

Retrieving Facts or Reinforcing Beliefs? Towards Understanding Confirmation Bias in RAG Systems for Scientific Claim Verification

Qile Wan¹ Davide Venditti^{2,3} Fabio Massimo Zanzotto^{3,5}
Iryna Gurevych⁴ Lizhen Qu^{1*}

¹ Monash University, Clayton, Victoria, Australia

² Department of Computer Science, University of Luxembourg

³ Human-Centric ART, University of Rome Tor Vergata

⁴ Ubiquitous Knowledge Processing Lab (UKP), Technical University of Darmstadt, Darmstadt, Germany

⁴ Department of Computer Science and Hessian Center for AI (hessian.AI), TU Darmstadt

⁵ Almaywave Labs, Italy

qile.wan@monash.edu

davide.venditti@uni.lu

fabio.massimo.zanzotto@uniroma2.it

iryna.gurevych@tu-darmstadt.de

lizhen.qu@monash.edu

Abstract

The rapid growth of scientific literature has made automated verification of scientific claims increasingly important, with Retrieval-Augmented Generation (RAG) systems emerging as a promising solution. However, their multi-stage architecture raises concerns about confirmation bias, where systems preferentially retrieve and generate evidence that supports a given claim. In this paper, we present CoBiT (Confirmation Bias Test), the first benchmark for systematically evaluating confirmation bias in RAG systems, consisting of 707 minimally perturbed claim-counterclaim pairs. We propose a unified framework with stage-specific metrics to quantify bias in both re-ranking and generation, including Ranking Bias Magnitude (RBM), Confirmation Bias Rate (CBR), and consistency measures. Our experiments on OpenScholar, ScholarQA, and LLM baselines reveal that confirmation bias arises at different stages of the RAG pipeline and can be amplified by re-ranking and generation. In particular, SciRAG achieves the lowest generation bias (CBR = 0.13), it exhibits higher bias in re-ranking, and OpenScholar shows the greatest inconsistency between claims and counterclaims. These findings highlight the need for a fine-grained evaluation of epistemic risks in RAG systems and provide practical tools to diagnose confirmation bias.¹

1 Introduction

The exponential expansion of scientific literature has rendered the manual verification of scientific claims increasingly intractable, necessitating automated, evidence-based validation frameworks (Wadden et al., 2022a).

* Corresponding author.

¹Code and Data available at [CoBiT](#)

Original Claim Example

Claim: CRISPR-Cas9 gene editing **decreases** off-target mutations despite low-fidelity variants.

Retrieval: Mixed

Rerank: All Support

Score: 8.42

RAG Output: CRISPR-Cas9 gene editing has been shown to decrease off-target mutations despite the presence of low-fidelity variants (Mengstie et al. 2024). ...

Perspective of the Output: Support

Generation Bias Label: Unbiased

Counter Claim Example

Claim: CRISPR-Cas9 gene editing **increases** off-target mutations despite low-fidelity variants.

Retrieval: Mixed

Rerank: Mixed

Score: -7.91

RAG Output: CRISPR-Cas9 gene editing is a powerful tool for making targeted changes to DNA, but it can also increase off-target mutations (Guo et al. 2023). ...

Perspective of the Output: Support

Generation Bias Label: Confirmation Bias

Figure 1: Example outputs generated by OpenScholar for a claim and its counter claim. We report metrics of confirmation bias for both re-ranking and generation.

Retrieval-Augmented Generation (RAG), which combines retrieval mechanisms with powerful Large Language Models (LLMs), provides a scalable and contextually grounded solution (Fan et al., 2024). Recent advances have led to the emergence of *Deep Research* systems (Xu and Peng, 2025), including models such as OpenScholar (Asai et al., 2024), ScholarQA (Singh et al., 2025), and proprietary systems, which generate responses grounded in explicitly cited sources. As RAG systems are increasingly adopted by researchers, it is critical to

investigate whether retriever, ranking mechanisms or generator evidence selection may inadvertently reinforce *confirmation bias*, that is, the tendency to seek and interpret evidence in ways that confirm existing beliefs or hypotheses (Stewart, 2024).

As illustrated in Fig. 1, for the scientific claim “CRISPR-Cas9 gene editing *decreases* off-target mutations despite low-fidelity variants.” and its corresponding counter claim obtained by replacing *decreases* with *increases*, OpenScholar retrieves only supporting evidence and generates a synthesis supporting the queried claim, indicating confirmation bias despite the change of a single word.

Although recent work (Wong et al., 2025) *qualitatively* identifies confirmation bias in proprietary medical RAG systems, automatic detection and quantification of such biases in scientific claim verification remain underexplored, as does the contribution of individual pipeline stages.

To automate the evaluation of RAG models for scientific claim verification, we construct CoBiT (**C**onfirmation **B**ias **T**est), the *first* benchmark consisting of 707 claim-counterclaim pairs and a set of novel metrics to quantify confirmation bias.

We focus on the scientific domain because its strong logical structure and objective factual grounding make it ideal for isolating confirmation bias. This explicit setting allows for precise evaluation without the confounding ambiguities of more subjective topics.

We use claim-counterclaim pairs because scientific claim verification is largely studied as a bipolar task of validating or refuting claims against evidence, making the stance-flipped pair the natural unit of analysis, though confirmation bias may surface in subtler multi-perspective forms elsewhere. Because typical RAG systems re-rank top- k evidence and then generate a synthesis, we design stage-specific metrics for each.

Our metrics and empirical findings are summarized as follows: (i) For the re-ranking module, which produces top- k lists of retrieved documents, we propose *Ranking Bias Magnitude* (RBM), a novel metric that learns a parametric linear function to measure the similarity between a target ranking and reference (biased and unbiased) rankings; the same function yields Var_r , capturing bias consistency between a claim and its counterclaim. (ii) For generation, we introduce rule-based metrics: *Confirmation Bias Rate* (CBR), the proportion of claims exhibiting confir-

mation bias, and Var_g , which assesses bias consistency between a claim and its counterclaim in generated responses. (iii) Experiments show that SciRAG (Ding et al., 2026) achieves the lowest generation bias (CBR=0.13), ScholarQA’s entity-focused retrieval and re-ranking yield the lowest RBM, while OpenScholar shows the highest claim-counterclaim inconsistency under Var_g , worse than all evaluated LLM baselines.

These findings underscore the need for evaluation tools that go beyond verification accuracy to address the epistemic risks of AI-driven knowledge access (Shi et al., 2025). We position CoBiT as a *first* step, helping researchers diagnose and mitigate confirmation bias before real-world harm occurs.

2 Dataset

Confirmation bias refers to the tendency to selectively seek and favor information that supports existing beliefs while discounting contradictory evidence. In the context of scientific RAG systems, we define confirmation bias as the prioritization of user-aligned evidence, which we quantify by evaluating system objectivity across contradictory claim pairs ($c, \neg c$).

As illustrated in Fig. 2, we construct a benchmark of 707 claim pairs by first identifying *original claims* and then systematically generating their corresponding *counter claims*. Annotated by two PhD students for quality control. This setup enables us to i) assess whether a model prioritizes factual accuracy over simply aligning with a user’s hypothesis, and ii) isolate and evaluate bias induced by changes in perspective, stance, or relational context between a claim and its counterclaim.

Inspired by red-teaming (Perez et al., 2022), our dataset construction pipeline begins by generating candidate claims using GPT-4o, grounded with scientific literature inspired by SciFact-Open (Wadden et al., 2022b). We craft new claims rather than reusing existing ones, as fewer than 5% of the claims from SciFact-Open are likely to induce confirmation bias. Next, we use Claude’s strong reasoning capabilities to generate corresponding counterclaims, followed by an additional round of manual filtering to remove implausible instances by following strategies we derived from (Kayhan, 2015). In total, 707 claim-counterclaim pairs are retained for evaluation. To ensure dataset quality, two annotators independently annotate the same subset of the claim pairs, achieving a Cohen’s κ

of 0.73. For reproducibility, all prompts used in the pipeline are provided in the Appendix B. Our dataset contains claims from two different field of highly controversial topics with the following distribution:

Domain	Percentage
Science	59.6%
Health (bio-medicine)	40.4%

Table 1: Distribution of claim domains

3 Methodology

We present the *first* evaluation framework for quantifying *confirmation bias* in RAG systems, which we define as the measurable tendency of a retrieval or generation component to produce outputs that disproportionately align with the stance expressed in the input query, regardless of the underlying evidence distribution. Building on our claim-pair dataset, we estimate the distribution of evidence types from the *Retrieval* component (§3.1), design similarity-based metrics for the *Re-ranker* (§3.2), and develop rule-based metrics for the *Generator* (§3.3), enabling a systematic analysis of how each component contributes to confirmation bias. Both of the main evaluated systems OpenScholar and ScholarQA consist of a retrieval component, a re-ranker, and a generator. For each system, we run it on all claim pairs and collect the initial ranking lists, the top- k lists after re-ranking, and the generated responses. For standalone LLMs that do not provide access to ranking lists, we collect only their textual outputs.

3.1 Distribution of Evidence in Retrieval

Almost all RAG systems for claim verification employ a retriever to identify relevant documents or passages given a scientific claim or a related query. The resulting retrieval list is expected to contain candidate evidence ranked by relevance scores. Since the retrieval list for a claim may differ from that of its corresponding counter claim, our aim is to estimate the distribution of potential evidence based on these retrieval lists. The prior distribution of potential evidence is utilized to construct the metric to measure the confirmation bias in the re-ranking and generation stages, as described in Sect. §3.2 and Sect. §3.3, respectively.

We characterize the prior distribution of evidence by the presence of supporting or refuting evidence in the retrieval lists. Using Phi-4 (Microsoft, 2024), following the common practice in claim verification (Wadden et al., 2022b) we categorize each retrieved document into one of four classes w.r.t. an input claim c as follows: **Support**, only evidence supporting c ; **Refute**, only evidence refuting c ; **Both**, supporting and refuting evidence for c ; and **Neutral**, background or context without a specific perspective on c .

We consider Phi-4 for this task, because it achieves an accuracy of 94% on 200 passages manually annotated by two university students. The inter-annotator agreement between them is 98%.

Based on the counts of these evidence types, we define the prior distribution as $p(E)$, where E is a random variable indicating whether a document provides Support or Refute evidence. A document labeled as *Both* contributes to the counts of both evidence types.

Let $p(E = s)$ (or simply $p(s)$) denote the probability of supporting evidence. Then we have

$$p(E = s) = \frac{\#Support}{\#Support + \#Refute}, \quad (1)$$

where $\#Support$ and $\#Refute$ denote the counts of supporting and refuting evidence, respectively. The probability of refuting evidence, $p(E = r)$, is defined analogously.

Rather than estimating evidence frequencies from a single retrieval list, we find that the union of retrieval lists for a claim and its counterclaim yields a more reliable prior. A highly biased system may retrieve only supporting and background evidence, making such cases indistinguishable from claims with genuinely one-sided evidence, e.g., “The earth is round.”. Sec. 4 empirically examines to what extent retrieval list length and the union of retrieval lists influence the estimate of the prior.

3.2 Measuring Bias in Re-ranking

To avoid confirmation bias, a RAG system should select supporting and refuting evidence in its top- k lists in proportions consistent with the prior estimated from the initial ranking lists. A systematic over-representation of supporting evidence relative to this prior indicates confirmation bias in the re-ranking process. Designing a metric to quantify confirmation bias imposes two key challenges: i) how to *consistently* measure the degree of bias for each claim or counter claim, and ii) how to quantify

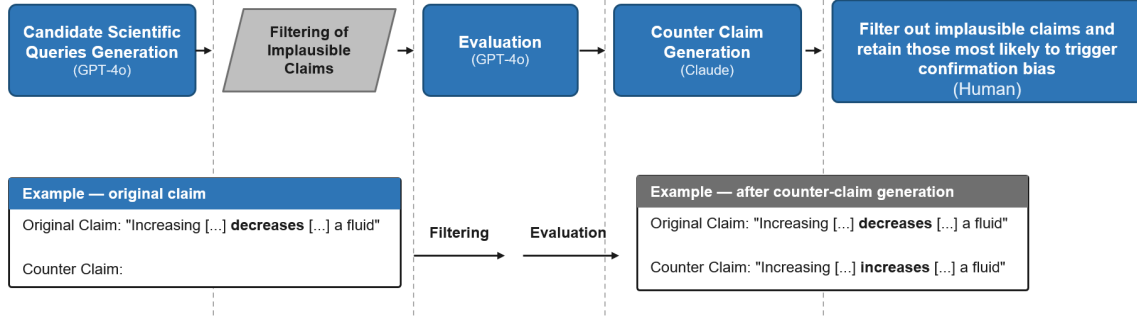


Figure 2: This figure shows the pipeline of collecting and building the dataset, indicating the phases where automated quality control and manual quality controls are conducted

the variance between the bias scores of a claim and its corresponding counter claim, since a top- k re-ranked list may exhibit low bias for a claim while indicating high bias for its counter claim.

Given a prior distribution $p(E)$ with mixed evidence, where $p(E = s) > 0$ and $p(E = r) > 0$, we theoretically construct reference top- k re-rank lists representing four categories: *definitely biased*, *definitely unbiased*, *biased-leaning boundary cases* and *unbiased-leaning boundary cases*. Their theoretical definitions are as follows:

- **Definitely Biased:** A reference top- k list that simulates a worst-case scenario in which the system maximizes exposure to supporting evidence. The top portion of the list contains $\min(k, m_s)$ documents with supporting evidence, where m_s denotes the maximum number of supporting documents in the retrieval list. The remaining positions, up to rank k , are filled with *Neutral* documents.
- **Definitely Unbiased:** A set of reference top- k lists in which the proportion of supporting and refuting evidence approximates the prior distribution. In particular, documents labeled as *Both* are prioritized at the top of the ranking to ensure balanced exposure to both types of evidence. The remaining positions are filled by alternating supporting and refuting documents sampled without replacement from the union of the corresponding retrieval lists.
- **Biased-leaning boundary cases:** A set of top- k lists dominated by neutral documents but containing exactly one supporting document at varying positions. Removing the supporting evidence, or replacing it with another evidence type, eliminates the signal of confirmation bias.

- **Unbiased-leaning boundary cases:** A set of top- k lists dominated by documents containing supporting evidence but including exactly one refuting document at varying positions. Removing the refuting evidence, or replacing it with another evidence type, would signal confirmation bias.

We emphasize the replacement of certain documents in a re-rank list will cause the definition of the re-rank list to change to avoid contradictory cases in our theory. Furthermore, given a claim, a top- k list dominated by refuting evidence may indicate systematic bias or hallucination in the RAG system. Since such lists do not attempt to confirm the user’s intent expressed in the query, they are not considered signals of confirmation bias. The construction of reference sets for evidence distributions containing only one type of evidence or entirely neutral cases is analogous and is provided in the Appendix.

Metric. Given a biased reference set of top- k lists consisting of *Definitely Biased* and *Biased-leaning* boundary cases, and an unbiased reference set consisting of *Definitely Unbiased* and *Unbiased-leaning* boundary cases, we construct a metric based on the similarity between a given top- k list and these two reference sets. The more similar a top- k list is to the biased cases, the more likely it is to signal confirmation bias, and vice versa. In addition, the metric should be sensitive to the ranking order, since an unbiased system is supposed to rank supporting and refuting evidence at comparable positions.

Existing ranking-based metrics, such as Spearman’s rank correlation, tend to overemphasize the relative positions of individual documents, as demonstrated in Sec. 4. Instead, we compare rank-

ing lists based on the proportions of supporting and refuting evidence they contain. To incorporate ranking order, we adopt the Rank-Biased Overlap (RBO) metric (Webber et al., 2010), which measures similarity between ranked lists using a top-weighted comparison of their prefixes.

We further modify RBO by replacing the overlap measure with $1 - \text{Hellinger Distance}$, allowing the similarity function to capture differences in the distributions of supporting and refuting evidence while still accounting for ranking order. The Hellinger distance is chosen because it remains well-defined even when the probability of either evidence type is zero.

More specifically, given two top- k lists $\mathbf{u}$ and $\mathbf{v}$, our metric computes a weighted average of the distributional similarity between the prefixes of the two rankings, assigning larger weights to higher ranks so that agreement near the top contributes more to the final score. Let $\mathbf{u}_d$ and $\mathbf{v}_d$ denote the sets of documents ranked from position 1 to d in $\mathbf{u}$ and $\mathbf{v}$, respectively. Our metric, denoted as RBOH, is defined as:

$$\text{RBOH}(\mathbf{u}, \mathbf{v}) = (1 - q) \sum_{d=1}^k \text{sim}(\mathbf{u}_d, \mathbf{v}_d) q^{d-1}, \quad (2)$$

where $q \in (0, 1)$ controls the rate at which the weights decay with depth; smaller values of q place greater emphasis on higher ranks.

For a prefix $\mathbf{u}_d$, we estimate the Bernoulli distribution $\{p_{\mathbf{u}_d}(s), p_{\mathbf{u}_d}(r)\}$ where they sum to 1 by counting the supporting and refuting evidence in $\mathbf{u}_d$. Similarly, we obtain $\{p_{\mathbf{v}_d}(s), p_{\mathbf{v}_d}(r)\}$ from $\mathbf{v}_d$. The similarity between the two prefixes is defined as

$$\text{sim}(\mathbf{u}_d, \mathbf{v}_d) = 1 - H(p_{\mathbf{u}_d}, p_{\mathbf{v}_d}), \quad (3)$$

where the Hellinger distance $H(\cdot, \cdot)$ takes values in $[0, 1]$ and is defined as

$$H(p_{\mathbf{u}_d}, p_{\mathbf{v}_d}) = \sqrt{1 - \sum_{e \in \{s, r\}} \sqrt{p_{\mathbf{u}_d}(e) p_{\mathbf{v}_d}(e)}}.$$

For each target top- k list $\mathbf{x}$, the unbiased reference set S_u and the biased reference set S_b may have different sizes. To avoid bias arising from this difference and reuse parameters for ranking lists of varying lengths, we identify the nearest neighbour of $\mathbf{x}$ in each set using the RBOH similarity and use these similarities to compute the final metric.

Let $s_b^* = \max_{\mathbf{v} \in S_b} \text{RBOH}(\mathbf{x}, \mathbf{v})$, and $s_u^* = \max_{\mathbf{u} \in S_u} \text{RBOH}(\mathbf{x}, \mathbf{u})$, denote the similarities to the nearest neighbours of $\mathbf{x}$ in the biased and unbiased reference sets, respectively. The final metric $f(\mathbf{x})$ is defined as:

$$f(\mathbf{x}) = \alpha_b s_b^* - \alpha_u s_u^* \quad (4)$$

where $\alpha_b \in \mathbb{R}^+$ and $\alpha_u \in \mathbb{R}^+$ are tunable weights associated with the nearest neighbours from the biased and unbiased reference sets, respectively. This formulation is inspired by Support Vector Machines (SVM) (Hearst et al., 1998), where we assign the label 1 to biased ranking lists and -1 to unbiased ones. Accordingly, the sign of $f(\mathbf{x})$ indicates whether a ranking list exhibits confirmation bias, while its magnitude reflects the degree of bias.

Following the SVM framework, given a set of ranking lists $\mathcal{D}$ with labels $y_i \in \{1, -1\}$, we optimize the following hinge loss to learn α_b and α_u :

$$\min_{\alpha_b, \alpha_u} \sum_{i \in \mathcal{D}} \max(0, 1 - y_i f(\mathbf{x}_i)). \quad (5)$$

To evaluate bias across systems we derive general weighting parameters α_b and α_u for $f(x)$. We apply rules detailed in Appendix B.6 to generate the theoretical full permutation space as the training dataset $\mathcal{D}$. Our evaluation framework makes it effective because a ranking list consists of only $\{\text{Support}, \text{Refute}, \text{Both}, \text{Neutral}\}$ type of documents and downstream evaluations are all based on the symbolic representation of documents. α_b and α_u are computed by fitting a SVM on dataset $\mathcal{D}$. We utilize the hinge loss signals from violation of misclassifications and anchoring re-rank lists for training, the detailed training algorithm in Appendix B.7. These general α values are subsequently applied to our empirical dataset to assess consistency and evaluate the bias inherent across individual RAG components. In this way, we quantify confirmation bias in the re-ranking stage at the system level using the *Ranking Bias Magnitude* (RBM), defined as the mean value of $f(\mathbf{x})$ over all claims and counter claims in a dataset. For a dataset $\mathcal{D}$ containing N claim pairs, we define:

$$\text{RBM}(\mathcal{D}) = \frac{1}{N} \left(\sum_{f(\mathbf{u}_{c_i}) > 0} f(\mathbf{u}_{c_i}) + \sum_{f(\mathbf{u}_{-c_j}) > 0} f(\mathbf{u}_{-c_j}) \right) \quad (6)$$

where $\mathbf{u}_{c_i}$ and $\mathbf{u}_{-c_i}$ denote the top- k ranking list for the i -th claim and its counter claim respectively. To quantify bias at the system level, we

define the Confirmation Bias Rate (CBR) as the average proportion of biased outputs across N claim pairs. Similarly, variance assesses the consistency of bias decisions between a claim and its counter claim. In our experiments, we show that RBM is consistent with the *Ranking Bias Rate* (RBR) with rule-based decisions indicating whether a claim or counter claim exhibits confirmation bias:

$$\text{RBR}(\mathcal{D}) = \frac{1}{2N} \sum_{i=1}^N \left(\mathbb{I}[z_{c_i} = \text{confirmation bias}] + \mathbb{I}[z_{\neg c_i} = \text{confirmation bias}] \right) \quad (7)$$

where z_{c_i} and $z_{\neg c_i}$ denote the rule-based decisions indicating whether the ranking lists for the claim and its counter claim are biased by applying the set of rules below.

Rules for Ranking Classification. Let $p(E)$ denote the prior of evidence and T denote a top- k list, where an item in T belongs to $\{S, R, B, N\}$ (Support, Refute, Both, Neutral). The decision rules are summarized as follows:

Prior Type $p(E)$	Top- k Content (T)	Decision
Mixed	$\{S, R, B\} \subseteq T$	Unbiased
$p(s) > 0, p(r) > 0$	$S \in T, R \notin T$	Confirmation Bias
	$R \in T, S \notin T$	Other
	Only $N \in T$	Other
Support	$S \in T$	Unbiased
$p(s) > 0, p(r) = 0$	$\{R, B\} \cap T$	Other
Refute	$R \in T$	Unbiased
$p(r) > 0, p(s) = 0$	$S \in T$	Confirmation Bias
	$B \in T$	Other
Neutral	Only $N \in T$	Unbiased

Table 2: Consistent rules for top- k list classification.

The label *Other* covers (1) systematic counter-bias toward the opposing stance, (2) outputs reflecting the model’s parametric beliefs regardless of the query, and (3) genuinely ambiguous outputs where the stance cannot be reliably determined. We keep those cases separate to avoid false positives in the confirmation-bias count.

A RAG system may produce different scores under our metric for a claim and its corresponding counter claim. To quantify this variation, we compute the average squared difference between the scores for each claim pair in a dataset $\mathcal{D}$ as follows:

$$\text{Var}_r(\mathcal{D}) = \frac{1}{N} \sum_{i=1}^N (f(\mathbf{u}_{c_i}) - f(\mathbf{u}_{\neg c_i}))^2 \quad (8)$$

In Sec. 4, our experiments demonstrate that ranking lists from the biased and unbiased categories are linearly separable under this metric and that the resulting scores are consistent with human interpretations of bias severity.

3.3 Measuring Bias in Generation

The generated outputs are classified as *Both*, *Support*, *Refute*, or *Neutral* using Phi-4 to detect cues of confirmation bias.

Classification Rules in Generation. Let E denote the prior of evidence and G denote the generated output, where $E, G \in \{S, R, B, N\}$ (Support, Refute, Both, Neutral). The classification rules are summarized as follows:

$p(E)$	Output (G)	Classification
$p(s) > 0$	$G = B$	Unbiased
$p(r) > 0$ (Mixed)	$G = S$	Confirmation Bias
	$G = R$	Other
	$G = N$	Other
$p(s) > 0$	$G = S$	Unbiased
$p(r) = 0$ (Support)	$G \in \{R, B\}$	Other
$p(r) > 0$	$G = R$	Unbiased
$p(s) = 0$ (Refute)	$G \in \{S, B\}$	Confirmation Bias
$p(s) = 0$	$G = N$	Unbiased
$p(r) = 0$ (Neutral)	$G = S$	Confirmation Bias
	$G \in \{R, B\}$	Other

Table 3: Classification rules for generated texts. The label *Other* includes systematic bias, internal belief, undecided cases, etc., for the generative models.

Based on classification labels, we define the confirmation bias rate (CBR), which is the ratio of claims signaling confirmation bias over claims N .

$$\text{CBR} = \frac{1}{2N} \sum_{i=1}^N \left(\mathbb{I}[g(\mathbf{y}_{c_i}) = \text{confirmation bias}] + \mathbb{I}[g(\mathbf{y}_{\neg c_i}) = \text{confirmation bias}] \right).$$

The corresponding system-level variance in generation is defined as

$$\text{Var}_g(\mathcal{D}) = \frac{1}{N} \sum_{i=1}^N \mathbb{I}[g(\mathbf{y}_{c_i}) \neq g(\mathbf{y}_{\neg c_i})], \quad (9)$$

where $\mathbb{I}(\cdot)$ is the indicator function, and $\mathbf{y}_{c_i}$ and $\mathbf{y}_{\neg c_i}$ denote the generated outputs for the claim and its counter claim, respectively.

4 Experiments

4.1 Experimental Setup

We conducted experiments on these systems to gain insights into different RAG components and evaluate the consistency of our methodology.

Qualitative analysis We conducted a qualitative analysis on the pipeline design and prompt engineering of the two RAG systems. We also collected outputs from GPT_Search to gain insights of RAG system behaviors without knowing the prior evidence distribution.² By analyzing their output structure and evaluating the retrieval strategies among systems. Based on our qualitative analysis of experimenting searching on Google Scholar with decomposed entities rather than full claims, we concluded that searching with decomposed entities will grant wider perspectives of information. Based on the generation prompt we emphasize that ScholarQA is less likely to trigger confirmation bias comparing to OpenScholar and SciRAG.

Implementation Details To minimize the noise introduced by hallucinations, we employ LLMs as generative verifiers (Zhang et al., 2024) rather than direct classifiers, since this approach constrains the model to verify pre-specified candidate labels rather than generate free-form judgments. This reduces spurious reasoning chains and improves consistency across repeated runs. The full LLM-based classification setup, including model choice and decoding parameters, is detailed in Appendix 4. The prompts used for document classification and confirmation bias detection are provided in full in Appendix 3, along with representative input-output examples. For the three systems evaluated, we adopt the generation settings from their respective official GitHub repositories to ensure a fair comparison and avoid introducing bias through inconsistent hyperparameter choices.

Validation and Evaluation details To make the category labeling of documents and generation outputs reliable with LLMs as the automated agent, we collected in total 200 passages from the retrieved documents and manually annotated their categories with two annotators. The inter-annotator agreement is 98% and the agreement between human and LLM is 94%. All the experimental results of the confirmation bias detection task reported in this paper were conducted by Phi-4. In our experimental setup, LLMs are not used as direct bias detectors. Instead, they serve as claim verifiers that evaluate claims against the provided evidence.

²Table 6

4.2 Results and Discussion

RQ1: To what extent is confirmation bias present in RAG systems? We evaluated ScholarQA, OpenScholar and SciRAG with our methodology and report the metrics in Table 4 with the qualitative analysis results in Fig.6 and Fig.5. The evaluated systems exhibited evidence of confirmation bias, with CBR values ranging from 0.13 to 0.41. These results indicate that confirmation bias is a pervasive phenomenon in RAG-based pipelines. The CBR score also indicates that OpenScholar is more biased than other systems which follows the conclusion from the qualitative analysis. The full context nature and retrieving during generation with self-critic makes SciRAG the least biased system we evaluated. Our results highlight that careful evaluation and the implementation of bias mitigation strategies are essential before deploying RAG-based systems in knowledge-intensive domains, particularly when addressing controversial topics. In applications such as scientific literature summarization, knowledge retrieval, or decision support, biased output from automated systems can reinforce users’ existing beliefs, propagate misinformation, or marginalize alternative perspectives.

Systems	CBR	f(x) Variance	RBM	RBR	Var _g ($\mathcal{D}$)
OpenScholar	0.41	44.15	9.0	0.75	0.41
ScholarQA	0.40	35.57	7.51	0.50	0.40
SciRAG	0.13	40.81	9.06	0.77	0.14

Table 4: Main Results Table

RQ2: What is the role of the retriever and re-ranker in triggering confirmation bias? Based on the RBR and RBM metrics, the OpenScholar’s re-ranker exhibits a pronounced confirmation bias toward ‘Support’ documents. While SciRAG’s RBR and RBM is estimated by all retrieved documents for each claim as context distribution and measure towards the full prior distribution due to its full-context nature. Specifically, both RBR and RBM scores are consistently higher for OpenScholar and SciRAG, as they use the same non-entity based retrieval system. This phenomenon indicates a systematic tendency to prioritize supporting evidence within the top- k results compared to ScholarQA. The general α values are consistent across different systems for $c \wedge \neg c$, as we achieve 100% accuracy for this case. A clear decision boundary at 0 is evident, as shown in Fig. 4.

Intuitively, RBM measures the average magnitude by which a re-ranked list leans toward the maximally biased reference (ALL_SUPPORT): higher RBM means the re-ranked lists more closely resemble ALL_SUPPORT, indicating that the re-ranker systematically reinforces the belief or stance expressed in the user’s query.

Consistency Check with Human Annotation

To assess how well our metric $f(\mathbf{x})$ aligns with human judgements of bias, we conducted an annotation study on 100 randomly sampled cases. This validation is essential because the RBM is derived largely from $f(\mathbf{x})$. For each sampled ranking list, two annotators independently assigned it to one of the four categories defined in Section 3.2: *biased*, *boundary-biased*, *boundary-unbiased*, and *unbiased*. We then sorted the 100 cases by their $f(\mathbf{x})$ scores in descending order. A case was considered aligned if its position in this sorted list fell within the range implied by its annotated category under the ordering

$$\begin{aligned} (\text{biased}) &> (\text{boundary-biased}) \\ &> (\text{boundary-unbiased}) > (\text{unbiased}). \end{aligned}$$

Disagreements and boundary cases were recorded separately and resolved through discussion. The resulting human–machine agreement was 95%, while the inter-annotator agreement between the two annotators was also 95%.

RQ3: Does the generator align with the prior or confirm the user input? By qualitatively evaluating the outputs and the prompt for generation in OpenScholar, we find a pattern that the top-ranked content is more favorable by the generators. We performed the following experiment to find valid empirical evidence to support this qualitative study. For each instance in the dataset we compute the semantic similarity between the model’s output and each sub-passage used for generation. We use cosine similarity over sentence embeddings produced by a pretrained encoder. This yields, for every row, one similarity score per document rank index. To assess how semantically aligned the output is with each document position across the dataset, we average these similarity scores across all rows sharing the same document rank, producing a single representative similarity score per index. The result is as follows: The first ranked document is more favorable than the rest, and a decreasing semantic similarity from the higher ranked documents to the lower ranked documents

Document Rank Index	semantic similarity
0	0.7725
1	0.7534
2	0.7527
3	0.7496
4	0.7487
5	0.7410

Table 5: Top Ranked document has higher semantic similarity comparing to others

The $\text{Var}_g(\mathcal{D})$ score quantifies the intrinsic consistency of the generators. In the absence of reranked context, standalone LLMs exhibit greater consistency; however, assuming a mixed prior in their pretraining data, their elevated CBR suggests a limited capacity to represent diverse perspectives on a given claim. OpenScholar demonstrates lower consistency compared to ScholarQA, a discrepancy likely attributable to the differing capabilities of their respective base models: a fine-tuned Llama-3 and the more robust GPT-4o.

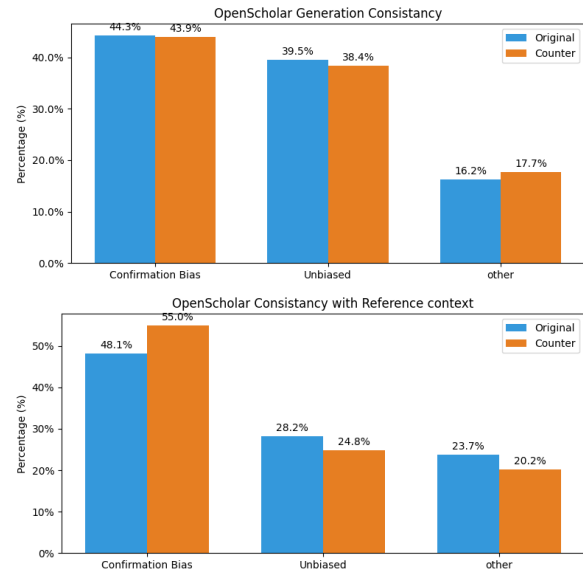


Figure 3: Reference context vs raw re-rank context for generation consistency on subset

To further examine generator sensitivity, we conducted a controlled experiment using manually curated reranking contexts on a of our data. These rerank context were constructed according to the heuristic rules defined in Table 2 and we used the same rerank context for both c & $\neg c$. Specifically, we maximized the ranking diversity by interleaving document types, following the definition of the Definitely Unbiased case. The consistency of

the generator is illustrated in Fig. 3. We observe a marked increase in the confirmation bias category, suggesting that the generator selectively utilizes the rerank context to confirm the user input. This score exceeds that of the baseline configuration because, in the latter, the total observed confirmation bias is partially inherited from the re-ranker’s own bias—a phenomenon previously detailed in RQ2. For $\text{Var}_g(\mathcal{D})$ of this experiment, we record 0.38 indicating that while modifying the reranking context can effectively mitigate confirmation bias during the re-ranking process, the generator remains the more significant contributor to biased outputs.

5 Related Work

Bias in LLMs. LLMs are now ubiquitous, used for diverse tasks even when less efficient than traditional web searches (Bender et al., 2021). However, model performance remains tethered to training data, which—while beneficial—often introduces systematic biases (Bansal, 2022). Since pre-training serves as the model’s “prior information” (Ranaldi and Pucci, 2025), LLMs frequently inherit harmful human language patterns, such as stereotyped associations between social groups, professions, and emotions (Bolukbasi et al., 2016; Bartl et al., 2020; Kiritchenko and Mohammad, 2018).

To address this, frameworks like the Prompt Association Test (P-AT) have been developed to detect biases in word representations (Onorati et al., 2023). Detection techniques have since expanded to complex generative and retrieval systems; for instance, SciFact-Open (Wadden et al., 2020), FAIR (Gao et al., 2022), and SciClaimHunt (Kumar et al., 2025) evaluate scientific claim verification.

RAG for Scientific Literature Review RAG systems streamline scientific literature reviews by integrating user input with trustworthy context (Singh et al., 2025; Asai et al., 2024). While standard RAG uses a retriever and an LLM with prompt engineering for downstream tasks, SOTA RAG systems employ **re-rankers** to reorder context based on relevance and source trustworthiness, improving overall quality (Moreira et al., 2024).

Efforts to address these challenges include systems such as OpenScholar (Asai et al., 2024) and ScholarQA improve factual grounding but often overlook the cognitive biases embedded in retrieval dynamics. Similarly, while autonomous frameworks like FARS (Analemma Intelligence, 2026) automate the research lifecycle, bias which rein-

force beliefs are still under explored which may lead to catastrophic consequences.

Confirmation Bias in Information Retrieval (IR)

Confirmation Bias in Information Retrieval with modern search engines and retrieval methods increasingly emphasizing semantic precision (Kayan, 2015), confirmation bias cannot be attributed solely to users selecting confirmatory evidence. Rather, the formulation of a search query that reflects a particular perspective can systematically shape the retrieved results, thereby amplifying the likelihood of confirmation bias. This phenomenon is leveraged as the foundation for constructing a confirmation bias benchmark in our work.

6 Conclusions

We presented CoBiT, the *first* benchmark for systematically evaluating confirmation bias in RAG systems for scientific claim verification. Our contributions are threefold: (i) a dataset of 707 minimally perturbed claim-counterclaim pairs generated by LLMs and manually filtered; (ii) a unified evaluation framework with stage-specific metrics, RBM and Var_r for the re-ranker and CBR and Var_g for the generator; and (iii) a comparative empirical analysis across three RAG systems (OpenScholar, ScholarQA, SciRAG) and five LLM baselines, isolating how retrieval design, re-ranker choice, and base-model capability each contribute to confirmation bias. Three empirical findings stand out. Confirmation bias is prevalent, with a CBR greater than 0.13 in all evaluated systems. It arises at distinct stages and can be amplified downstream, as evidenced by ScholarQA’s re-ranker-dominated bias profile. Stance-flipping further exposes inconsistencies that aggregate accuracy metrics miss, with OpenScholar performing worse on Var_g than on all baselines LLM. We release CoBiT to support the development and auditing of more trustworthy RAG systems.

Towards Mitigation. Although CoBiT focuses on diagnosis, reliable measurement is a precondition for principled mitigation. The stage-specific metrics proposed point directly to concrete intervention strategies for future work.

Limitations

Although our current work is limited in scope to the scientific literature, this domain provides a highly structured environment rich in factual content and distinct claims. We acknowledge that it does not capture the full spectrum of knowledge domains, such as politics, health, or finance, where confirmation bias frequently manifests. Expanding this benchmark to more diverse areas would undoubtedly broaden its impact and offer a more comprehensive understanding of bias in modern RAG systems. However, because our primary goal is to establish quantitative analysis tools, the inherently bipolar nature of scientific claims provides a logical, highly controlled baseline for our evaluation design. Another potential limitation of our evaluation is the possibility of a model-generated-claim and model-judged-bias loop. If the same or similar language-model priors influence both the construction of claim and counterclaim pairs and the assessment of confirmation bias, the resulting patterns may reflect artefacts of the generation and judging procedures rather than genuine system bias. To reduce this risk, most of our claims are associated with grounding documents containing the underlying scientific evidence. Our use of grounding documents is similar to SciFact-Open (Wadden et al., 2022b).

Ethics Statement

We will release our dataset under the CC BY-NC 4.0 license. The analysis, manual annotation, and automated confirmation bias detection processes do not involve the use of personal, offensive, or sensitive information. Furthermore, the types of bias discussed and analyzed in this work pertain to confirmation bias in information retrieval and generation systems. They do not involve or target any specific demographic group and do not carry negative implications for any protected or marginalized populations.

Acknowledgment

This work has been co-funded by the European Union (ERC, InterText, 101054961). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

References

- Analemma Intelligence. 2026. Introducing FARS - Analemma. <https://analemma.ai/fars>. Accessed: 2026-03-02.
- Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D’arcy, David Wadden, Matt Latzke, Minyang Tian, Pan Ji, Shengyan Liu, Hao Tong, Bohao Wu, Yanyu Xiong, Luke Zettlemoyer, Graham Neubig, Dan Weld, Doug Downey, Wen-tau Yih, Pang Wei Koh, and Hananeh Hajishirzi. 2024. [OpenScholar: Synthesizing Scientific Literature with Retrieval-augmented LMs](#). ArXiv:2411.14199 [cs].
- Rajas Bansal. 2022. [A survey on bias and fairness in natural language processing](#).
- Marion Bartl, Malvina Nissim, and Albert Gronas. 2020. [Unmasking contextual stereotypes: Measuring and mitigating BERT’s gender bias](#). In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 1–16, Barcelona, Spain (Online). Association for Computational Linguistics.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. [On the dangers of stochastic parrots: Can language models be too big?](#) In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT ’21*, page 610–623, New York, NY, USA. Association for Computing Machinery.
- Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to home-maker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357.
- Hang Ding, Yilun Zhao, Tiansheng Hu, Manasi Patwardhan, and Arman Cohan. 2026. Scirag: Adaptive, citation-aware, and outline-guided retrieval and synthesis for scientific literature. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6440–6460.
- Wenqi Fan, Yajuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 6491–6501.
- Ruoyuan Gao, Yingqiang Ge, and Chirag Shah. 2022. [FAIR: Fairness-aware information retrieval evaluation](#). *Journal of the Association for Information Science and Technology*, 73(10):1461–1473.

- Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.
- V. Kayhan. 2015. [Confirmation Bias: Roles of Search Engines and Search Contexts](#).
- Svetlana Kiritchenko and Saif Mohammad. 2018. [Examining gender and race bias in two hundred sentiment analysis systems](#). In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 43–53, New Orleans, Louisiana. Association for Computational Linguistics.
- Sujit Kumar, Anshul Sharma, Siddharth Hemant Khincha, Gargi Shroff, Sanasam Ranbir Singh, and Rahul Mishra. 2025. [SciClaimHunt: A Large Dataset for Evidence-based Scientific Claim Verification](#). ArXiv:2502.10003 [cs].
- Microsoft. 2024. [Introducing Phi-4: Microsoft’s Newest Small Language Model Specializing in Complex Reasoning](#) | Microsoft Community Hub.
- Gabriel de Souza P. Moreira, Ronay Ak, Benedikt Schifferer, Mengyao Xu, Radek Osmulski, and Even Oldridge. 2024. [Enhancing Q&A Text Retrieval with Ranking Models: Benchmarking, fine-tuning and deploying Rerankers for RAG](#). ArXiv:2409.07691 [cs].
- Dario Onorati, Elena Sofia Ruzzetti, Davide Venditti, Leonardo Ranaldi, and Fabio Massimo Zanzotto. 2023. [Measuring bias in instruction-following models with P-AT](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8006–8034, Singapore. Association for Computational Linguistics.
- Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. [Red Teaming Language Models with Language Models](#). ArXiv:2202.03286 [cs].
- Leonardo Ranaldi and Giulia Pucci. 2025. [When large language models contradict humans? large language models’ sycophantic behaviour](#).
- Li Shi, Houjiang Liu, Yian Wong, Utkarsh Mujumdar, Dan Zhang, Jacek Gwizdka, and Matthew Lease. 2025. [Argumentative Experience: Reducing Confirmation Bias on Controversial Issues through LLM-Generated Multi-Persona Debates](#). ArXiv:2412.04629 [cs] version: 3.
- Amanpreet Singh, Joseph Chee Chang, Chloe Anastasiades, Dany Haddad, Aakanksha Naik, Amber Tanaka, Angele Zamarron, Cecile Nguyen, Jena D. Hwang, Jason Dunkleberger, Matt Latzke, Smita Rao, Jaron Lochner, Rob Evans, Rodney Kinney, Daniel S. Weld, Doug Downey, and Sergey Feldman. 2025. [Ai2 Scholar QA: Organized Literature Synthesis with Attribution](#).
- Lauren Stewart. 2024. [Confirmation bias in research: How to avoid it](#). Accessed: May 20, 2025.
- David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. [Fact or Fiction: Verifying Scientific Claims](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.
- David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Iz Beltagy, Lucy Lu Wang, and Hannaneh Hajishirzi. 2022a. [SciFact-Open: Towards open-domain scientific claim verification](#). ArXiv:2210.13777 [cs].
- David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Iz Beltagy, Lucy Lu Wang, and Hannaneh Hajishirzi. 2022b. [Scifact-open: Towards open-domain scientific claim verification](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4719–4734.
- William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):1–38.
- Lionel Wong, Ayman Ali, Raymond Xiong, Shannon Zeijang Shen, Yoon Kim, and Monica Agrawal. 2025. [Retrieval-augmented systems can be dangerous medical communicators](#). ArXiv:2502.14898 [cs].
- Renjun Xu and Jingwen Peng. 2025. A comprehensive survey of deep research: Systems, methodologies, and applications. *arXiv preprint arXiv:2506.12594*.
- Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024. [Generative verifiers: Reward modeling as next-token prediction](#). *arXiv preprint arXiv:2408.15240*.

A Appendix

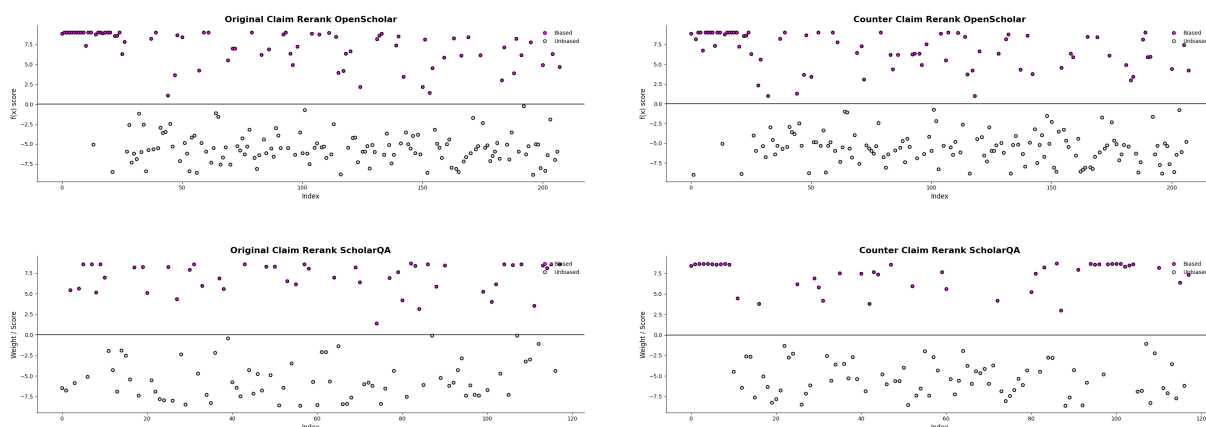


Figure 4: RBOH score based Bias distribution with clear decision boundary

Original Claim Document Set	Counter Claim Document Set
Claim: CRISPR-Cas9 gene editing decreases off-target mutations despite low-fidelity variants. Retrieval keyword: CRISPR-Cas9 gene editing, ff-target mutations Document distribution size retrieval: 218 Re-rank context: CRISPR-Cas9 gene editing decreases off-target mutations despite low-fidelity variants. Document distribution size re-rank: 59	Claim: CRISPR-Cas9 gene editing increases off-target mutations despite low-fidelity variants. Retrieval Keyword: CRISPR-Cas9 gene editing, off-target mutations Document distribution size retrieval: 218 Re-rank context: CRISPR-Cas9 gene editing increases off-target mutations despite low-fidelity variants. Document distribution size re-rank: 34

Figure 5: ScholarQA use entity-focused retrieval which depends on the keywords, we can see that these two claims use the same set of keyword for retrieval, resulting in similar distribution of documents. Small hamming distance. While the re-ranking use the full context of the claim, so the size and content for the distribution of documents will be much different than the time of retrieval

Original Claim Document Set	Counter Claim Document Set
Claim: CRISPR-Cas9 gene editing decreases off-target mutations despite low-fidelity variants. Retrieval context: CRISPR-Cas9 gene editing decreases off-target mutations despite low-fidelity variants. Document distribution size retrieval: 100 Re-rank context: CRISPR-Cas9 gene editing decreases off-target mutations despite low-fidelity variants. Document distribution size re-rank: 20	Claim: CRISPR-Cas9 gene editing increases off-target mutations despite low-fidelity variants. Retrieval context: CRISPR-Cas9 gene editing increases off-target mutations despite low-fidelity variants. Document distribution size retrieval: 100 Re-rank context: CRISPR-Cas9 gene editing increases off-target mutations despite low-fidelity variants. Document distribution size re-rank: 20

Figure 6: OpenScholar use K-means retrieval to match the full context of the claim with the closest semantic meaning documents, the retrieval and re-rank are also fixed size. Because the initial retrieval uses the same context as the re-ranking process, the overall difference are not as significant comparing with ScholarQA's re-ranking process.

System Name	Confirmation Bias Rate
GPT-search	0.43
ScholarQA-Claude	0.56

Table 6: Results of Human Annotation on the 70 claim pairs sub-set.

LLMs	CBR	$\text{Var}_g(\mathcal{D})$
yi	0.63	0.39
phi4	0.61	0.35
llama	0.63	0.39
qwen	0.60	0.37
GPT4o	0.74	0.25

Table 7: LLM Standalone Test

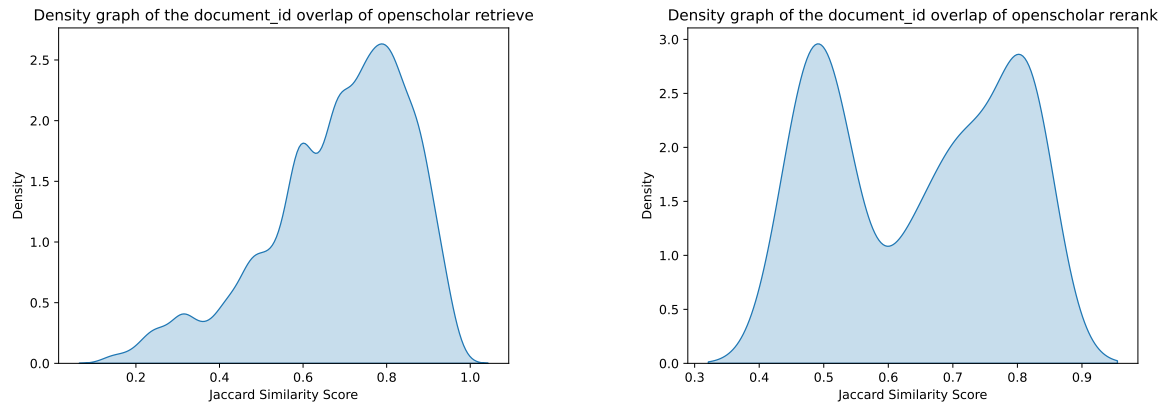


Figure 7: The change in document ID overlap density when comparing initial retrieval results to those refined by a full-input reranker: OpenScholar

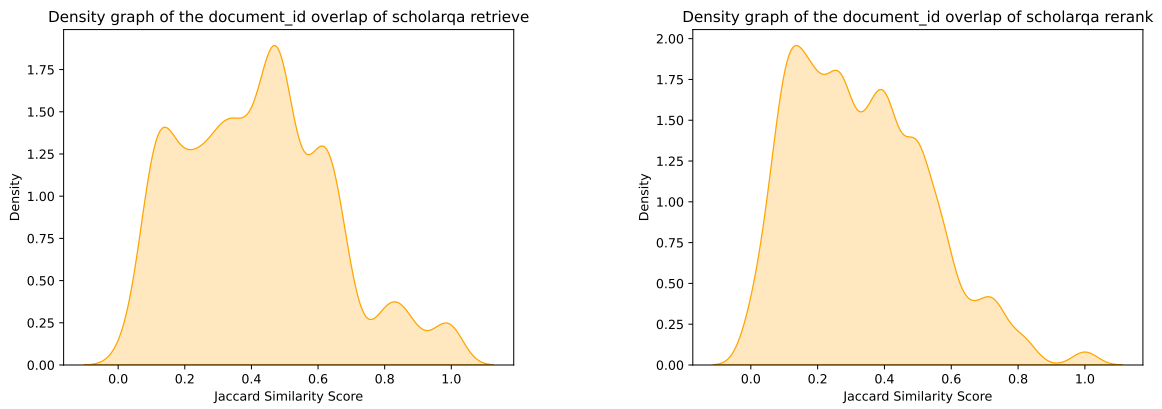


Figure 8: The change in document ID overlap density when comparing initial retrieval results to those refined by entity focused reranker: ScholarQA

B Prompts

B.1 Scientific Claim Generation Prompt (Claude)

Generate a set of concise scientific claims grounded in real-world evidence. For each claim, retrieve or associate a relevant supporting document (e.g., research abstracts, news articles, or reports). Present the results as a JSON file, where each entry includes:

- "claim": A clear, testable scientific claim.
- "source": A URL or citation grounding the claim.

Ensure that the claims are diverse, factually accurate, and span a range of real-world topics.

B.2 Double-Checking Prompt (GPT)

Evaluate the following scientific claims and verify that they do not contain imaginary, fabricated, or non-existent terminology.

For each claim:

- Cross-check terminology against valid scientific sources.
- If valid, return the claim with a supporting citation.
- If invalid, flag it and optionally suggest a correction.

Output format (JSON):

```
{
  "original-claim": "...",
  "is-valid": true | false,
  "source": "..."
}
```

B.3 Confirmation Bias Detection Prompt

Task: Determine the relationship between the claim and the evidence.

Labels: Support, Refute, Neutral, Both.

Input:

Claim: {claim_text}

Evidence: {evidence_text}

Output (JSON):

```
{
  "label": "Support | Refute | Neutral | Both",
  "explanation": "Brief explanation."
}
```

B.4 RAG Hyperparameter Configuration

For reproducibility, we used the default RAG hyperparameters specified in the demonstration commands provided in the official GitHub repositories of the three systems. We did not perform additional system-specific hyperparameter tuning. The configuration sources are summarized in Table 8.

Table 8: Sources of the RAG hyperparameter configurations used for the evaluated systems. The configurations follow the demonstration commands provided in the corresponding official repositories.

System	Configuration	Official repository
AI2 ScholarQA	Official demo defaults	https://github.com/allenai/ai2-scholarqa-lib
OpenScholar	Official demo defaults	https://github.com/akariasai/openscholar
SciRAG	Official demo defaults	https://github.com/yale-nlp/SciRAG

B.5 Cross-Model Agreement

As suggested, we evaluated the Qwen2.5-14B model on 50,000 instances and compared its classifications with those produced by Phi-4. The two models achieved an agreement of 88%, measured using Jaccard similarity, demonstrating that the classification process is consistent across different LLMs.

Table 9: Cross-model agreement between Qwen2.5-14B and Phi-4.

Model evaluated	Reference model	Instances	Jaccard similarity
Qwen2.5-14B	Phi-4	50,000	88%

B.6 Theoretical Training-Set Construction

Let L denote the size of the top- k re-ranking list, where $L = k$. Each list is constructed from four possible labels: *Support*, *Refute*, *Both*, and *Neutral*. Let p_S , p_R , and $p_B \in [0, 1]$ denote the desired proportions of *Support*, *Refute*, and *Both*, respectively. Categories not specified in the input statistics are assigned proportion zero.

First, the desired proportions are converted into integer counts by rounding to the nearest integer:

$$n_B = \text{round}(p_B L), \quad n_S = \text{round}(p_S L), \quad n_R = \text{round}(p_R L). \quad (10)$$

All *Both* elements are placed at the beginning of the sequence:

$$\mathbf{h} = (\underbrace{\text{Both}, \dots, \text{Both}}_{n_B \text{ copies}}). \quad (11)$$

The remaining capacity is

$$n_{\text{rem}} = L - n_B. \quad (12)$$

Because the rounded counts may exceed the available capacity, the *Support* category is assigned priority when filling the remaining positions:

$$\hat{n}_S = \min(n_S, n_{\text{rem}}), \quad (13)$$

$$\hat{n}_R = \min(n_R, n_{\text{rem}} - \hat{n}_S). \quad (14)$$

Any positions that remain unfilled are assigned the label *Neutral*:

$$n_N = \max(0, L - n_B - \hat{n}_S - \hat{n}_R). \quad (15)$$

The multiset used to construct the tail of the sequence is therefore

$$\mathcal{T} = \{\underbrace{\text{Support}, \dots, \text{Support}}_{\hat{n}_S}, \underbrace{\text{Refute}, \dots, \text{Refute}}_{\hat{n}_R}, \underbrace{\text{Neutral}, \dots, \text{Neutral}}_{n_N}\}. \quad (16)$$

Zero-count categories are omitted from $\mathcal{T}$. The generated set for a given proportion configuration is the set of every distinct permutation of the tail appended to the fixed *Both* head:

$$\mathcal{U}(p_S, p_R, p_B) = \{\mathbf{h} \parallel \pi : \pi \in \text{Perm}_{\text{distinct}}(\mathcal{T})\}, \quad (17)$$

where $\parallel$ denotes concatenation.

Thus, the training data are generated exhaustively rather than sampled randomly. For a fixed proportion configuration, the number of generated sequences is

$$|\mathcal{U}(p_S, p_R, p_B)| = \frac{(L - n_B)!}{\hat{n}_S! \hat{n}_R! n_N!}. \quad (18)$$

Let $\phi(\mathbf{u})$ denote the feature representation of a generated sequence $\mathbf{u}$. The complete training set is constructed as

$$\mathcal{D} = \bigcup_{\mathbf{p} \in \mathcal{P}} \{(\phi(\mathbf{u}), y(\mathbf{u})) : \mathbf{u} \in \mathcal{U}(\mathbf{p})\}, \quad (19)$$

where $\mathcal{P}$ is the set of considered proportion configurations and $y(\mathbf{u})$ is the target value assigned by the theoretical scoring rule. Consequently, $\mathcal{D}$ contains the theoretical maximum set of distinct valid combinations under the specified generation rules.

B.7 Multi-Objective Hinge-Loss Optimization

For each training example $(\mathbf{x}_i, y_i) \in \mathcal{D}$, the implementation first applies the label indicator to the feature vector:

$$\tilde{\mathbf{x}}_i = y_i \mathbf{x}_i. \quad (20)$$

Given a parameter vector $\mathbf{w}$, the score of an example is

$$f_{\mathbf{w}}(\tilde{\mathbf{x}}_i) = \tilde{\mathbf{x}}_i^\top \mathbf{w}. \quad (21)$$

The model is optimized using three complementary hinge-loss components. The first component handles ordinary classification violations, while the second and third components impose ordering constraints for edge and reference cases.

Classification hinge loss. The classification loss penalizes examples whose signed score does not reach the unit margin:

$$\mathcal{L}_{\text{cls}} = \frac{1}{N} \sum_{i=1}^N [1 - y_i f_{\mathbf{w}}(\tilde{\mathbf{x}}_i)]_+, \quad (22)$$

where $[z]_+ = \max(0, z)$. This term is positive whenever an example is misclassified or classified with insufficient confidence.

Reference-case hinge loss. A reference case is an anchor example whose desired relative position is known in advance from the theoretical construction or annotation. In the current implementation, the reference example, indexed by r , is required to obtain a higher score than every other training example.

The reference loss is therefore

$$\mathcal{L}_{\text{ref}} = \frac{1}{N-1} \sum_{\substack{i=1 \\ i \neq r}}^N [m_{\text{ref}} - (f_{\mathbf{w}}(\tilde{\mathbf{x}}_r) - f_{\mathbf{w}}(\tilde{\mathbf{x}}_i))]_+, \quad (23)$$

where $m_{\text{ref}} > 0$ is the required reference margin. The loss is zero only when the reference score exceeds every other score by at least m_{ref} .

Edge-case hinge loss. Edge cases are examples located at important boundaries or extremes of the score space. Their desired ordering is known, but the score difference may be small. They are therefore explicitly constrained during training.

Let ℓ denote the lower-scoring edge case and h denote the higher-scoring edge case. The pairwise edge constraint is

$$\mathcal{L}_{\text{edge}}^{\text{pair}} = [m_{\text{edge}} - (f_{\mathbf{w}}(\tilde{\mathbf{x}}_h) - f_{\mathbf{w}}(\tilde{\mathbf{x}}_\ell))]_{+}. \quad (24)$$

This term is zero when the higher edge case exceeds the lower edge case by at least m_{edge} .

In addition, let b denote the designated lowest-scoring edge case. To ensure that this case remains globally lowest, we impose

$$\mathcal{L}_{\text{edge}}^{\text{lowest}} = \frac{1}{N-1} \sum_{\substack{i=1 \\ i \neq b}}^N [m_{\text{lowest}} - (f_{\mathbf{w}}(\tilde{\mathbf{x}}_i) - f_{\mathbf{w}}(\tilde{\mathbf{x}}_b))]_{+}. \quad (25)$$

The complete edge-case loss is

$$\mathcal{L}_{\text{edge}} = \omega_{\text{pair}} \mathcal{L}_{\text{edge}}^{\text{pair}} + \omega_{\text{lowest}} \mathcal{L}_{\text{edge}}^{\text{lowest}}, \quad (26)$$

where ω_{pair} and ω_{lowest} control the importance of the two edge constraints.

Total objective. The three hinge-loss components are combined into the total training objective:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{cls}} \mathcal{L}_{\text{cls}} + \lambda_{\text{edge}} \mathcal{L}_{\text{edge}} + \lambda_{\text{ref}} \mathcal{L}_{\text{ref}}. \quad (27)$$

The parameter vector is optimized using stochastic gradient descent:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} \mathcal{L}_{\text{total}}, \quad (28)$$

where η is the learning rate. The optimization continues for a fixed number of iterations, after which the final scores are computed as $f_{\mathbf{w}}(\tilde{\mathbf{x}}_i)$.

Algorithm 1: Pseudo code for demonstration of Main, Edge-Case, and Reference-Case Hinge Losses

```

for  $t \leftarrow 1$  to MAX_ITERS do
    optimizer.zero_grad();
     $\mathbf{s} \leftarrow X\mathbf{w} + b$ ;
     $\mathcal{L}_{\text{main}} \leftarrow \frac{1}{N} \sum_{i=1}^N \max(0, 1 - y_i s_i)$ ;
     $s_{\text{edge}} \leftarrow \mathbf{x}_{\text{edge}}^\top \mathbf{w} + b$ ;
     $\mathcal{L}_{\text{edge}} \leftarrow \max(0, 1 - y_{\text{edge}} s_{\text{edge}})$ ;
     $s_{\text{ref}} \leftarrow \mathbf{x}_{\text{ref}}^\top \mathbf{w} + b$ ;
     $\mathcal{L}_{\text{ref}} \leftarrow \max(0, 1 - y_{\text{ref}} s_{\text{ref}})$ ;
     $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{main}} + \mathcal{L}_{\text{edge}} + \mathcal{L}_{\text{ref}}$ ;
     $\mathcal{L}_{\text{total}}.$ backward();
    optimizer.step();

```
