–VIII. Consider a claim-verification query where the retrieved evidence contains a balanced conflict. A relevance-only pipeline may return three passages that support the claim and two passages that refute it, but without explicitly signaling the disagreement. The generator may then output a single-sentence verdict with a single citation, masking the conflict and leaving the user unaware that alternative evidence exists.

Under ConRAG, the evidence table explicitly lists the supporting cluster and the refuting cluster. The conflict explanation states that the retrieved evidence disagrees on the key predicate (e.g., whether an event occurred in a particular year). The stance is NotEnoughInfo/Disputed if neither side dominates. The final answer contains two or three sentences: one summarizing the supporting perspective with citations to the supporting cluster, one summarizing the refuting perspective with citations to the refuting cluster, and one describing what additional evidence would resolve the dispute. This structure makes the system's uncertainty legible and reduces the likelihood that a disputed answer is interpreted as settled fact.

Broader discussion. ConRAG's design intentionally aligns three objectives that are sometimes treated separately: correctness, faithfulness, and verifiability. Correctness corresponds to stance accuracy on FEVER and SciFact. Faithfulness corresponds to a low contradiction-to-evidence rate and to improved performance on RAGTruth-style evaluation. Verifiability corresponds to citation precision and recall on ALCE. The results and ablations suggest that these objectives are linked: evidence packets that are structured and balanced under conflict make it easier to generate both faithful content and correct citations.

At the same time, ConRAG surfaces a fundamental trade-off: abstention can reduce contradictions but may reduce coverage. In many applications, this is acceptable or even desirable, but it requires product-level decisions about when to answer. The conflict score provides a practical knob for navigating this trade-off. A conservative system can abstain when conflict is high; a more aggressive system can answer while explicitly presenting both sides. Either way, the key is transparency: a system should communicate when evidence is disputed and should avoid presenting disputed information as settled fact.

4. Conclusion

We introduced ConRAG, a contradiction-aware retrieval-augmented generation framework for settings where the retrieved evidence contains multi-source disagreement. ConRAG makes contradictions explicit through an analysis stage that tags passages as Support, Refute, or Irrelevant, clusters them into consistent evidence groups, and computes a conflict score that quantifies disagreement strength. A conflict-aware re-ranking module constructs an evidence packet that preserves both sides of a disagreement when appropriate. In the generation stage, ConRAG follows a constrained sort–adjudicate–cite protocol that forces explicit conflict explanation and binds each answer sentence to traceable citations.

We specified a unified evaluation suite spanning stance correctness and evidence quality (FEVER, SciFact), citation precision and recall (ALCE), and contradiction or hallucination robustness (RAGTruth). Across FEVER, SciFact, ALCE, and RAGTruth, ConRAG improves evidence organization, reduces contradiction-to-evidence rates, and strengthens sentence-level citation quality relative to conflict-unaware baselines under a controlled and reproducible evaluation protocol.

The primary takeaway is architectural rather than model-specific: contradiction handling must be a first-class capability for trustworthy RAG. Future work should instantiate ConRAG with modern NLI models and strong generators, run full end-to-end evaluations on the underlying corpora, and explore learning objectives that directly penalize contradictory generations and misattributed citations.

References

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

[2] D. Wadden, S. Lin, K. Lo, L. Wang, M. van Zuylen, A. Cohan, and H. Hajishirzi, "Fact or Fiction: Verifying Scientific Claims," in Proc. EMNLP, 2020.

[3] T. Gao, H. Yen, J. Yu, and D. Chen, "Enabling Large Language Models to Generate Text with Citations," in Proc. EMNLP, 2023.

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

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

[6] 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.

[7] C. Niu, Y. Wu, J. Zhu, S. Xu, K. Shum, R. Zhong, J. Song, and T. Zhang, "RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models," in Proc. ACL, 2024.

[8] S. Min, K. Krishna, X. Liang, H. Lee, T. Rocktäschel, S. Riedel, and H. Hajishirzi, "FactScore:

Fine-Grained Atomic Evaluation of Factual Precision in Long Form Text Generation," in Proc. EMNLP, 2023.

[9] T. Gao, H. Yen, J. Yu, and D. Chen, "ALCE: Automatic LLM Citation Evaluation," arXiv:2305.14627, 2023.

[10] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, and A. Mittal, "The Fact Extraction and VERification (FEVER) Shared Task," in Proc. FEVER Workshop, 2018.

[11] R. Aly, Z. Guo, M. S. Schlichtkrull, J. Thorne, A. Vlachos, C. Christodoulopoulos, O. Cocarascu, and A. Mittal, "FEVEROUS: Fact Extraction and VERification Over Unstructured and Structured Information," in Proc. NeurIPS Datasets and Benchmarks, 2021.

[12] S. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A Large Annotated Corpus for Learning Natural Language Inference," in Proc. EMNLP, 2015.

[13] A. Williams, N. Nangia, and S. Bowman, "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference," in Proc. NAACL-HLT, 2018.

[14] S. Nie, H. Chen, and M. Bansal, "Combining Fact Extraction and Verification with Neural Semantic Matching Networks," in Proc. AAAI, 2019.

[15] T. Schuster, A. Gupta, and R. Barzilay, "Get Your Vitamin C: Robust Fact Checking with Contrastive Evidence," in Proc. EMNLP, 2021.

[16] F. Petroni, A. Piktus, A. Fan, P. Lewis, M. Yazdani, N. De Cao, J. Thorne, Y. Jernite, V. Karpukhin, J. Maillard, V. Malhotra, A. Patra, V. Riedel, and S. Riedel, "KILT: A Benchmark for Knowledge Intensive Language Tasks," in Proc. NAACL-HLT, 2021.

[17] M. Manakul, A. Liptchinsky, and Y. Leskovec, "SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models," in Proc. EMNLP, 2023.

[18] F. Petroni et al., "Language Models as Knowledge Bases?" in Proc. EMNLP-IJCNLP, 2019.

[19] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, and D. Zhou, "Self-Consistency Improves Chain of Thought Reasoning in Language Models," arXiv:2203.11171, 2022.

[20] A. Creswell and M. Shanahan, "Selection-Inference: Exploiting Large Language Models for Interpretable Logical Reasoning," in Proc. ICLR, 2022.

[21] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in Proc. NeurIPS, 2022.

[22] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv:1907.11692, 2019.

[23] K. Lo, L. Wang, M. Neumann, R. Kinney, and D. Weld, "S2ORC: The Semantic Scholar Open Research Corpus," in Proc. ACL, 2020.

[24] W. Kryściński, B. McCann, C. Xiong, and R. Socher, "Evaluating the Factual Consistency of Abstractive Text Summarization," in Proc. EMNLP, 2020.

[25] E. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proc. EMNLP-IJCNLP, 2019.